Altova XMLSpy 2023 Enterprise Edition



Manual del usuario y referencia

Altova XMLSpy 2023 Enterprise Edition
Manual del usuario y referencia

Todos los derechos reservados. Ningún fragmento de esta publicación podrá ser reproducido de manera alguna (ya sea de forma gráfica, electrónica o mecánica, fotocopiado, grabado o reproducido en sistemas de almacenamiento y recuperación de información) sin el consentimiento expreso por escrito de su autor/editor.

Los productos a los que se hace referencia en este documento pueden ser marcas registradas de sus respectivos propietarios. El autor y editor no afirman ser propietarios de dichas marcas registradas.

Durante la elaboración de este documento se tomaron todas las precauciones necesarias para prevenir errores. Sin embargo, el autor y editor no se responsabilizan de los errores u omisiones que pudiese contener el documento ni de los posibles daños o perjuicios derivados del uso del contenido de este documento o de los programas y código fuente que vengan con el documento. Bajo ninguna circunstancia se podrá considerar al autor y editor responsables de la pérdida de beneficios ni de cualquier otro daño y perjuicio derivado directa o indirectamente del uso de este documento.

Fecha de publicación: 2022

© 2016-2022 Altova GmbH

Contenido

1 Acerca de XMLSpy y su documentación 29

1.1 Novedades 2023 ... 30
1.1.1 Versión 2022 ... 30
1.1.2 Versión 2021 ... 31
1.1.3 Versión 2020 ... 32
1.1.4 Versión 2019 ... 33
1.1.5 Versión 2018 ... 34
1.2 Rutas de acceso de archivos Windows ... 36
1.3 RaptorXML Server ... 37

2 Tutorial de XMLSpy 38

2.1 Interfaz de XMLSpy ... 39
2.1.1 Vistas ... 40
2.1.2 Ventanas ... 41
2.1.3 Menús y barras de herramientas ... 43
2.1.4 Configuración de la vista Texto ... 46
2.1.5 Opciones de la aplicación ... 49
2.2 Esquemas XML: aspectos básicos ... 51
2.2.1 Crear un archivo de esquema XML nuevo ... 51
2.2.2 Definir espacios de nombres ... 53
2.2.3 Definir un modelo de contenido ... 54
2.2.4 Agregar elementos mediante operaciones de arrastrar y colocar ... 59
2.2.5 Configurar la vista del modelo de contenido ... 60
2.2.6 Terminar un esquema básico ... 62
2.3 Esquemas XML: aspectos avanzados ... 67
2.3.1 Tipos complejos y tipos simples ... 67
2.3.2 Referencias a elementos globales ... 75
2.3.3 Atributos y enumeraciones de atributos ... 77
2.4 Esquemas XML: características de XMLSpy ... 81

2.4.1 Navegar por el esquema ... 81
2.4.2 Documentación del esquema ... 83
2.5 Documentos XML ... 88
2.5.1 Crear un archivo XML nuevo ... 88
2.5.2 Especificar el tipo de un elemento ... 90
2.5.3 Introducir datos en la vista Cuadrícula ... 92
2.5.4 Introducir datos en la vista Texto ... 93
2.5.5 Validar el documento ... 98
2.5.6 Agregar elementos y atributos ... 102
2.5.7 Edición en la vista tabular y de base de datos ... 104
2.5.8 Modificar el esquema ... 108
2.6 Transformaciones XSLT ... 110
2.6.1 Asignar un archivo XSLT ... 110
2.6.2 Transformar el archivo XML ... 111
2.6.3 Modificar el archivo XSL ... 112
2.7 Gestión de proyectos ... 114
2.7.1 Ventajas de trabajar con proyectos ... 114
2.7.2 Crear un proyecto ... 114
2.8 ¡Eso es todo! ... 117

3 Interfaz y entorno 118

3.1 La interfaz gráfica de usuario (IGU) ... 119
3.1.1 Ventana principal ... 120
3.1.2 Ventana Proyecto ... 122
3.1.3 Ventana Información ... 124
3.1.4 Ayudantes de entrada ... 124
3.1.5 Ventana de resultados: Mensajes ... 125
3.1.6 Ventana de resultados: XPath/XQuery ... 127
3.1.7 Ventana de resultados: Esquema XSL ... 128
3.1.8 Ventana de resultados: HTTP ... 129
3.1.9 Ventana de resultados: Buscar en archivos ... 129
3.1.10 Ventana de resultados: Buscar en esquemas ... 131
3.1.11 Ventana de resultados: Buscar en XBRL ... 132
3.1.12 Ventana de resultados: Gráficos ... 132

3.1.13 Ventana de resultados: XULE ... 134
3.1.14 Barra de menú, barras de herramientas y barra de estado ... 134
3.2 El entorno de aplicación ... 136
3.2.1 Configuración y personalización ... 136
3.2.2 Tutoriales, proyectos y ejemplos ... 139
3.2.3 Características y archivos de ayuda ... 139

4 Vistas de edición 141

4.1 Copia de seguridad automática de archivos ... 143
4.2 Vista Texto ... 145
4.2.1 Formato en la vista Texto ... 146
4.2.2 Visualización de documentos ... 149
4.2.3 Edición en la vista Texto ... 151
4.2.4 Navegar por los documentos ... 154
4.2.5 Buscar y reemplazar ... 157
4.2.6 Ayudantes de entrada de la vista Texto ... 164
4.2.7 Vista dividida ... 165
4.2.8 Teclas de acceso rápido ... 167
4.3 Vista Cuadrícula ... 169
4.4 Vista Esquema ... 171
4.4.1 Modo XSD: XSD 1.0 o XSD 1.1 ... 173
4.4.2 Vista general del esquema ... 177
4.4.3 Vista del modelo de contenido ... 190
4.4.4 Atributos, aserciones y restricciones de identidad ... 212
4.4.5 Ayudantes de entrada de la vista Esquema ... 228
4.4.6 Validación y corrección rápida ... 238
4.4.7 Mensajes de aserción ... 240
4.4.8 Modificación del tipo base ... 242
4.4.9 Restricciones inteligentes ... 244
4.4.10 xml:base, xml:id, xml:lang, xml:space ... 248
4.4.11 Adelante y atrás: navegar de una posición a otra ... 250
4.5 Vista WSDL ... 251
4.5.1 Ventana principal ... 252
4.5.2 Ventana de vista general ... 256

4.5.3 Ayudante de entrada de detalles ... 263
4.6 Vista XBRL ... 264
4.6.1 Ventana principal: Elementos ... 265
4.6.2 Ventana principal: Definiciones, Presentación, Cálculo, Fórmula y Tabla ... 269
4.6.3 Ayudantes de entrada de la vista XBRL ... 272
4.6.4 Configuración de la vista XBRL ... 277
4.7 Vista Authentic ... 278
4.8 Vista Explorador ... 279
4.9 Vista Archivo ... 281
4.10 Teclas de acceso rápido ... 283

5 XML 285

5.1 Crear, abrir y guardar documentos XML ... 286
5.2 Asignación de esquemas y validación ... 288
5.3 Datos XML en la vista Texto ... 290
5.4 Datos XML en la vista Cuadrícula ... 293
5.4.1 Vista documentos ... 295
5.4.2 Estructura del documento ... 299
5.4.3 Contenido del documento ... 302
5.4.4 Arrastrar y colocar en la vista Cuadrícula ... 306
5.4.5 Vista de tablas ... 309
5.4.6 Filtros ... 314
5.4.7 Fórmulas ... 316
5.4.8 Imágenes ... 319
5.4.9 Gráficos ... 320
5.4.10 Menú contextual en la vista Cuadrícula ... 323
5.4.11 Configuración de la vista Cuadrícula ... 326
5.5 Datos XML en la vista Authentic ... 331
5.6 Ayudantes de entrada para documentos XML ... 333
5.7 Validar un documento XML ... 335
5.8 Espacios en blanco ... 337
5.9 Insertar fragmentos XML ... 339
5.10 Procesamiento con XSLT y XQuery ... 341
5.11 Fuentes en documentos PDF ... 343

5.12 Gráficos ... 346
5.12.1 Creación de gráficos ... 349
5.12.2 XPath de origen ... 353
5.12.3 Selección del eje X ... 357
5.12.4 Selección del eje Y ... 362
5.12.5 Datos del gráfico ... 366
5.12.6 Gráficos multicapa ... 368
5.12.7 Configuración de gráficos: referencia rápida ... 369
5.12.8 Configuración y aspecto ... 372
5.12.9 Exportación ... 399
5.12.10 Ejemplo de gráfico básico ... 399
5.12.11 Ejemplo de gráfico avanzado ... 401
5.12.12 Ejemplo de gráfico de velas ... 407
5.13 Firmas XML ... 411
5.13.1 Crear firmas XML ... 413
5.13.2 Verificar firmas XML ... 416
5.13.3 Trabajar con certificados ... 419
5.14 Características adicionales ... 423

6 DTD y esquemas XML 425

6.1 Gestor de esquemas ... 426
6.1.1 Ejecutar el gestor de esquemas ... 429
6.1.2 Categorías de estado ... 431
6.1.3 Aplicar parches o instalar un esquema ... 433
6.1.4 Desinstalar o restaurar esquemas ... 434
6.1.5 Interfaz de la línea de comandos (ILC) ... 435
6.2 Archivos DTD ... 442
6.3 Esquemas XML ... 444
6.4 Subconjuntos de esquema ... 445
6.5 Reglas de esquema ... 449
6.5.1 Administrar conjuntos de reglas ... 449
6.5.2 Definir un conjunto de reglas ... 451
6.6 Catálogos en XMLSpy ... 456
6.6.1 Funcionamiento de los catálogos ... 456

6.6.2 Estructura de los catálogos en XMLSpy ... 457
6.6.3 Personalizar catálogos ... 459
6.6.4 Variables de entorno ... 460
6.7 Trabajar con SchemaAgent ... 463
6.7.1 Conectarse a SchemaAgent Server ... 464
6.7.2 Abrir esquemas encontrados en la ruta de búsqueda ... 466
6.7.3 Relaciones IIR ... 467
6.7.4 Visualizar esquemas en SchemaAgent ... 472
6.7.5 Validación con SchemaAgent ... 472
6.8 Búsqueda en esquemas ... 475
6.8.1 Término de búsqueda ... 477
6.8.2 Componentes ... 478
6.8.3 Propiedades ... 480
6.8.4 Ámbito ... 483
6.8.5 Comandos Buscar y Reemplazar ... 484
6.8.6 Resultados e información ... 486
6.8.7 Buscar y renombrar componentes globales ... 487

7 XSLT 490

7.1 Documentos XSLT ... 491
7.2 Procesamiento XSLT ... 493
7.3 Esquema XSL ... 496
7.3.1 Ventana Esquema XSL ... 497
7.3.2 Ventana Información ... 500
7.4 Optimizador de velocidad XSL ... 503

8 XQuery 505

8.1 Editar documentos XQuery ... 507
8.1.1 Documentos XQuery ... 507
8.1.2 Ayudantes de entrada para XQuery ... 509
8.1.3 Color de sintaxis para XQuery ... 510
8.1.4 Edición inteligente para XQuery ... 510
8.2 Evaluar expresiones XQuery ... 514

8.3 Validar documentos XQuery ... 515
8.4 Ejecutar XQuery/XQuery Update ... 516
8.5 XQuery Update Facility ... 519
8.5.1 Vista previa de actualizaciones ... 519
8.5.2 Operaciones y sintaxis de XQuery Update ... 522
8.6 XQuery y bases de datos XML ... 526

9 Depurador y generador de perfiles XSLT y XQuery 530

9.1 Depurador XSLT y XQuery ... 531
9.1.1 Funcionamiento e interfaz ... 532
9.1.2 Comandos e iconos de la barra de herramientas ... 534
9.1.3 Configurar el depurador ... 537
9.1.4 Iniciar una sesión de depuración ... 539
9.1.5 Ventanas de información ... 541
9.1.6 Puntos de interrupción ... 548
9.1.7 Puntos de seguimiento ... 551
9.1.8 Teclas de acceso rápido ... 557
9.2 Generador de perfiles XSLT y XQuery ... 558
9.2.1 Generación de perfiles XSLT ... 563
9.2.2 Generación de perfiles XQuery ... 567
9.2.3 Resultados del generador de perfiles ... 570

10 Expresiones XPath/XQuery 573

10.1 La ventana XPath/XQuery ... 574
10.2 Modo Evaluador ... 576
10.3 Modo Depurador ... 582
10.4 Generador de expresiones ... 590
10.5 Expresiones XQuery para JSON ... 593
10.6 Aspectos importantes ... 595

11 Authentic 597

11.1 Tutorial de la vista Authentic ... 599
11.1.1 Abrir un documento en la vista Authentic ... 600

11.1.2 La interfaz de la vista Authentic ... 601
11.1.3 Operaciones con nodos ... 604
11.1.4 Introducir datos en la vista Authentic ... 607
11.1.5 Introducir valores de atributo ... 609
11.1.6 Añadir entidades ... 610
11.1.7 Imprimir el documento ... 611
11.2 Interfaz de la vista Authentic ... 613
11.2.1 La interfaz gráfica del usuario (IGU) ... 613
11.2.2 Iconos de la barra de herramientas de la vista Authentic ... 614
11.2.3 Ventana principal de la vista Authentic ... 617
11.2.4 Ayudantes de entrada de la vista Authentic ... 619
11.2.5 Menús contextuales de la vista Authentic ... 624
11.3 Edición en la vista Authentic ... 627
11.3.1 Edición básica ... 627
11.3.2 Tablas en la vista Authentic ... 632
11.3.3 Editar una BD ... 639
11.3.4 Trabajar con fechas ... 646
11.3.5 Definir entidades ... 648
11.3.6 Firmas XML ... 650
11.3.7 Imágenes en la vista Authentic ... 651
11.3.8 Teclas de acceso rápido en la vista Authentic ... 652
11.4 Scripting de Authentic ... 653

12 HTML y CSS 656

12.1 HTML ... 657
12.2 CSS ... 659

13 JSON y esquemas JSON 665

13.1 Datos JSON ... 668
13.2 Esquemas JSON ... 671
13.3 Líneas JSON y JSON con comentarios ... 673
13.4 Documentos JSON en la vista Texto ... 674
13.5 Vista Cuadrícula JSON ... 679

13.5.1 Mostrar documentos en la Vista Cuadrícula ... 682
13.5.2 Editar la estructura de un documento ... 685
13.5.3 Editar el contenido de un documento ... 686
13.5.4 Arrastrar y colocar en la vista Cuadrícula ... 689
13.5.5 Vista Diseño de tabla ... 692
13.5.6 Filtros ... 696
13.5.7 Fórmulas ... 697
13.5.8 Imágenes ... 702
13.5.9 Gráficos ... 703
13.5.10 Menú contextual de la vista Cuadrícula ... 706
13.5.11 Configurar la vista Cuadrícula ... 709
13.6 Vista Esquema JSON ... 714
13.6.1 Versión del esquema JSON ... 715
13.6.2 Agregar definiciones globales ... 717
13.6.3 Ayudantes de entrada: Vista general, Detalles y Restricciones ... 718
13.6.4 Definiciones globales y locales ... 721
13.6.5 Vista de diseño ... 723
13.6.6 Objetos y propiedades ... 725
13.6.7 Propiedades sin especificar ... 729
13.6.8 Objetos y dependencias ... 732
13.6.9 Matrices ... 736
13.6.10 Tipos atómicos ... 738
13.6.11 Selectores de tipo (cualquiera, múltiple, etc.) ... 740
13.6.12 BSON (JSON binario) para MongoDB ... 742
13.6.13 Operadores ... 746
13.6.14 Condicionales ... 748
13.6.15 Configurar la vista Diseño ... 749
13.6.16 Generar documentación de esquemas JSON ... 751
13.7 Validar documentos JSON ... 754
13.8 Insertar fragmentos JSON ... 756
13.9 Transformaciones JSON con XSLT/XQuery ... 758
13.10 Expresiones XQuery para JSON ... 760
13.11 Generar esquemas JSON a partir de instancias JSON ... 762
13.12 Generar instancias JSON a partir de esquemas JSON ... 765

13.13 Convertir datos JSON en XML y viceversa ... 766

14 Avro y Avro Schema 767

14.1 Esquemas Avro ... 769
14.2 Datos Avro en formato JSON ... 772
14.3 Vista Avro: vista cuadrícula de binarios Avro ... 773

15 WSDL y SOAP 775

15.1 Tutorial de WSDL ... 776
15.1.1 Crear un documento nuevo ... 776
15.1.2 Crear un portType ... 777
15.1.3 Crear un enlace ... 780
15.1.4 Crear un servicio y puertos ... 781
15.1.5 Validar el documento WSDL ... 783
15.1.6 Conectarse a un servicio web y abrir archivos ... 783
15.1.7 Enviar una solicitud SOAP desde el archivo WSDL ... 785
15.1.8 Crear documentación WSDL ... 786
15.1.9 Conversión en WSDL 2.0 ... 787
15.2 SOAP ... 788
15.2.1 Validación SOAP ... 788
15.2.2 Depurador SOAP ... 790

16 HTTP 805

16.1 Enviar la solicitud ... 806
16.2 Importar una solicitud para enviarla ... 811
16.3 Recibir la respuesta ... 814

17 XBRL 817

17.1 Gestor de taxonomías ... 818
17.1.1 Ejecutar el gestor de taxonomías ... 822
17.1.2 Categorías de estado ... 824
17.1.3 Aplicar parches o instalar una taxonomía ... 825

17.1.4 Desinstalar o restaurar taxonomías ... 826
17.1.5 Interfaz de la línea de comandos (ILC) ... 827
17.2 Procedimientos básicos ... 834
17.2.1 Taxonomías nuevas y existentes ... 834
17.2.2 Introducción a los archivos de la taxonomía ... 835
17.2.3 Crear una taxonomía nueva ... 838
17.2.4 Importar una taxonomía ... 839
17.2.5 Espacios de nombres de la taxonomía ... 843
17.2.6 Configurar los archivos de la taxonomía ... 845
17.2.7 Agregar elementos a una taxonomía ... 847
17.2.8 Relaciones y funciones de vínculo ... 851
17.2.9 Crear relaciones: primera parte ... 853
17.2.10 Crear relaciones: segunda parte ... 855
17.3 Procedimientos adicionales ... 859
17.3.1 Etiquetas preferidas ... 859
17.3.2 Dominios con tipo ... 861
17.3.3 Detectar duplicados y deduplicar ... 862
17.3.4 Inline XBRL ... 863
17.4 Editor de fórmulas XBRL ... 864
17.4.1 Bases de enlaces y funciones de vínculo de fórmulas ... 864
17.4.2 Componentes de fórmulas ... 866
17.4.3 Editar el contenido y las propiedades de los componentes ... 882
17.4.4 Relaciones entre los componentes ... 884
17.4.5 Parámetros ... 885
17.4.6 Buscar componentes de fórmulas ... 888
17.5 Editor de definiciones de tabla XBRL ... 889
17.5.1 Bases de enlaces y funciones de vínculo de tablas ... 891
17.5.2 Estructura de las tablas ... 892
17.5.3 Componentes de tablas ... 905
17.5.4 Editar el contenido y las propiedades de los componentes ... 910
17.5.5 Relaciones entre los componentes ... 911
17.5.6 Parámetros de tablas ... 912
17.5.7 Vista previa del diseño de tabla ... 916
17.5.8 Buscar componentes de tabla ... 919

17.6 XULE ... 921
17.6.1 Documentos XULE ... 922
17.6.2 Ventana XULE ... 924
17.6.3 Ejecutar XULE ... 929
17.7 Búsqueda en XBRL ... 930
17.7.1 Término de búsqueda ... 930
17.7.2 Ejecución del comando ... 933
17.7.3 Resultados e información ... 935
17.8 OIM ... 936
17.9 Validación de instancias y taxonomías XBRL ... 937

18 Office Open XML, ZIP y EPUB 938
18.1 Trabajar con archivos OOXML ... 940
18.2 Archivos OOXML de ejemplo ... 942
18.3 Archivos ZIP ... 943
18.4 Archivos EPUB ... 945

19 Bases de datos 950
19.1 Conectarse a un origen de datos ... 952
19.1.1 Iniciar el asistente para la conexión de base de datos ... 953
19.1.2 Resumen de controladores de base de datos ... 955
19.1.3 Conexiones ADO ... 958
19.1.4 Conexiones ADO.NET ... 964
19.1.5 Conexiones ODBC ... 971
19.1.6 Conexiones JDBC ... 974
19.1.7 Conexiones PostgreSQL ... 979
19.1.8 Conexiones SQLite ... 980
19.1.9 Usar una conexión a través de recursos globales ... 984
19.1.10 Ejemplos de conexión a bases de datos ... 985
19.2 Bases de datos compatibles ... 1040

20 Recursos globales de Altova 1042
20.1 Definir recursos globales ... 1043

20.1.1 Archivos ... 1045
20.1.2 Carpetas ... 1050
20.1.3 Bases de datos ... 1051
20.2 Usar recursos globales ... 1054
20.2.1 Asignar archivos y carpetas ... 1054
20.2.2 Asignar bases de datos ... 1057
20.2.3 Cambiar de configuración ... 1058

21 Proyectos 1060

21.1 Crear y editar proyectos ... 1061
21.2 Usar proyectos ... 1066

22 RaptorXML(+XBRL) Server 1068

22.1 Agregar sevidores y configuraciones de servidor ... 1069
22.2 Validación de datos con RaptorXML Server ... 1073
22.3 Opciones de validación ... 1074
22.3.1 Opciones comunes ... 1075
22.3.2 XML con DTD ... 1075
22.3.3 DTD ... 1075
22.3.4 XML con W3C Schema ... 1076
22.3.5 W3C Schema ... 1077
22.3.6 Instancia inline XBRL ... 1078
22.3.7 Instancia XBRL ... 1080
22.3.8 Taxonomía XBRL ... 1081
22.3.9 Paquete de taxonomía XBRL ... 1083
22.3.10 Informe de versiones XBRL ... 1083
22.3.11 XSLT ... 1084
22.3.12 XQuery ... 1085
22.3.13 JSON ... 1086
22.3.14 JSON Schema ... 1086
22.3.15 AVRO ... 1087
22.3.16 AVRO JSON ... 1087
22.3.17 AVRO Schema ... 1087

22.3.18 EDGAR ... 1088
22.4 Transformaciones XSLT/XQuery con RaptorXML Server ... 1091

23 Comparación de archivos y directorios 1093

23.1 Comparación de archivos ... 1094
23.2 Comparación de directorios ... 1095

24 Control de código fuente 1097

24.1 Configurar el control de código fuente ... 1099
24.2 Sistemas de control de código fuente compatibles ... 1100
24.3 Carpeta de trabajo local ... 1102
24.4 Proyecto de la aplicación ... 1103
24.5 Agregar al control de código fuente ... 1105
24.6 Trabajar con control de código fuente ... 1108
24.6.1 Agregar o quitar del control de código fuente ... 1108
24.6.2 Proteger, desproteger ... 1109
24.6.3 Obtener archivos como archivos de solo lectura ... 1112
24.6.4 Copiar y compartir desde el control de código fuente ... 1113
24.6.5 Cambiar control de código fuente ... 1116
24.7 Control de código fuente con Git ... 1118
24.7.1 Habilitar Git con el complemento de control de código fuente ... 1119
24.7.2 Agregar un proyecto al control de código fuente de Git ... 1119
24.7.3 Clonar un proyecto desde el control de código fuente de Git ... 1121

25 XMLSpy en Visual Studio 1123

25.1 Instalación del complemento de XMLSpy ... 1124
25.2 Diferencias entre XMLSpy y XMLSpy para Visual Studio ... 1125
25.3 Depuradores de XMLSpy en Visual Studio ... 1128

26 XMLSpy en Eclipse 1129

26.1 Instalación del complemento de XMLSpy para Eclipse ... 1130
26.2 Perspectiva de XMLSpy en Eclipse ... 1132

26.3 Otros puntos de entrada de XMLSpy en Eclipse ... 1135
26.4 Perspectivas del depurador de XMLSpy ... 1137

27 Generador de código 1138

27.1 Novedades ... 1140
27.2 Generar código fuente a partir de esquemas XML o DTD ... 1142
27.2.1 Bibliotecas contenedoras de esquemas (C++) ... 1145
27.2.2 Bibliotecas contenedoras de esquemas (C#) ... 1148
27.2.3 Bibliotecas contenedoras de esquemas (Java) ... 1150
27.2.4 Integrar las bibliotecas contenedoras de esquemas ... 1152
27.2.5 Ejemplo: biblioteca ... 1155
27.2.6 Ejemplo: orden de compra ... 1179
27.3 Referencia de las clases generadas (C++) ... 1187
27.3.1 altova::DateTime ... 1187
27.3.2 altova::Duration ... 1191
27.3.3 altova::DayTimeDuration ... 1193
27.3.4 altova::YearMonthDuration ... 1194
27.3.5 altova::meta::Attribute ... 1194
27.3.6 altova::meta::ComplexType ... 1195
27.3.7 altova::meta::Element ... 1196
27.3.8 altova::meta::SimpleType ... 1197
27.3.9 [SuEsquema]::[CDoc] ... 1198
27.3.10 [SuEsquema]::[ElementType] ... 1200
27.3.11 [SuEsquema]::MemberAttribute ... 1202
27.3.12 [SuEsquema]::MemberElement ... 1202
27.4 Referencia de las clases generadas (C#) ... 1204
27.4.1 Altova.Types.DateTime ... 1204
27.4.2 Altova.Types.DateTimeFormat ... 1207
27.4.3 Altova.Types.Duration ... 1209
27.4.4 Altova.Xml.Meta.Attribute ... 1211
27.4.5 Altova.Xml.Meta.ComplexType ... 1212
27.4.6 Altova.Xml.Meta.Element ... 1213
27.4.7 Altova.Xml.Meta.SimpleType ... 1213
27.4.8 [SuEsquema].[Doc] ... 1214

27.4.9 [SuEsquema].[ElementType] 1216
27.4.10 [SuTipoEsquema].MemberAttribute 1217
27.4.11 [SuTipoEsquema].MemberElement 1218
27.5 Referencia de las clases generadas (Java) 1220
27.5.1 com.altova.types.DateTime 1221
27.5.2 com.altova.types.Duration 1226
27.5.3 com.altova.xml.meta.Attribute 1229
27.5.4 com.altova.xml.meta.ComplexType 1230
27.5.5 com.altova.xml.meta.Element 1231
27.5.6 com.altova.xml.meta.SimpleType 1231
27.5.7 com.[SuEsquema].[Doc] 1232
27.5.8 com.[SuEsquema].[ElementType] 1234
27.5.9 com.[SuEsquema].[SuTipoEsquema].MemberAttribute 1235
27.5.10 com.[SuEsquema].[SuTipoEsquema].MemberElement 1236
27.6 Consejos para la generación de código 1238
27.7 Opciones del generador de código 1239
27.8 SPL: el lenguaje de programación Spy 1241
27.8.1 Estructura básica de SPL 1241
27.8.2 Declaraciones 1242
27.8.3 Variables 1243
27.8.4 Variables predefinidas 1244
27.8.5 Crear archivos de salida 1246
27.8.6 Operadores 1247
27.8.7 Condiciones 1247
27.8.8 Colecciones y foreach 1248
27.8.9 Subrutinas 1250
27.8.10 Tipos integrados 1252

28 Comandos de menú 1257

28.1 Menú Archivo 1258
28.1.1 Nuevo 1258
28.1.2 Abrir 1262
28.1.3 Volver a cargar 1267
28.1.4 Codificación 1268

28.1.5 Cerrar, Cerrar todos, Cerrar documentos inactivos ... 1268
28.1.6 Guardar, Guardar como, Guardar todos ... 1269
28.1.7 Enviar por correo electrónico ... 1274
28.1.8 Imprimir ... 1275
28.1.9 Vista previa de impresión, Configurar impresión ... 1277
28.1.10 Archivos recientes, Salir ... 1278
28.2 Menú Edición ... 1279
28.2.1 Deshacer, Rehacer ... 1280
28.2.2 Cortar, Copiar, Pegar, Eliminar ... 1280
28.2.3 Copiar como texto XML/JSON ... 1281
28.2.4 Copiar como texto estructurado ... 1282
28.2.5 Copiar como imagen ... 1282
28.2.6 Copiar XPath ... 1283
28.2.7 Copiar XPointer/JSON-Pointer ... 1283
28.2.8 Insertar ... 1284
28.2.9 Guardar como imagen ... 1287
28.2.10 Pretty-Print ... 1288
28.2.11 Eliminar espacios en blanco ... 1289
28.2.12 Seleccionar todo ... 1289
28.2.13 Buscar, Buscar siguiente ... 1289
28.2.14 Reemplazar ... 1293
28.2.15 Buscar en archivos ... 1294
28.2.16 Marcadores ... 1296
28.2.17 Insertar o quitar comentario ... 1297
28.3 Menú Proyecto ... 1298
28.3.1 Proyecto nuevo ... 1301
28.3.2 Abrir proyecto ... 1301
28.3.3 Volver a cargar el proyecto ... 1301
28.3.4 Cerrar el proyecto ... 1301
28.3.5 Guardar el proyecto, Guardar el proyecto como ... 1302
28.3.6 Control de código fuente ... 1302
28.3.7 Agregar archivos al proyecto ... 1317
28.3.8 Agregar recurso global al proyecto ... 1317
28.3.9 Agregar URL al proyecto ... 1317

28.3.10 Agregar archivo activo al proyecto ... 1318
28.3.11 Agregar archivo activo y relacionados al proyecto ... 1318
28.3.12 Agregar carpeta de proyecto al proyecto ... 1318
28.3.13 Agregar carpeta externa al proyecto ... 1319
28.3.14 Agregar carpeta web externa al proyecto ... 1322
28.3.15 Configuración de script ... 1328
28.3.16 Propiedades ... 1328
28.3.17 Proyectos usados recientemente ... 1331
28.4 Menú XML ... 1332
28.4.1 Tipo ... 1333
28.4.2 Insertar antes/después ... 1333
28.4.3 Anexar, Agregar secundario ... 1333
28.4.4 Envolver en matriz/objeto ... 1333
28.4.5 Editar como texto sin formato ... 1334
28.4.6 Mover ... 1334
28.4.7 Mostrar como tabla ... 1334
28.4.8 Orden ascendente/descendente ... 1334
28.4.9 Intercambiar filas y columnas ... 1334
28.4.10 Evaluar XPath ... 1335
28.4.11 Comprobar formato XML ... 1335
28.4.12 Validar el documento XML ... 1337
28.4.13 Validar XML en el servidor (alto rendimiento) ... 1342
28.4.14 Validar archivos WSDL ... 1343
28.4.15 Validar al editar ... 1344
28.4.16 Actualizar ayudantes de entrada ... 1344
28.4.17 Prefijo de espacio de nombres ... 1344
28.4.18 Crear firma XML ... 1344
28.4.19 Verificar la firma XML ... 1347
28.5 Menú JSON ... 1351
28.5.1 Tipo ... 1351
28.5.2 Insertar, anexar, agregar secundario ... 1352
28.5.3 Envolver en matriz/objeto ... 1352
28.5.4 Mover ... 1353
28.5.5 Mostrar como tabla ... 1353

28.5.6 Orden ascendente/descendente ... 1353
28.5.7 Intercambiar filas y columnas ... 1353
28.5.8 Eliminar comentarios, volver a evaluar todo ... 1354
28.6 Menú DTD/Esquema ... 1355
28.6.1 Asignar DTD ... 1355
28.6.2 Asignar esquema ... 1356
28.6.3 Incluir otra DTD ... 1356
28.6.4 Ir a la DTD ... 1357
28.6.5 Ir al esquema ... 1357
28.6.6 Ir a la definición ... 1357
28.6.7 Generar DTD o esquema ... 1358
28.6.8 Nivelar documento DTD ... 1360
28.6.9 Convertir DTD en esquema ... 1360
28.6.10 Nivelar el esquema ... 1362
28.6.11 Convertir el esquema en DTD ... 1363
28.6.12 Convertir en UML ... 1363
28.6.13 Generar XML con MapForce desde BD, Excel, EDI ... 1365
28.6.14 Diseñar resultado en HTML, PDF o Word con StyleVision ... 1365
28.6.15 Generar archivo XML/JSON de muestra ... 1365
28.6.16 Generar código de programa ... 1369
28.6.17 Vaciar caché en memoria ... 1369
28.7 Menú Diseño de esquemas ... 1370
28.7.1 Configuración del esquema ... 1370
28.7.2 Guardar diagrama ... 1373
28.7.3 Generar documentación ... 1373
28.7.4 Configurar opciones de vista ... 1378
28.7.5 Zoom ... 1381
28.7.6 Mostrar nodos globales ... 1382
28.7.7 Mostrar diagrama ... 1382
28.7.8 Extensiones de esquema para BD ... 1383
28.7.9 Conectarse al servidor de SchemaAgent ... 1387
28.7.10 Desconectarse del servidor de SchemaAgent ... 1388
28.7.11 Mostrar en SchemaAgent ... 1388
28.7.12 Validación con SchemaAgent ... 1388

28.7.13 Crear subconjunto de esquema ... 1389
28.7.14 Nivelar el esquema ... 1390
28.8 Menú XSL/XQuery ... 1392
28.8.1 Transformación XSL ... 1394
28.8.2 Optimizador de velocidad XSL ... 1394
28.8.3 Transformación XSL-FO ... 1395
28.8.4 Parámetros de XSL / Variables de XQuery ... 1396
28.8.5 Ejecutar XQuery/XQuery Update ... 1400
28.8.6 Habilitar asignación inversa ... 1401
28.8.7 Generación de perfiles para XSLT o XQuery ... 1403
28.8.8 Asignar XSL ... 1403
28.8.9 Asignar XSL-FO ... 1403
28.8.10 Asignar archivo XML de muestra ... 1404
28.8.11 Ir al archivo XSL ... 1404
28.8.12 Ir a la instrucción de origen ... 1404
28.8.13 Ir al nodo de contexto ... 1405
28.8.14 Iniciar el depurador ... 1405
28.8.15 Detener el depurador ... 1405
28.8.16 Reiniciar el depurador ... 1405
28.8.17 Finalizar la sesión de depurador ... 1406
28.8.18 Paso a paso por instrucciones ... 1406
28.8.19 Paso a paso para salir ... 1406
28.8.20 Paso a paso por procedimientos ... 1406
28.8.21 Mostrar nodo en ejecución ... 1407
28.8.22 Insertar o quitar punto de interrupción ... 1407
28.8.23 Insertar o quitar punto de seguimiento ... 1407
28.8.24 Habilitar o deshabilitar punto de interrupción ... 1407
28.8.25 Habilitar o deshabilitar punto de seguimiento ... 1408
28.8.26 Puntos de interrupción y seguimiento ... 1408
28.8.27 Ventanas del depurador ... 1409
28.8.28 Configurar depuración ... 1410
28.9 Menú Authentic ... 1411
28.9.1 Documento nuevo ... 1412
28.9.2 Editar datos de una base de datos ... 1412

28.9.3 Asignar una hoja de estilos de StyleVision ... 1413
28.9.4 Editar una hoja de estilos de StyleVision ... 1414
28.9.5 Seleccionar y editar una fila nueva con datos XML ... 1414
28.9.6 Firma XML ... 1415
28.9.7 Definir entidades XML ... 1417
28.9.8 Ver marcado ... 1419
28.9.9 RichEdit ... 1420
28.9.10 Anexar/insertar/duplicar/eliminar fila ... 1420
28.9.11 Mover fila, Eliminar fila ... 1421
28.9.12 Generar un documento HTML, RTF, PDF o Word 2007+ ... 1421
28.9.13 Ubicaciones de confianza ... 1421
28.10 Menú Base de datos ... 1423
28.10.1 Consultar la base de datos ... 1423
28.10.2 IBM DB2 ... 1439
28.10.3 SQL Server ... 1445
28.10.4 Oracle XML DB ... 1447
28.11 Menú Conversión ... 1452
28.11.1 Importar archivo de texto ... 1452
28.11.2 Importar datos de una base de datos ... 1455
28.11.3 Importar documento de Microsoft Word ... 1460
28.11.4 Crear esquema XML a partir de la estructura de la BD ... 1461
28.11.5 Importación de BD basada en el esquema XML ... 1466
28.11.6 Crear estructura de la BD a partir del esquema XML ... 1467
28.11.7 Exportar a archivos de texto ... 1471
28.11.8 Exportar a una base de datos ... 1474
28.11.9 Convertir instancia XML en instancia JSON y viceversa ... 1476
28.11.10 Convertir esquema XML en esquema JSON y viceversa ... 1480
28.11.11 Convertir en OIM xBRL-XML ... 1482
28.11.12 Convertir en OIM xBRL-JSON ... 1482
28.11.13 Convertir en OIM xBRL-CSV ... 1483
28.12 Menú Vista ... 1484
28.12.1 Vista Texto ... 1484
28.12.2 Vista Cuadrícula mejorada ... 1485
28.12.3 Vista Diseño de esquemas ... 1485

28.12.4 Vista Diseño WSDL ... 1485
28.12.5 Vista Taxonomía XBRL ... 1486
28.12.6 Vista Authentic ... 1486
28.12.7 Vista Explorador ... 1486
28.12.8 Expandir ... 1487
28.12.9 Contraer ... 1487
28.12.10 Expandir todo ... 1487
28.12.11 Contraer elementos no seleccionados ... 1487
28.12.12 Ancho de tabla óptimo ... 1488
28.12.13 Ajuste automático de línea ... 1488
28.12.14 Ir a la línea o carácter ... 1488
28.12.15 Ir al archivo ... 1489
28.12.16 Configurar la vista Texto ... 1489
28.13 Menú Explorador ... 1491
28.14 Menú WSDL ... 1492
28.14.1 Componentes WSDL 1.1 ... 1492
28.14.2 Componentes WSDL 2.0 ... 1495
28.14.3 Tipos, Guardar diagrama ... 1498
28.14.4 Generar documentación ... 1499
28.14.5 Reanalizar el documento WSDL ... 1503
28.14.6 Convertir en WSDL 2.0 ... 1503
28.14.7 Generar código WSDL con MapForce ... 1503
28.15 Menú SOAP ... 1504
28.15.1 Crear solicitud SOAP nueva ... 1504
28.15.2 Enviar solicitud al servidor ... 1506
28.15.3 Configuración de la solicitud SOAP ... 1507
28.15.4 Sesión de depurador SOAP ... 1511
28.15.5 Ejecutar ... 1512
28.15.6 Paso a paso ... 1512
28.15.7 Interrumpir en la siguiente solicitud ... 1513
28.15.8 Interrumpir en la siguiente respuesta ... 1513
28.15.9 Detener el servidor proxy ... 1513
28.15.10 Opciones del depurador SOAP ... 1513
28.16 Menú XBRL ... 1515

28.16.1 Funciones de arco ... 1515
28.16.2 Funciones de vínculo ... 1517
28.16.3 Prefijos de espacios de nombres ... 1519
28.16.4 Establecer espacio de nombres de destino ... 1519
28.16.5 Valores de parámetro ... 1520
28.16.6 Importar o hacer referencia ... 1521
28.16.7 Buscar componente por su Id ... 1523
28.16.8 Generar documentación ... 1523
28.16.9 Configurar la vista ... 1527
28.16.10 Generar XBRL con MapForce a partir de BD, Excel, CSV ... 1528
28.16.11 Presentar XBRL como HTML, PDF o Word con StyleVision ... 1528
28.16.12 Ejecutar fórmula (en el servidor) ... 1528
28.16.13 Generar tabla (en el servidor) ... 1531
28.16.14 Detectar duplicados (en el servidor) ... 1533
28.16.15 Ejecutar XULE ... 1534
28.16.16 Transformar Inline XBRL ... 1535
28.16.17 Validate EDGAR on Server ... 1536
28.16.18 Opciones de procesamiento ... 1536
28.17 Menú Herramientas ... 1538
28.17.1 Ortografía ... 1539
28.17.2 Opciones de ortografía ... 1542
28.17.3 Editor de script ... 1546
28.17.4 Macros ... 1547
28.17.5 Comparaciones ... 1547
28.17.6 Herramientas definidas por el usuario ... 1557
28.17.7 Recursos globales ... 1557
28.17.8 Configuración activa ... 1558
28.17.9 Gestionar servidores Raptor ... 1559
28.17.10 Servidores Raptor y configuraciones ... 1563
28.17.11 Gestor de taxonomías XBRL ... 1563
28.17.12 Gestor de esquemas XML ... 1563
28.17.13 Personalizar ... 1564
28.17.14 Restaurar barras de herramientas y ventanas ... 1583
28.17.15 Opciones ... 1583

28.18 Menú Ventanas ... 1629
28.19 Menú Ayuda ... 1631
28.19.1 Contenido, Índice, Buscar ... 1631
28.19.2 Mapa de teclado ... 1632
28.19.3 Activación, formulario de pedido, registro, actualizaciones ... 1633
28.19.4 Otros comandos ... 1636
28.20 Línea de comandos ... 1638

29 Referencia del programador 1639
29.1 Editor de scripts ... 1641
29.1.1 Crear un proyecto de scripting ... 1642
29.1.2 Comandos integrados ... 1656
29.1.3 Habilitar scripts y macros ... 1666
29.2 Complementos de XMLSpy para entornos IDE ... 1669
29.2.1 Cómo registrar complementos para entornos IDE ... 1669
29.2.2 Controles ActiveX ... 1670
29.2.3 Configuración XML ... 1670
29.2.4 Archivos de muestra ATL ... 1673
29.2.5 IXMLSpyPlugIn ... 1679
29.3 Application API ... 1686
29.3.1 Overview ... 1687
29.3.2 Interfaces ... 1723
29.3.3 Interfaces (obsolete) ... 1950
29.3.4 Enumerations ... 1987
29.3.5 Application API for Java (obsolete) ... 2001
29.4 ActiveX Integration ... 2052
29.4.1 Prerequisites ... 2052
29.4.2 Adding the ActiveX Controls to the Toolbox ... 2053
29.4.3 Integration at Application Level ... 2055
29.4.4 Integration at Document Level ... 2057
29.4.5 ActiveX Integration Examples ... 2060
29.4.6 Command Reference ... 2072
29.4.7 Object Reference ... 2091

30 Anexos 2113

30.1 Información sobre motores XSLT y XQuery ... 2114
30.1.1 XSLT 1.0 ... 2114
30.1.2 XSLT 2.0 ... 2114
30.1.3 XSLT 3.0 ... 2116
30.1.4 XQuery 1.0 ... 2117
30.1.5 XQuery 3.1 ... 2119
30.2 Funciones XSTL y XPath/XQuery ... 2120
30.2.1 Funciones de extensión de Altova ... 2121
30.2.2 Funciones de extensión varias ... 2213
30.3 Tipos de datos en esquemas XML generados a partir de BD ... 2233
30.3.1 ADO ... 2233
30.3.2 MS Access ... 2234
30.3.3 MS SQL Server ... 2235
30.3.4 MySQL ... 2236
30.3.5 ODBC ... 2236
30.3.6 Oracle ... 2237
30.3.7 Sybase ... 2238
30.4 Tipos de datos en BD generadas a partir de esquemas XML ... 2240
30.4.1 MS Access ... 2240
30.4.2 MS SQL Server ... 2242
30.4.3 MySQL ... 2244
30.4.4 Oracle ... 2246
30.5 Datos técnicos ... 2248
30.5.1 Requisitos de SO y memoria ... 2248
30.5.2 Motores XSLT y XQuery de Altova ... 2248
30.5.3 Compatibilidad con Unicode ... 2249
30.5.4 Uso de Internet ... 2249
30.6 Información sobre licencias ... 2251
30.6.1 Distribución electrónica de software ... 2251
30.6.2 Activación del software y medición de licencias ... 2252
30.6.3 Contrato de licencia para el usuario final ... 2253

Índice 2254

1 Acerca de XMLSpy y su documentación

Altova XMLSpy 2023 Enterprise Edition es el editor XML y JSON más avanzado del mercado para diseñar, editar y depurar aplicaciones empresariales relacionadas con tecnologías XML, XML Schema, XSLT, XQuery, SOAP, WSDL, servicios web, OOXML y XBRL. XMLSpy es compatible con Windows 7 SP1 con actualización de la plataforma, Windows 8, Windows 10, Windows 11 y Windows Server 2008 R2 SP1 con actualización de la plataforma o superior. XMLSpy está disponible en versiones de 64 y 32 bits y como parte del económico conjunto de herramientas Altova MissionKit.



Esta documentación se divide en cuatro grandes secciones:

- Tutorial de XMLSpy 38
- Descripción de las funciones de XMLSpy
- Descripción de los comandos de menú 1257
- Referencia del programador 1639
- Anexos 2113

Última actualización: *13 October 2022*

1.1 Novedades 2023

- Puede elegir entre los temas clásico, ligero u oscuro[1629]. Active el tema que active, puede configurar el texto de los tipos de documento aparte y visualizarlo en el cuadro de diálogo Opciones[1583].
- También puede seleccionar un tema de XMLSpy al integrar en la aplicación Eclipse[1129] y Visual Studio[1123].
- El nuevo componente Gestor de esquemas de Altova[426] permite instalar cómodamente los DTD y esquemas XML más comunes e integrarlos para usarlos con XMLSpy y otros productos de Altova. Puede acceder al cuadro de diálogo del Gestor de esquemas con el comando de menú **Herramientas | Gestor de esquemas XML**[1563].
- Ahora puede elegir Microsoft Edge WebView2 como explorador para la vista Explorador[1597]. Esta es la alternativa más reciente a Internet Explorer que puede usar como motor de navegación en la vista Explorador[279].
- Si está usando el motor de navegación Microsoft Edge WebView2 en la vista Explorador[279], puede usar las herramientas de desarrollo[279] para depurar y probar su código HTML.
- Al validar XBRL inline ahora puede comprobar la validez del documento con respecto al Manual de informes ESEF (Formato Electrónico Único Europeo). Puede encontrar esta opción en la sección XBRL inline del cuadro de diálogo Opciones[1620].
- Hemos mejorado la ventana de resultados HTTP[129] con un panel de registro[805] e información sobre la configuración del proxy[806].
- También hemos ampliado la compatibilidad con bases de datos[1040] a: SQLite 3.38.5.
- Hemos ampliado la compatibilidad con Eclipse[1129] a las versiones: 2022-09; 2022-06; 2022-03; 2021-12.

1.1.1 Versión 2022

Versión 2022r2

- Una función nueva permite activar automáticamente el formato pretty-print para documentos XML y JSON[1591] al cargarlos en la vista Texto.
- En el cuadro de diálogo Opciones[1587] ahora puede indicar el tratamiento predeterminado que quiere dar a los archivos al abrirlos en XMLSpy. Este tratamiento afecta a los archivos que tengan extensiones para las que no se ha definido ningún tratamiento.
- Al generar documentos XML de muestra a partir de un esquema ahora puede indicar el nivel de anidación con el que se generan los elementos no obligatorios[1365].
- La conversión de esquemas XML en esquemas JSON[1480] ahora admite la definición de tipos dentro de objetos JSON. Este método es una alternativa a la definición de tipos mediante referencias.
- Las imágenes que estén almacenadas como cadenas cifradas en base 64 de documentos XML[285] y JSON[665] ahora se pueden guardar como imágenes- Esta función está disponible tanto en la vista Texto (para *XML*[290] y para *JSON*[674]) como en la vista Cuadrícula (para *XML*[319] y para *JSON*[702]) de estos documentos.
- La ventana Proyecto[122] ahora cuenta con una barra de herramientas que permite acceder rápidamente a los comandos de proyecto de uso frecuente.
- Al pasar el cursor del ratón por encima de un archivo de imagen que se encuentra en una carpeta de proyecto (en la ventana Proyecto[122]) aparece una vista previa de esa imagen.
- Hemos ampliado la compatibilidad con Eclipse[1129] a las versiones siguientes: 2021-12, 2021-09; 2021-06; 2021-03.

© 2016-2022 Altova GmbH

- Hemos ampliado la compatibilidad con Visual Studio[1123] a su versión 2022 (64 bits).
- Ahora la compatibilidad para generar código C++ y C#[1138] también incluye Visual Studio 2022 y .NET 6.
- También hemos ampliado la compatibilidad con bases de datos[1040] a: PostGreSQL 14.1, SQLite 3.37.2, MariaDB 10.6.5, MySQL 8.0.28.

Versión 2022

- Ahora XMLSpy permite editar esquemas BSON[742].
- Compatibilidad para convertir datos XBRL en formatos OIM[936] y validar documentos xBRL OIM como documentos XBRL[936].
- En los esquemas JSON de las versiones draft-2019-09 y posteriores se puede hacer referencia a otra definición de esquema y también contener definiciones locales propias. Se trata de un cambio con respecto a definiciones anteriores, donde se permitía una definición o definiciones locales, pero no ambas. La Vista Esquema JSON[714] actualizada ahora también admite estas referencias ampliadas[721].
- Mejoras en la administración de los espacios en blanco[337].
- Hemos mejorado la vista Cuadrícula[169] para que al desplazarse hacia abajo en un documento aparezca una barra anclada de encabezados que muestre los nodos antecesores del nodo activo.
- Ahora puede establecer un límite para el número de mensajes de validación[1595] que aparecen por separado para errores, advertencias e incoherencias XBRL.
- Las opciones del modo de salida[1584] permiten elegir qué hacer con los cambios sin guardar y los documentos abiertos al cerrar XMLSpy.
- En la ventana XPath/XQuery puede cargar/guardar snippets XPath/XQuery[590] desde/en archivos XQuery para más flexibilidad a la hora de usar expresiones y snippets XPath/XQuery.
- Hemos agregado funciones de extensión de Altova nuevas[2162] para acceder a la información del esquema: las funciones para calcular el tiempo Unix[2142].
- También hemos ampliado la compatibilidad con bases de datos[1040] a: PostGreSQL 13, IBM DB2 11.5 y MySQL 8.0.25.
- Hemos ampliado la compatibilidad con Eclipse[1129] a las versiones: 2021-09, 2021-06; 2021-03; 2021-12.

1.1.2 Versión 2021

Versión 2021 Release 3

Novedades de la **versión 2021r3** de XMLSpy:

- Hemos ampliado la compatibilidad con los esquemas JSON[671] a las versiones 2020-12 y 2019-09.
- Hemos ampliado la compatibilidad con Eclipse[1129] a las versiones: 2021-03, 2020-12; 2020-09; 2020-06.
- Hemos mejorado la vista Cuadrícula para documentos XML[293] y documentos JSON[679].

Versión 2021 Release 2

Novedades de la **versión 2021r2** de XMLSpy:

- Hemos mejorado la Vista Cuadrícula de datos XML[293] con funciones nuevas. También hemos reorganizado los comandos y las opciones relacionados con la vista para que sean más sencillos de usar: véanse la configuración de la vista Cuadrícula[326] y el cuadro de diálogo "[1591]Opciones"[1591].

- Hemos reorganizado las opciones de la vista Cuadrícula de documentos JSON[679] para que sea más fácil de usar: véanse la configuración de la vista Cuadrícula[326] y el cuadro de diálogo "Opciones"[1591].
- Hemos mejorado con nuevas funciones la vista Cuadrícula de documentos DTD[442].
- La nueva capa de arrastre de la vista Cuadrícula XML[306] y la vista Cuadrícula JSON[689] muestra información para ayudarle a arrastrar y colocar elementos en la vista Cuadrícula.
- También hemos incluido una característica nueva que permite añadir rápidamente fragmentos XML[339] y JSON[756] desde fuentes externas.
- Ahora en la vista Cuadrícula puede asignar formato específico para ciertos componentes[1606] a más tipos de componentes (como los comentarios XML). En la vista Cuadrícula puede ver cada tipo de componente con el formato que le haya asignado. Por ejemplo, puede hacer que los comentarios aparezcan resaltados en amarillo.
- Hemos introducido una funcionalidad nueva para validar documentos conforme los edita[1344] que puede activar o desactivar.
- Hemos ampliado la compatibilidad con Eclipse[1129] a las versiones siguientes: 2020-12; 2020-09; 2020-06; 2020-03.

Versión 2021

Novedades de la **versión 2021** de XMLSpy:

- La herramienta Gestor de taxonomías[818], que permite instalar, actualizar y administrar taxonomías fácilmente para usarlas con XMLSpy.
- El menú JSON[1351] presenta varias mejoras: ahora puede seleccionar un componente de la vista Cuadrícula JSON e insertar un elemento antes o después del mismo; también desplazar componentes dentro de la vista Cuadrícula hacia arriba, abajo, la derecha o la izquierda, así como eliminar todos los comentarios del documento.
- Más novedades de la vista Cuadrícula JSON: (i) hemos simplificado el menú contextual[706]; (ii) ahora puede acercar y alejar la cuadrícula[682]; (iii) puede expandir/contraer juntos[682] varios componentes del mismo nivel; (iv) las celdas que contienen cadenas pueden leerse también de derecha a izquierda (por ejemplo para idiomas como el árabe o el hebreo); también hemos mejorado el comando Copiar[706].
- La validación en la Vista Esquema JSON[714] toma en consideración la propiedad `$id`.
- New funciones de extensión de Altova que permiten acceder a la información del esquema[2162].
- Hemos ampliado la compatibilidad con Eclipse[1129] a las versiones siguientes: 2020.06; 2020.03; 2019.12; 2019.09.

1.1.3 Versión 2020

Versión 2020 Release 2

Novedades de la **versión 2020r2** de XMLSpy:

- XMLSpy guarda una copia de seguridad automática que se puede restaurar[143] de los archivos que se editen en su interfaz en caso de que la aplicación se cierre de forma inesperada. Puede indicar si quiere que se guarden copias de seguridad y con qué frecuencia[1584].
- El comando Validar al editar[335] resalta los errores de validación y formato conforme teclea tanto en la Vista Texto[335] como en la Vista Cuadrícula JSON[686]. Esta función se puede activar o desactivar en las opciones de programa[1584] o con el icono correspondiente de la barra de herramientas.
- En la Vista Esquema[171], el comando **Ir a la definición del tipo**[199] al que se accede desde distintos menús contextuales permite ir directamente a la definición del tipo (simple o complejo) de un componente.
- Las nuevas opciones de la vista Cuadrícula JSON permiten: (i) determinar si los filtros y las fórmulas[696] se guardan en un archivo de metadatos JSON para toda la aplicación, bajo demanda o

 © 2016-2022 Altova GmbH

nunca, y (ii) definir la ubicación de este archivo de metadatos JSON. Puede usar el archivo más tarde para aplicar los filtros y las fórmulas guardados en él a los archivos JSON asociados en distintas instancias de XMLSpy, lo que convierte esos a esos filtros y esas fórmulas en portátiles.
- Vista Cuadrícula JSON[706]: el menú contextual contiene comandos (i) para contraer los componentes no seleccionados y (ii) para eliminar todos los componentes de los documentos, incluyendo las fórmulas (que se guardan como comentarios).
- Vista Cuadrícula JSON[706]: hemos mejorado la función de búsqueda.
- Opciones para indicar qué se debe hacer con los espacios en blanco en XBRL Inline[1620].
- Métodos nuevos para generar código[1140].

Versión 2020:

Novedades de la **versión 2020** de XMLSpy:

- La Vista Cuadrícula JSON[679] cuenta con más funciones de edición, como (i) la detección automática de tipos, (ii) los comandos dentro de las celdas, (iii) los filtros XQuery para modificar la vista y (iv) las fórmulas XQuery para generar resultados complementarios a partir de datos JSON.
- Compatibilidad con Líneas JSON y JSON Comentarios[673] para su validación y edición.
- El cuadro de diálogo Opciones[1583] (**Herramientas | Opciones**) permite configurar la Vista Cuadrícula JSON y el comando pretty-print para documentos JSON en la vista Texto.
- Se han añadido un validador de XULE y un procesador de XULE. XULE es un lenguaje de consultas para documentos de instancia XBRL. Consulte la sección XBRL > XULE[921] para ver un resumen de las características XULE.
- Puede hacer consultas interactivas en documentos de instancia XBRL desde la Ventana XULE[924].
- Asistentes de edición para documentos XULE.[922]
- Se pueden procesar varios varios documentos Inline XBRL[863]como conjunto en lugar de tener que procesar primero cada documento por separado.
- El Evaluador y el Generador de expresiones de la ventana XPath/XQuery[573] estrenan diseño nuevo.
- La función Depurador XPath[582] de la ventana XPath/XQuery[127] incluye características nuevas: (i) expresiones de inspección, (ii) una interfaz mejorada para mayor comodidad y (iii) funciones de análisis aún más potentes. Gracias a estas mejoras ahora es incluso más fácil probar y depurar expresiones XPath/XQuery con respecto a archivos XML y JSON.
- Hasta ahora, la vista predeterminada de edición de archivos con distintos tipos de archivos se seleccionaba en el cuadro de diálogo "Opciones"[1587]. Ahora también puede elegirla directamente en la propia vista de edición[141].
- La compatibilidad para la integración de XMLSpy con Visual Studio[1123] se ha ampliado a Visual Studio 2019.
- La compatibilidad con Eclipse[1129] se ha ampliado a Eclipse 4.11 y 4.12.

1.1.4 Versión 2019

Versión 2019 Release 3

Novedades de la **versión 2019r3** de XMLSpy:

- La Vista Texto se ha mejorado con la Vista dividida[165], que permite trabajar con dos vistas del mismo documento a la vez para que no tenga que desplazarse arriba y abajo en el documento que está editando para consultar otras partes del mismo, algo que ahora puede hacer en la segunda vista.
- Compatibilidad con las nuevas versiones del esquema JSON: draft-06 y draft-07. Consulte el apartado de la documentación dedicado a JSON[665].

- El cuadro de diálogo "Opciones" (1583) (**Herramientas | Opciones**) ahora permite definir una ruta de acceso a un equipo virtual Java que sobrescriba a la que se detecta automáticamente.
- Compatibilidad con Windows Server 2019.
- La compatibilidad con Eclipse (1129) se ha ampliado a Eclipse 4.9 y 4.10.

Versión 2019

Novedades de la **versión 2019** de XMLSpy:

- Compatibilidad con más bases de datos: Firebird 3.0, IBM Informix 12.10, MariaDB 10.3, Microsoft SQL Server 2017, MySQL 8.0 y PostgreSQL 10.
- Integración con Eclipse 4.8.
- La compatibilidad con la recomendación Generic Preferred Label 1.0 permite especificar etiquetas preferidas para recursos XBRL (859), como relaciones de cálculos.
- Las definiciones de dominios XBRL con tipo (861) se muestran en la pestaña nueva *Dominios con tipo*, en el ayudante de entrada Detalles.
- Nuevos comandos de menú XBRL para detectar hechos duplicados en documentos de instancia XBRL (1533).
- Cuadro de diálogo Opciones de procesamiento XBRL (1536) que contiene opciones para: (i) ignorar hechos duplicados y (ii) generar tablas.
- En la sección XBRL del cuadro de diálogo Opciones (Herramientas **| Opciones**) ahora hay una pestaña que contiene opciones de validación de documentos de instancia XBRL (1619). Puede controlar documentos para (i) comprobar si son conformes a Dimensions 1.0 y Units Registry 1.0, (ii) si contienen hechos duplicados y (iii) si contienen incoherencias derivadas de cálculos.
- El ayudante de entrada Detalles (276) ahora también muestra los atributos personalizados de los elementos XBRL.

1.1.5 Versión 2018

Versión 2018 Release 2

Novedades de la **versión 2018 2018r2** de XMLSpy:

- En la ventana HTTP (805) puede importar una solicitud HTTP desde un archivo WSDL 1.1 o WSDL 2.0 (811). Por supuesto, sigue pudiendo importar solicitudes desde archivos WADL.
- XPath/XQuery 3.1 compatible con transformaciones y consultas JSON (758): (i) la ventana ventana de resultados XPath/XQuery (127) se ha mejorado para admitir consultas en documentos JSON, (ii) se pueden transformar documentos JSON (758) con documentos XSLT 3.0 y XQuery 3.0, y (iii) el depurador XSLT y XQuery (531) ahora también depura archivos JSON.
- Edición inteligente de XPath para plantillas de valores XSLT 3.0 en Vista Texto (491).
- Mejoras en la edición de la vista Texto (i) con más opciones de formato de texto relacionadas con sintaxis (1603) (espacio de nombres visible, etc.) y (ii) una mayor aplicabilidad (para más lenguajes).
- Compatibilidad de edición para los siguientes tipos de archivos: CSS, Python, lenguajes en C. Se puede configurar la conformidad de los tipos de archivos en la sección (1587) *Tipos de archivo* (1587) del cuadro de diálogo Opciones (1583).
- Mejoras en el diseño y la estructura del cuadro de diálogo Opciones (1583) (**Herramientas| Opciones**) para una mejor visibilidad de las opciones de la aplicación y mayor facilidad de uso.
- Distintas opciones para cambiar la configuración del proxy de red (1627).
- XML Spy obtiene la certificación de XBRL Internacional. Para más información acerca de la certificación XBRL, vaya a la página web (en inglés) XBRL Software Certification. Para saber cómo trabajar con XBRL en XMLSpy, consulte el apartado sobre XBRL de esta documentación (817).

© 2016-2022 Altova GmbH

Versión 2018

Novedades de la **versión 2018** de XMLSpy:

- Nueva ventana HTTP 805 donde se pueden crear solicitudes HTTP, enviar solicitudes a la URL de destino y recibir y consultar las respuestas.
- Mejoras en la característica de comparación de archivos 1547: ahora puede comparar tres archivos simultáneamente en una sola vista.
- Considerables mejoras en la velocidad del motor XSLT.
- Mejoras en el cuadro de diálogo Opciones 1583: mejora de la experiencia del usuario mediante la reorganización de los grupos de opciones, que ahora se presentan en forma de estructura jerárquica.
- Compatibilidad con XBRL Versioning 1.0.
- Compatibilidad con XML Signatures 1.1. 411
- Compatibilidad con Sybase 16, PostgreSQL 9.6, MySQL 5.7
- Integración con Eclipse 4.7
- Compatibilidad con FOP 2.2

1.2 Rutas de acceso de archivos Windows

Rutas de acceso de archivos en Windows 7, Windows 8 y Windows 10

Las rutas de acceso de archivo utilizadas en esta documentación no son iguales para todos los sistemas operativos.

- *Carpeta (Mis) Documentos:* esta carpeta se encuentra por defecto en la ubicación que aparece a continuación. Los archivos de ejemplo se encuentran en una subcarpeta de esta carpeta.

Windows 7/8/10	`C:/Usuarios/<usuario>/Documentos`

- *Carpeta de la aplicación:* esta es la carpeta en la que se encuentra su aplicación de Altova. Esta es la ruta de acceso predeterminada de la carpeta de la aplicación:

Windows 7/8/10	`C:\Archivos de programa\Altova\`
Versión de 32 bits en sistemas operativos de 64 bits	`C:\Archivos de programa (x86)\Altova\`

Nota: XMLSpy también es compatible con Windows Server 2008 R2 SP1 con actualización de la plataforma o superior.

© 2016-2022 Altova GmbH

1.3 RaptorXML Server

Altova RaptorXML(+XBRL) Server (en adelante RaptorXML o simplemente Raptor) es el rapidísimo procesador XML y XBRL* de tercera generación de Altova, optimizado para los estándares más recientes y para entornos de informática en paralelo. RaptorXML es compatible con múltiples plataformas y aprovecha la omnipresencia actual de equipos multinúcleo para ofrecer rapidísimas funciones de procesamiento de datos XML y XBRL.

Altova ofrece dos ediciones diferentes del procesador RaptorXML:

- **RaptorXML Server**, que se puede usar desde la red y puede transformar varios archivos a la vez.
- **RaptorXML+XBRL Server**, que se puede usar desde la red, puede transformar varios archivos a la vez y ofrece funciones de validación de datos XBRL.

RaptorXML puede ejecutarse desde la línea de comandos y tiene interfaces para COM, Java, .NET y Python. Es decir, puede utilizar RaptorXML desde estos entornos directamente para validar documentos XML, realizar transformaciones XSLT y ejecutar documentos XQuery.

También puede usar un servidor RaptorXML desde la interfaz gráfica de XMLSpy directamente 1068 para validar documentos XML y XBRL*.

*** Nota:** las funciones de procesamiento de datos XBRL solamente están disponibles en RaptorXML+XBRL Server (no están disponibles en RaptorXML Server).

Sitio web de Altova: Servidor de validación XML, validador XML

2 Tutorial de XMLSpy

Este tutorial ofrece una introducción al lenguaje XML y un recorrido por varias tareas claves relacionadas con XML. Durante el tutorial, aprenderá a usar algunas de las potentes funciones de XMLSpy.

El tutorial está compuesto por estas secciones:

- Interfaz de XMLSpy 39. En esta primera sección describimos la interfaz gráfica del usuario (IGU) de la aplicación para que se familiarice con ella.
- Crear un esquema XML 51. En esta sección aprenderá a crear un esquema XML en la intuitiva vista Esquema de XMLSpy. También aprenderá a crear modelos de contenido complejos mediante operaciones de arrastrar y colocar y a configurar la vista Esquema.
- Funciones avanzadas de la vista Esquema 67. En esta sección aprenderá a crear tipos complejos y simples, referencias a elementos globales y enumeraciones de atributos.
- También aprenderá a navegar por el esquema 81 en la vista Esquema y a generar documentación para los esquemas 81.
- Crear un documento XML 88. En esta sección aprenderá a asignar un esquema a un documento XML, editar el documento XML en la vista Texto o Cuadrícula y validar documentos XML con ayuda del validador integrado de XMLSpy.
- Transformar un archivo XML con una hoja de estilos XSLT 110. En esta sección aprenderá a asignar un archivo XSLT al documento y a realizar la transformación con ayuda de los motores XSLT integrados de XMLSpy.
- Trabajar con proyectos de XMLSpy 114. Esta sección le enseñará a organizar sus documentos XML con facilidad.

Instalación y configuración

Para poder completar este tutorial es necesario tener instalado XMLSpy en su equipo y tener un código clave de evaluación o estar registrado como usuario. La versión de evaluación de XMLSpy es gratis y totalmente funcional durante 30 días. Puede adquirir una licencia normal en nuestro servidor web seguro o a través de nuestros proveedores.

Archivos de ejemplo del tutorial

Los archivos del tutorial están guardados en esta carpeta:

```
C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Tutorial
```

La carpeta `Examples` contiene varios archivos XML para que experimente con ellos, mientras que la carpeta `Tutorial` incluye todos los archivos utilizados en este tutorial.

La carpeta `Template` ubicada en la carpeta de la aplicación (que suele estar en `C:\Archivos de programa\Altova`) incluye todas las plantillas XML que se utilizan si selecciona **Archivo | Nuevo**. Estos archivos ofrecen los datos necesarios (espacios de nombres y declaraciones XML) para empezar a trabajar inmediatamente con el documento XML correspondiente.

© 2016-2022 Altova GmbH

2.1 Interfaz de XMLSpy

En esta primera sección del tutorial iniciará XMLSpy y se familiarizará con su interfaz.

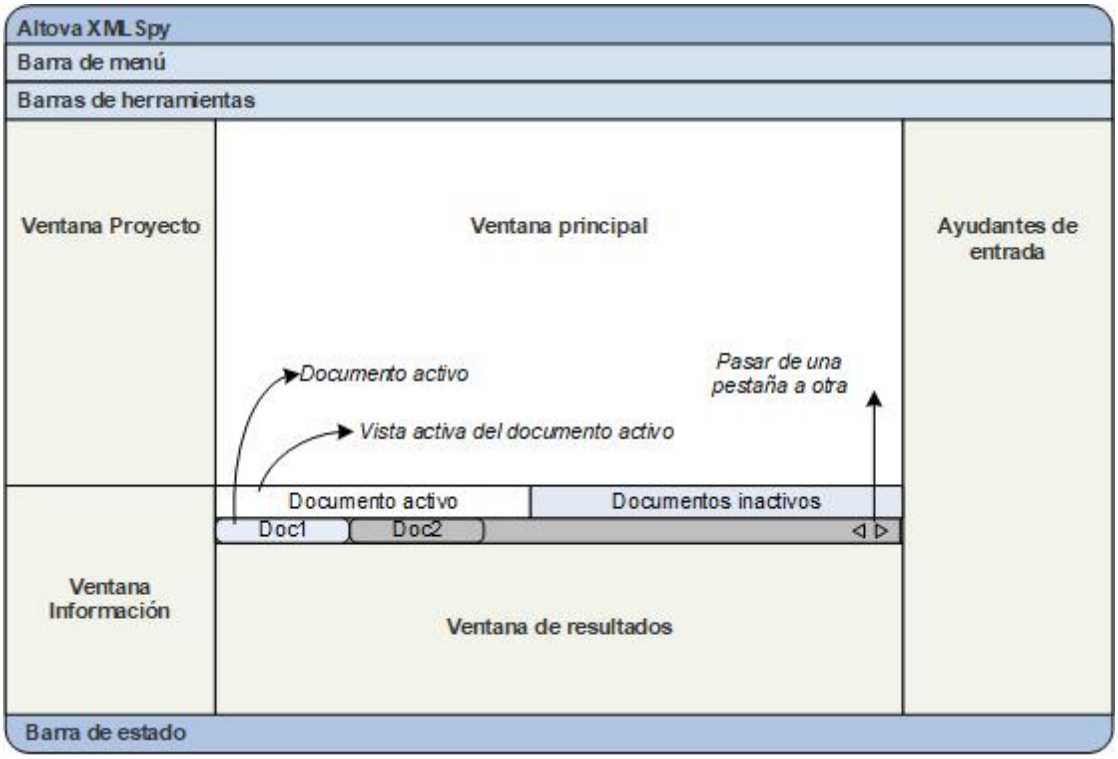
Iniciar XMLSpy

Para iniciar XMLSpy, haga doble clic en el icono de XMLSpy situado en el escritorio o utilice el menú **Inicio | Todos los programas**. XMLSpy se inicia con ningún documento abierto. Abra XMLSpy.

Introducción a la interfaz de XMLSpy

La vista predeterminada de la interfaz de XMLSpy se divide en tres áreas verticales (*imagen siguiente*). Estas tres áreas contienen de izquierda a derecha: (i) las ventanas Proyecto e Información, (ii) la ventana principal y las ventanas de resultados y (iii) las ventanas de los ayudantes de entrada. Mire la ventana Proyecto y el proyecto Examples, que se abre por defecto al iniciar XMLSpy por primera vez.

A continuación resumimos los demás componentes de la interfaz y sus funciones. Los apartados de esta sección explican estos componentes con más detalle.



Barra de documentos de la ventana principal: cuando hay varios documentos abiertos, cada documento aparece en una pestaña en la barra de documentos de la ventana principal (*imagen anterior*). Al hacer clic en una pestaña, el documento se convierte en el documento activo. Puede desplazarse por las pestañas de los documentos haciendo clic en las flechas situadas a la derecha de la barra de documentos. Abra varios documentos (del proyecto Examples, por ejemplo) y utilice las pestañas para ver cómo funcionan.

Vistas de edición de documentos: el documento activo puede verse en varias vistas de edición. Por ejemplo:

- Los documentos XML (`.xml`) se pueden ver en las vistas Texto, Cuadrícula, Authentic y Explorador, pero no se pueden ver en la vista Esquema.
- Los documentos de esquema XML (`.xsd`), por el contrario, se pueden ver en las vistas Texto, Cuadrícula, Esquema y Explorador, pero no en la vista Authentic.

Estas son todas las vista de XMLSpy: vista Texto 145, vista Cuadrícula 169, vista Esquema 171, vista WSDL 251, vista XBRL 264, vista Authentic 613 y vista Explorador 279.

Ayudantes de entrada: las ventanas de los ayudantes de entrada cambian dependiendo de qué tipo de documento esté activo (XML, XSD, CSS, WSDL, etc.) y de la vista activa (vista Texto, Esquema, etc.). Los ayudantes de entrada sirven para editar correctamente y con rapidez el documento activo ofreciendo opciones de edición contextuales.

2.1.1 Vistas

En este apartado del tutorial aprenderá (i) a cambiar de vista de edición y (ii) a cambiar la vista de edición predeterminada para cada tipo de documento.

Cambiar de vista de documento

Cuando abra un documento en XMLSpy, el documento se abre en la vista que se configuró como vista predeterminada para dicho tipo de documento. Siga estos pasos para abrir un documento:

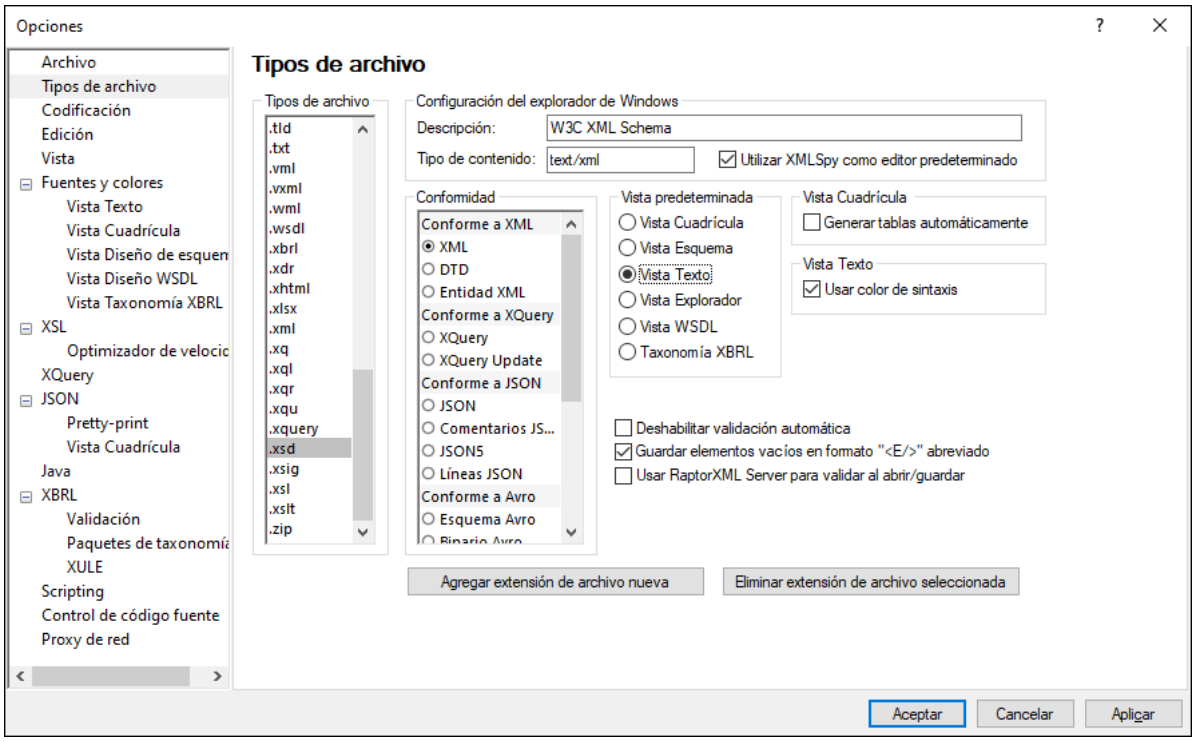
1. Haga clic en el comando **Archivo | Abrir**.
2. Busque el archivo `AddressFirst.xsd`, situado en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Tutorial`. Seleccione este archivo y haga clic en **Abrir**. El archivo se abre en la vista Esquema.
3. Cambie a las demás vistas haciendo clic en las pestañas situadas en la parte inferior de la ventana principal (Texto, Cuadrícula, etc.). Los documentos de esquema XML se pueden ver en las vistas Texto, Cuadrícula, Esquema y Explorador.
4. También puede cambiar de vista desde el menú **Vista**. Intente cambiar la vista del documento `AddressFirst.xsd` con los comandos del menú **Vista**.
5. Cierre el documento (**Archivo | Cerrar**).

Cambiar la vista de edición predeterminada para el documento

Todos los documentos con la extensión `.xsd` se abren por defecto en la vista Esquema. La vista predeterminada de cada tipo de documento se puede cambiar en el cuadro de diálogo "Opciones". Siga estos pasos para cambiar la vista predeterminada de todos los documentos `.xsd`:

1. Haga clic en el comando **Herramientas | Opciones** y abra la sección *Tipos de archivo* (*imagen siguiente*).
2. En el panel izquierdo *Tipos de archivo* busque la opción **.xsd** y selecciónela (*imagen siguiente*).
3. En el panel *Vista predeterminada* seleccione la opción *Vista Texto.*

 © 2016-2022 Altova GmbH



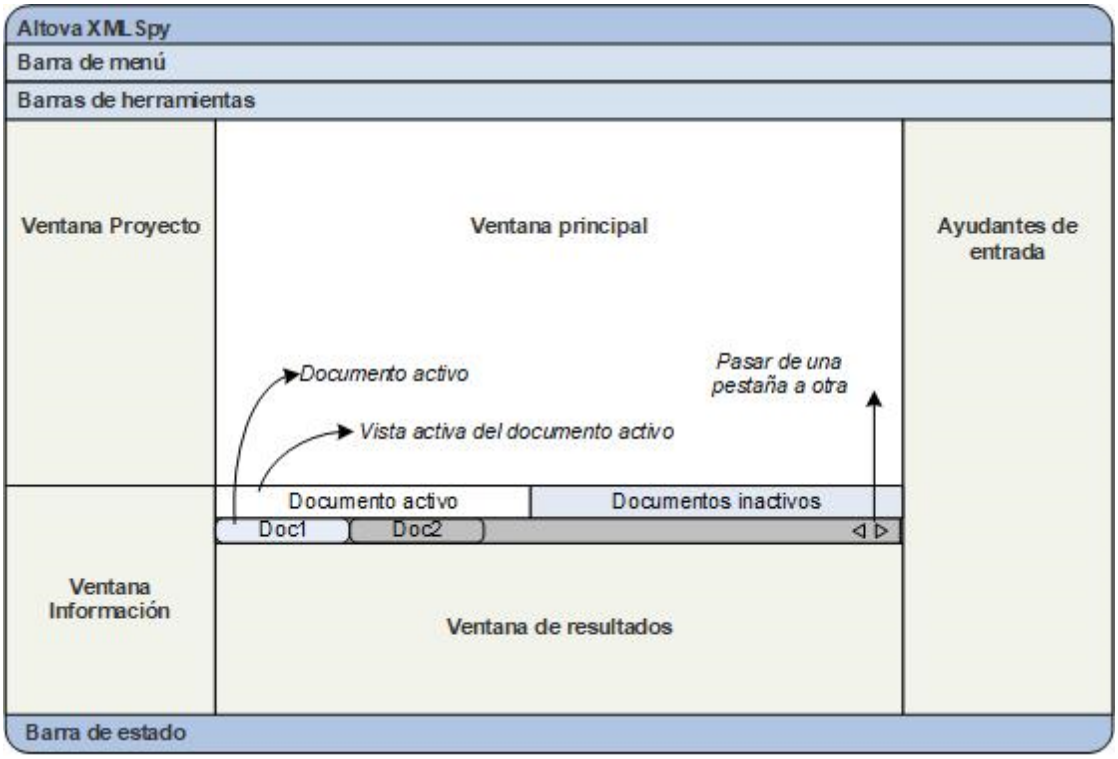
4. Haga clic en **Aceptar**.
5. Ahora seleccione el comando **Archivo | Abrir** y abra el archivo `AddressFirst.xsd`. El archivo se abre en la vista Texto.
6. Cambie a la vista Esquema para ver el archivo en esta vista y después ciérrelo (**Archivo | Cerrar**).
7. Vuelva al cuadro de diálogo "Opciones" (**Herramientas | Opciones**) y en la sección *Tipos de archivo* cambie la vista predeterminada de los archivos `.xsd` a la vista Esquema.

Nota: en la sección *Tipos de archivo* del cuadro de diálogo "Opciones" (*imagen anterior*) puede cambiar la vista predeterminada de las extensiones de archivo de la lista pero también puede añadir extensiones de archivo nuevas con el botón **Agregar extensión de archivo nueva**.

2.1.2 Ventanas

En la configuración predeterminada de XMLSpy, las ventanas están colocadas alrededor de la ventana principal (imagen siguiente). Se trata de estas ventanas:

- Ventana Proyecto
- Ventana Información
- Ayudantes de entrada (hay varios ayudantes, dependiendo del tipo de documento y de la vista)
- Ventanas de resultados: Mensajes, Gráficos, XPath, Esquema XSL, Buscar en archivos, Buscar en esquemas, Buscar en XBRL



En este apartado aprenderá a activar y desactivar los grupos de ventanas recién mencionados y a mover las ventanas por la pantalla. Esto le será muy útil más adelante cuando necesite utilizar todo el espacio posible de la interfaz.

Activar/desactivar los grupos de ventanas

Las ventanas de XMLSpy (ventana Proyecto, ventana Información, ventanas de los ayudantes de entrada y ventanas de resultados) se pueden mostrar u ocultar con los comandos del menú **Ventanas**. También puede ocultar un grupo de ventanas ya visible haciendo clic con el botón derecho en la barra de título y seleccionando el comando **Ocultar**. Las ventanas ocultas solamente se pueden activar otra vez desde el menú **Ventanas**.

Abra cualquier archivo XML de la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Tutorial` y pruebe estos comandos y operaciones hasta familiarizarse con ellos. Para más información consulte el apartado Interfaz de XMLSpy de la sección Introducción 119.

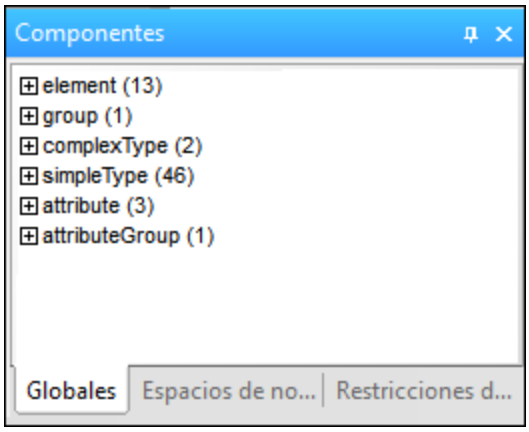
Estados guardado y con copia de seguridad

Por defecto, XMLSpy guarda copias de seguridad de los documentos sin guardar cada cinco segundos. Las pestañas de la parte inferior de la ventana principal informan mediante símbolos de si el archivo está o no guardado y de si se ha hecho una copia de seguridad del mismo. Es importante aprender qué significa cada símbolo para poder interpretarlos en el momento mientras trabaja. Para más información sobre estos indicadores consulte Copia de seguridad automática de archivos 143.

 © 2016-2022 Altova GmbH

Mover ventanas por la pantalla

Las ventanas pueden ser flotantes o estar acopladas a la interfaz. Las ventanas también se pueden acoplar en forma de pestaña dentro de un grupo de ventanas (ver párrafo anterior). Por ejemplo, la imagen siguiente muestra el ayudante de entrada Componentes de la vista Esquema, que tiene tres ventanas en pestañas: la ventana Globales, la ventana Espacios de nombres y la ventana Restricciones de identidad.



Para acoplar una ventana o hacerla flotante utilice cualquier método de los que se describen a continuación:

- Haga doble clic en la barra de título de la ventana. Si la ventana está acoplada, se convertirá en flotante. Si es una ventana flotante, se acoplará en la posición donde estuvo acoplada por última vez.
- Haga clic con el botón derecho en la barra de título de una ventana y elija el comando correspondiente (**Flotante** o **Acoplada**).
- Arrastre la ventana (usando su barra de título como manipulador) desde donde está acoplada hasta que sea flotante. También puede arrastrar una ventana flotante y colocarla en la posición donde desea acoplarla. En este caso aparecen dos conjuntos de flechas azules. Las cuatro flechas exteriores permiten acoplar la ventana en relación a la ventana de la aplicación (es decir, en el borde superior, inferior, derecha o izquierda de la IGU). Las cuatro flechas interiores permiten acoplar la ventana en relación a la ventana donde está el cursor en ese momento. Si coloca la ventana que está arrastrando en el botón situado en el centro de las flechas interiores (o en la barra de título de otra ventana), la ventana se acopla en forma de pestaña dentro de la otra ventana.

Para hacer flotante una ventana que es una pestaña, haga doble clic en su pestaña. Arrastre la pestaña de la ventana para sacarla del grupo de ventanas al que pertenece.

Abra cualquier archivo de esquema XML de la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Tutorial` y pruebe los métodos que acabamos de describir para mover las ventanas de posición hasta familiarizarse con ellos.

2.1.3 Menús y barras de herramientas

En este apartado del tutorial se familiarizará con las características principales de los menús y de las barras de herramientas de XMLSpy.

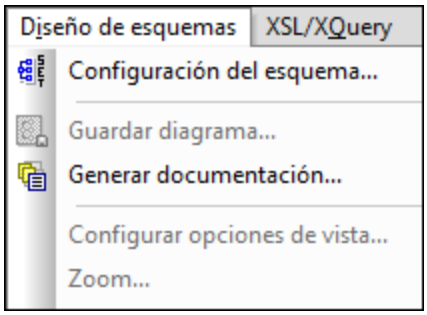
Menús

XMLSpy incluye dos barras de menús: (i) un menú predeterminado que aparece cuando no hay abierto ningún documento y (ii) el menú de aplicación de XMLSpy, que aparece en cuanto se abre un documento. Siga estas instrucciones:

1. Cierre todos los documentos con el comando de menú **Archivo | Cerrar todos**. Observe el menú predeterminado.
2. Abra el archivo `AddressFirst.xsd` haciendo clic en su nombre en la lista de archivos recientes situada al final del menú **Archivo**. Cuando el archivo se abre en la vista Esquema, el menú pasa al menú de aplicación de XMLSpy.

Los menús se organizan según su función y los comandos del menú se habilitan solamente cuando se pueden ejecutar en la posición del cursor o para la selección en la vista actual del documento activo. Siga estas instrucciones para comprender los factores que determinan si un comando se habilita o no:

1. Haga clic en el menú **Diseño de esquemas**. Observe que los comandos **Guardar diagrama**, **Configurar opciones de vista** y **Zoom** están deshabilitados (*imagen siguiente*).



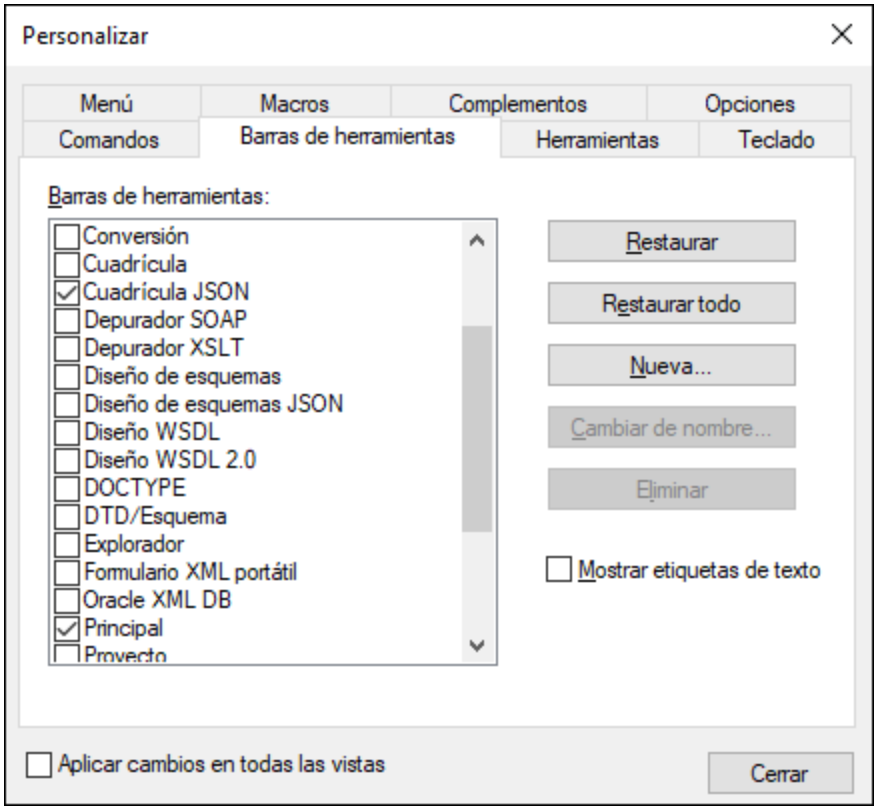
2. Haga clic en otro sitio para que desaparezca el menú. Ahora haga clic en el icono **Mostrar diagrama** situado a la izquierda del componente `element`. Se abre la vista del modelo de contenido de la vista Esquema (la segunda vista de la vista Esquema. La primera es la vista general del esquema). Abra otra vez el menú **Diseño de esquemas** y observe que los comandos **Guardar diagrama**, **Configurar opciones de vista** y **Zoom** ahora están habilitados. Estos comandos están disponibles solamente en la vista del modelo de contenido de la vista Esquema (no están disponibles en la vista general de la vista Esquema ni en ninguna otra vista). Recuerde además que en la vista Esquema solamente se pueden abrir archivos de esquema XML.
3. Los archivos de esquema XML también son archivos XML. Por eso, si los abre en la vista Texto o Cuadrícula, se presentan como archivos XML y todos los comandos relacionados con archivos XML están disponibles. Compare el menú **Edición** y sus comandos en la vista Esquema y en la vista Texto.
4. Después compare el menú **XML | Insertar** y sus comandos en la vista Texto y en la vista Cuadrícula. Estos comandos están disponibles solamente en la vista Cuadrícula.

Todos y cada uno de los comandos de la aplicación se describen en detalle en la *Comandos de menú* 1257 del manual.

© 2016-2022 Altova GmbH

Barras de herramientas

Dependiendo de la vista activa la aplicación incluye unas barras de herramientas u otras. Sin embargo, las barras de herramientas se pueden personalizar en la pestaña *Barras de herramientas* del cuadro de diálogo "Personalizar" (**Herramientas | Personalizar | Barras de herramientas**, *imagen siguiente*).



Ahora pruebe a mover las barras de herramientas por la interfaz del usuario. Haga clic en el manipulador de una barra de herramientas y arrástrela a otra posición. El manipulador de la barra de herramienta es la línea vertical de puntos situada a la izquierda de la barra (*imagen siguiente*).

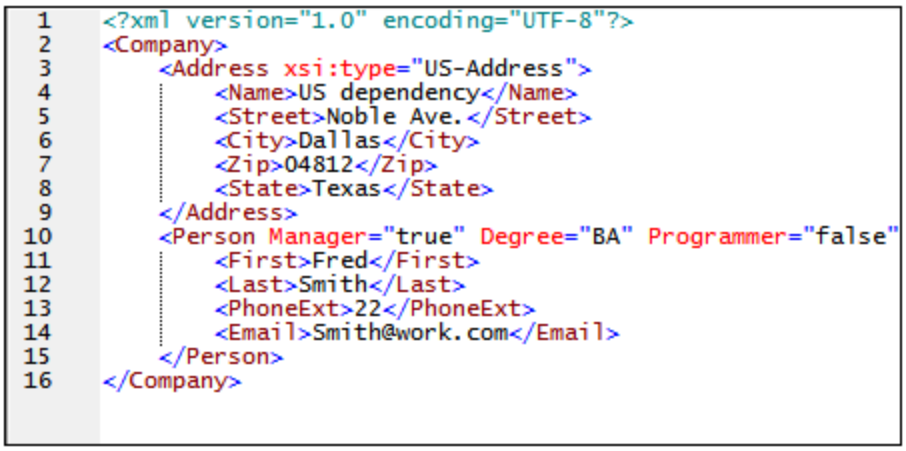


Por ejemplo, arrastre una de las barras de herramientas a estas posiciones: (i) a otra línea del área de barras de herramientas, (ii) a la izquierda o a la derecha de otra barra de herramientas, (iii) en el centro de la ventana principal o (iv) acoplada en el lateral izquierdo o derecho de la ventana de la aplicación (para ello debe colocar el manipulador sobre el borde izquierdo o derecho de la ventana de la aplicación).

Cuando termine, cierre el archivo `AddressFirst.xsd`.

2.1.4 Configuración de la vista Texto

En este apartado aprenderá a configurar un documento con formato *pretty-print* y a usar marcadores mientras edita el documento. Un documento tiene formato *pretty-print* cuando en la vista Texto se aplica sangría a cada nivel de la jerarquía XML de modo que los niveles inferiores aparecen con mayor sangría que los niveles anteriores (*imagen siguiente*). El uso de marcadores permite marcar posiciones del documento que desea volver a ver.

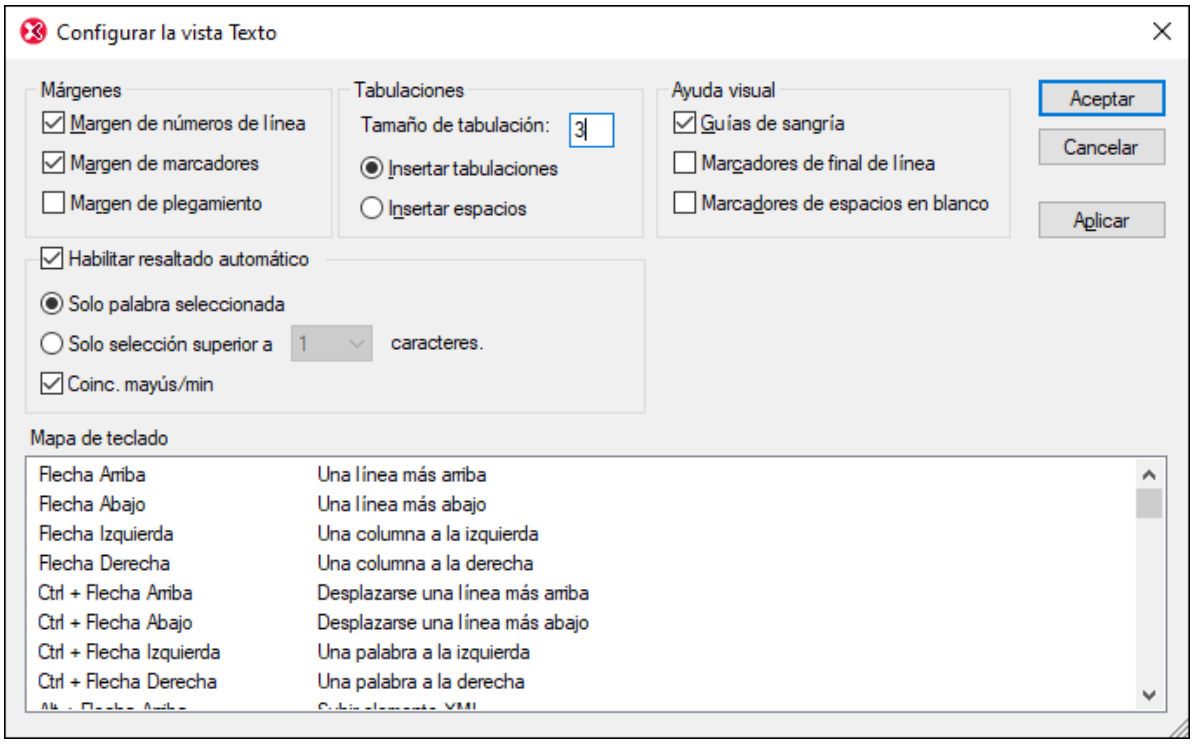


Formato pretty-print

Primero (i) configuramos el formato pretty-print, especificando la cantidad de sangría y después (ii) aplicamos el formato pretty-print.

1. Abra el archivo `CompanyFirst.xml` disponible en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Tutorial`.
2. Cambie a la vista Texto si esta vista no es la vista predeterminada de los documentos XML.
3. Abra el cuadro de diálogo "Configurar la vista Texto" con el comando de menú **Vista | Configurar la vista Texto** (*imagen siguiente*).

© 2016-2022 Altova GmbH



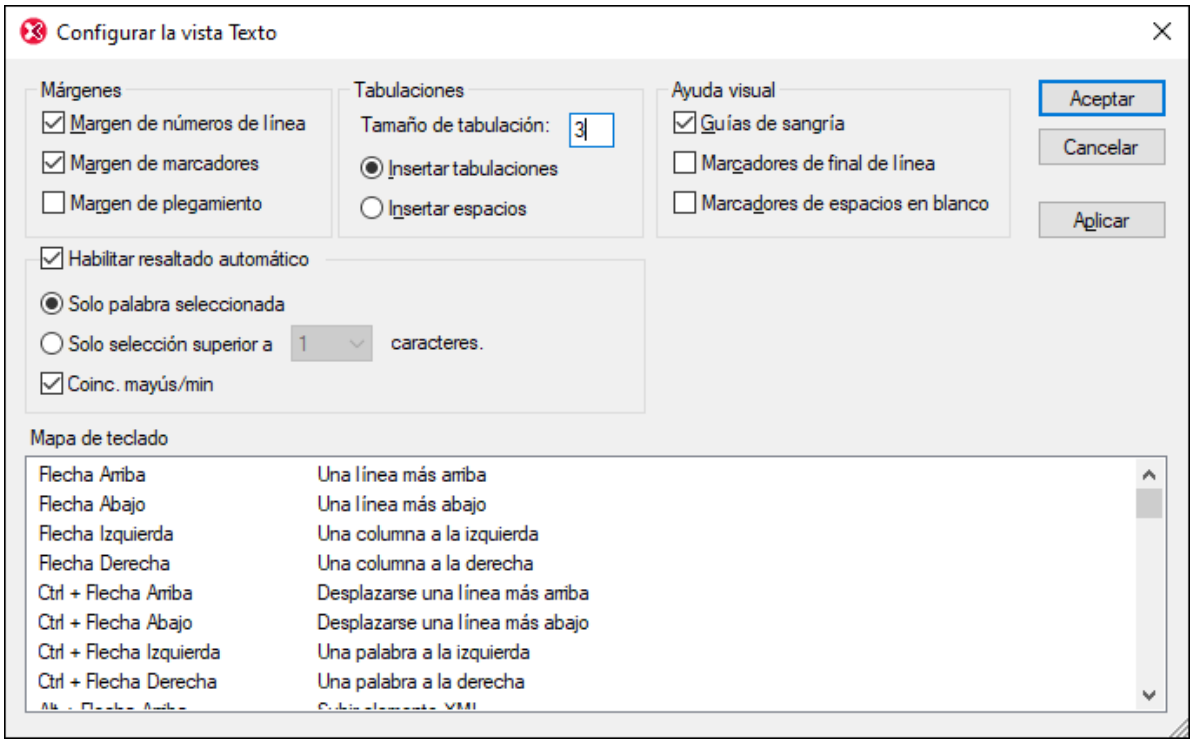
4. El panel *Tabulaciones* reduzca el tamaño de tabulación a 3. Como resultado la indentación del formato pretty-print[146] será una tabulación (en lugar de espacios) y cada tabulación tendrá un ancho de tres espacios. Para confirmar haga clic en **Aceptar**.
5. Haga clic en el comando de menú **Edición | Pretty-Print**. Al documento se le aplica el formato pretty-print con los nuevos valores de tabulación.
6. Abra otra vez el cuadro de diálogo "Configurar la vista Texto" (**Vista | Configurar la vista Texto**) y en el panel *Ayuda visual* marque la casilla *Marcadores de final de línea*. Para confirmar haga clic en **Aceptar**.
7. En la vista Texto vaya al final de una línea y elimine el marcador de final de línea para que la línea siguiente suba a la línea anterior.
8. Cambie a la vista Cuadrícula y después vuelva a la vista Texto.
9. Haga clic en el comando de menú **Edición | Pretty-Print**. Al documento se le aplica el formato pretty-print y se restaura el marcador de final de línea que eliminó.

Nota: si desactiva la casilla *Aplicar sangría* de la sección *Vista* del cuadro de diálogo "Opciones" (**Herramientas | Opciones | Pretty-print**[1591]) y aplica el formato pretty-print, entonces las líneas empiezan sin sangría.

Marcadores

Los marcadores se insertan en un margen de marcadores, a la izquierda de la línea que desea marcar. Después puede recorrer el documento de marcador en marcador.

1. Abra el cuadro de diálogo "Configurar la vista Texto" (**Vista | Configurar la vista Texto**, *imagen anterior*) y marque la opción *Margen de marcadores* del panel *Márgenes*. Para confirmar haga clic en **Aceptar**.



2. En la vista Texto del archivo `CompanyFirst.xml`, ponga el cursor en la línea donde desea insertar un marcador y seleccione el comando **Edición | Insertar o quitar el marcador**. En la línea se introduce un marcador, lo cual se señala con una esfera azul en el margen de marcadores (*imagen siguiente*).
3. Cree un marcador en otra línea.

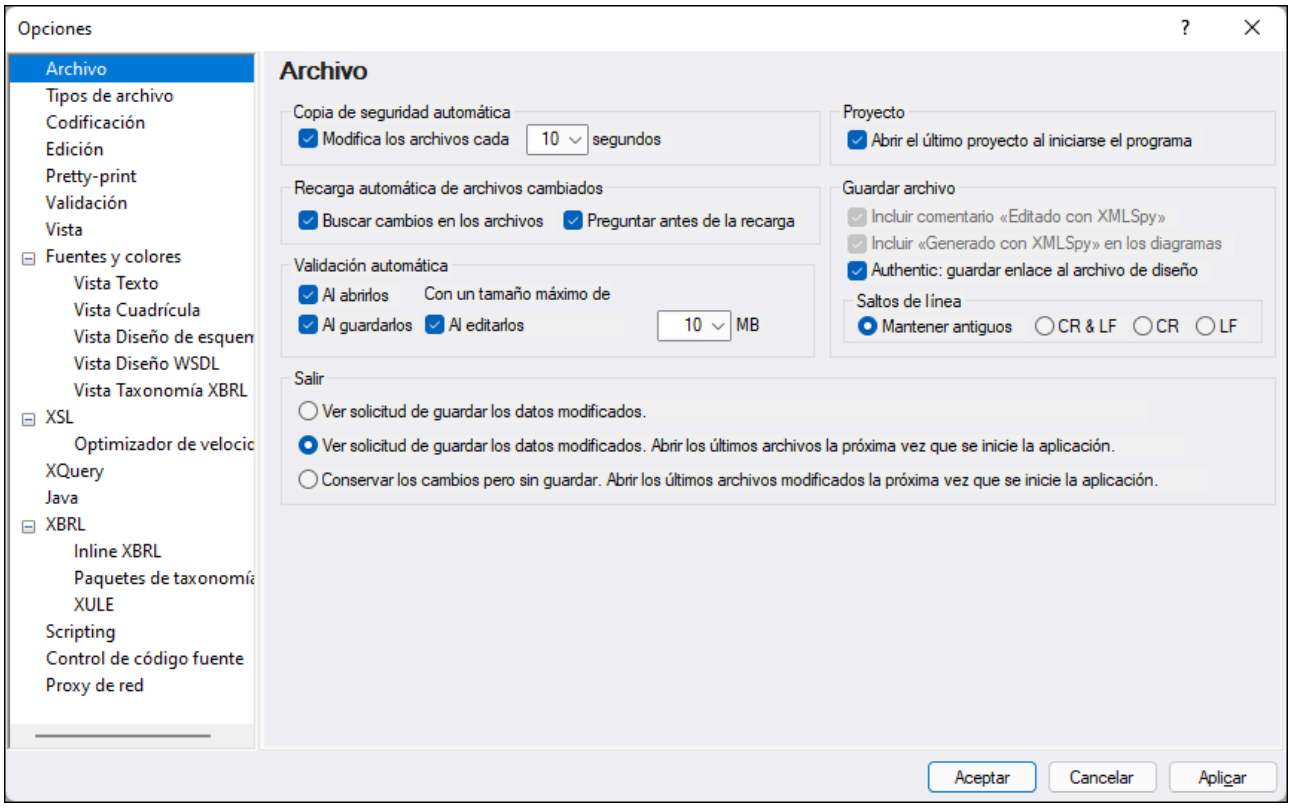
```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Company>
3     <Address xsi:type="US-Address">
4        <Name>US dependency</Name>
5        <Street>Noble Ave.</Street>
6        <City>Dallas</City>
7        <Zip>04812</Zip>
8        <State>Texas</State>
9     </Address>
10    <Person Manager="true" Degree="BA" Programmer="false">
11       <First>Fred</First>
12       <Last>Smith</Last>
13       <PhoneExt>22</PhoneExt>
14       <Email>Smith@work.com</Email>
15    </Person>
16 </Company>
17
```

4. Pulse F2 (o el comando **Edición | Ir al siguiente marcador**) para pasar al siguiente marcador. Pulse **Mayús+F2** (o el comando **Edición | Ir al marcador anterior**) para pasar al marcador anterior.
5. Ponga el cursor en una de las líneas que tienen un marcador y seleccione el comando **Edición | Insertar o quitar marcador**. El marcador se elimina.
6. Guarde el archivo y ciérrelo. Los marcadores no se guardan en el archivo. Abra el archivo otra vez para comprobarlo.

© 2016-2022 Altova GmbH

2.1.5 Opciones de la aplicación

XMLSpy cuenta con tantas funciones que la configuración de ciertas opciones puede influir considerablemente en su forma de trabajar con la aplicación. A largo plazo es recomendable conocer a fondo estas opciones para poder configurarlas de forma que se adapten a los requisitos de su trabajo y su manera de trabajar. Puede acceder a estas opciones desde el cuadro de diálogo "Opciones" (*imagen siguiente*), que se abre con el comando de menú **Herramientas | Opciones**[1583].



Estas opciones vienen organizadas por secciones, que puede ver en el panel de navegación izquierdo. Recomendamos que eche un vistazo a todas las secciones para hacerse una idea de las opciones que puede usar. A continuación explicamos las opciones que afectan a las funciones más usadas. Para ver las descripciones de todas las opciones consulte la documentación del cuadro de diálogo "Opciones"[1583]. Cuando haya terminado de configurar las opciones, haga clic en **Aceptar** para guardar los cambios y cerrar el cuadro de diálogo. El botón **Aplicar** sirve para que esos cambios se reflejen en los documentos que estén abiertos.

Archivo
Puede habilitar o deshabilitar la copia de seguridad automática (de los archivos que se estén editando). *Validar al editar* permite validar documentos o comprobar si su formato es correcto mientras escribe. Si esta opción le resulta molesta puede desactivarla y comprobar el formato y validar el documento cuando usted prefiera.

Tipos de archivo
Aquí puede seleccionar la vista que prefiera para cada tipo de archivo y establecerla como predeterminada para ese tipo de archivo.

Vista
En la barra de título de XMLSpy o título de la pantalla puede mostrar el nombre del archivo solamente o toda su ruta de acceso. Recuerde que puede ver la ruta de acceso si pasa el cursor del ratón por encima del nombre del archivo en la pestaña de ese archivo, en la parte inferior de la ventana principal.

Fuentes y colores
Puede definir la fuente, el tamaño de fuente y el color del texto y de otros componentes. Cada vista tiene unas opciones propias.

© 2016-2022 Altova GmbH

2.2 Esquemas XML: aspectos básicos

Un esquema XML describe la estructura de un documento XML. Un documento XML se puede validar con un documento XML para comprobar si se ajusta a los requisitos especificados en el esquema. Si es así, se dice que el documento es **válido**. Si no, se dice que **no es válido**. Los esquemas XML permiten a los diseñadores de documentos especificar qué estructura y contenido puede tener un documento XML y comprobar si el documento XML es válido.

La estructura y la sintaxis del documento de esquema XML es compleja y, al ser también un documento XML, el esquema XML debe ser válido con respecto a las reglas de la especificación XML Schema. En XMLSpy la vista Esquema permite construir esquemas XML válidos con facilidad, mediante operaciones de arrastrar y colocar. El documento de esquema XML que construya se puede editar también en las vistas Texto y Cuadrícula, pero es más fácil crearlo y modificarlo en la vista Esquema.

Objetivo

En esta sección del tutorial aprenderá a editar esquemas XML en la vista Esquema. En concreto, aprenderá a:

- crear un archivo de esquema nuevo
- definir espacios de nombres para el esquema
- definir un modelo de contenido básico
- agregar elementos al modelo de contenido con ayuda de los menús contextuales y de operaciones de arrastrar y colocar
- configurar la vista del modelo de contenido

Cuando termine de crear el esquema básico podrá pasar a la siguiente sección del tutorial 67, en la que aprenderá a trabajar con funciones más avanzadas de XMLSpy para esquemas XML. Tras esta sección sobre funciones avanzadas sigue una sección sobre cómo navegar por esquemas y documentarlos 81 en XMLSpy.

Comandos utilizados en esta sección

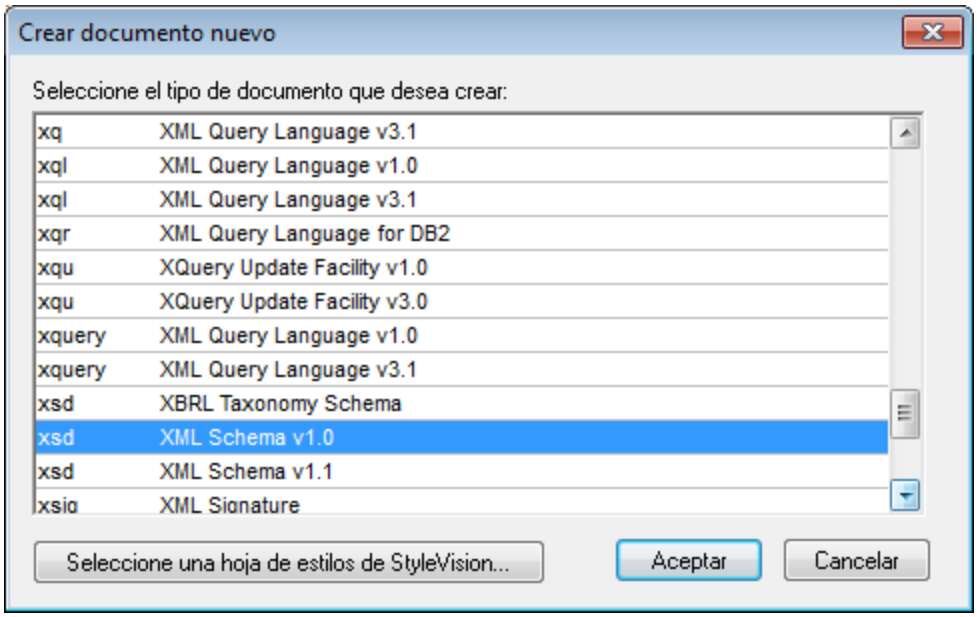
En esta sección del tutorial utilizará únicamente la vista Esquema. Además, utilizará estos comandos:

	**Mostrar diagrama** (o "mostrar vista del modelo de contenido"). Este icono aparece a la izquierda de todos los componentes globales en la vista general del esquema. Pulse este icono para ver el modelo de contenido del componente global correspondiente.

2.2.1 Crear un archivo de esquema XML nuevo

Para crear un archivo de esquema XML nuevo:

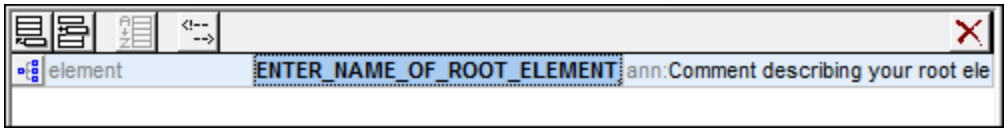
1. Seleccione la opción de menú **Archivo | Nuevo**. Aparece el cuadro de diálogo "Crear documento nuevo".



2. En el cuadro de diálogo, seleccione la opción `XSD (XML Schema v1.0)` (nota: la descripción del documento y la lista del cuadro de diálogo puede ser diferente a la que aparece en esta imagen) y haga clic en **Aceptar**. A continuación aparecerá un archivo de esquema vacío en la vista Esquema (ver imagen siguiente) de la ventana Principal.
3. En la barra de herramientas Diseño de esquemas pulse el botón del modo **XSD 1.0** (*imagen siguiente*) para que la vista Esquema esté en modo de edición XSD 1.0.

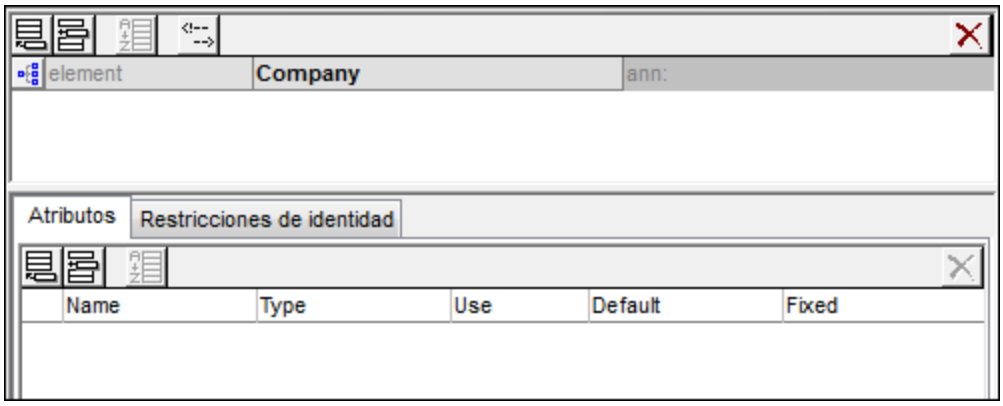


4. Se le pedirá que introduzca el nombre del elemento raíz.



5. Haga doble clic en el campo resaltado y escriba Company. Pulse la tecla Entrar para confirmar. Ahora Company es el elemento raíz de este esquema y se crea como elemento global. La vista que puede ver en la ventana Principal (*ver imagen siguiente*) se llama "vista general del esquema". Esta vista ofrece una visión de conjunto del esquema, mostrando una lista de todos los componentes globales en el panel superior de la ventana principal. En el panel inferior aparecen los atributos y restricciones de identidad del componente global seleccionado (Para ver y editar el modelo de contenido de cada componente global basta con pulsar el icono "Mostrar diagrama" situado a la izquierda de cada componente global.)

© 2016-2022 Altova GmbH



6. En el campo `ann` (anotaciones) del elemento `Company`, escriba una descripción del elemento, es decir, `Root element` (elemento raíz)
7. Haga clic en la opción de menú **Archivo | Guardar** y guarde su esquema XML con el nombre que prefiera (`AddressFirst.xsd`, por ejemplo).

Los círculos de colores de las pestañas indican el estado de la copia de seguridad del archivo correspondiente. Para más información sobre estos indicadores consulte Copia de seguridad automática de archivos 143.

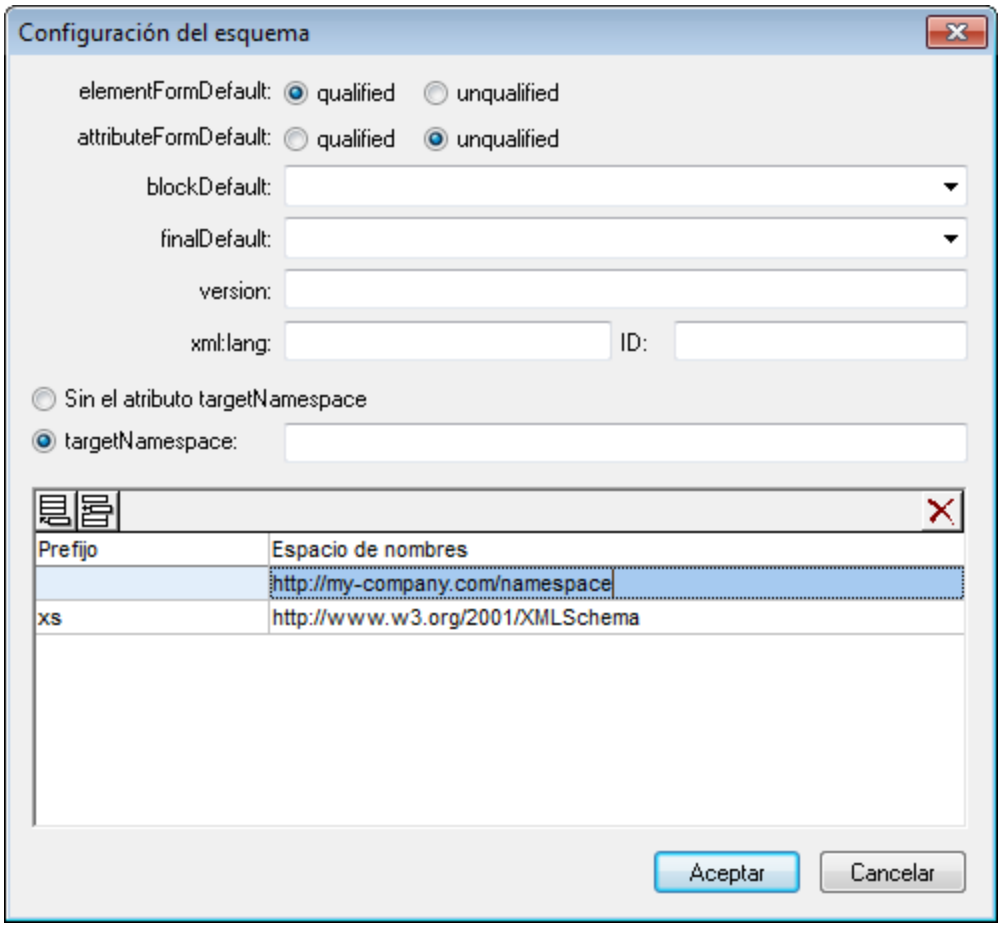
2.2.2 Definir espacios de nombres

Los espacios de nombres XML son un aspecto importante de los documentos XML y los esquemas XML. Un documento de esquema XML debe hacer referencia al espacio de nombres del esquema XML y, de manera opcional, puede definir un espacio de nombres de destino para la instancia de documento XML. Como diseñador del esquema, usted debe decidir cómo definir estos espacios de nombres (es decir, con qué prefijos).

En el esquema XML que está creando, definirá un espacio de nombres de destino para las instancias de documento XML. (Cuando se crea un documento de esquema XML nuevo, XMLSpy crea automáticamente la referencia necesaria al espacio de nombres del esquema XML.)

Para crear un espacio de nombres de destino:

1. Seleccione la opción de menú **Diseño de esquemas | Configuración del esquema**. A continuación se abre el cuadro de diálogo "Configuración del esquema" (*imagen siguiente*).



2. Haga clic en el botón de opción **targetNamespace** y escriba `http://my-company.com/namespace`. En XMLSpy, el espacio de nombres que escriba como espacio de nombres de destino se crea como espacio de nombres predeterminado del documento de esquema XML y aparece en la lista de espacios de nombres situado en el panel inferior del cuadro de diálogo.
3. Haga clic en **Aceptar**.

Tenga en cuenta que:

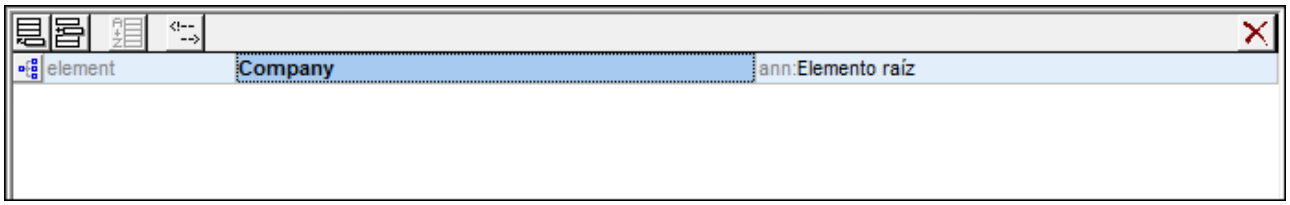
- XMLSpy crea automáticamente el espacio de nombres del esquema XML, al que se le da el prefijo `xs:`.
- cuando se crea la instancia de documento XML, debe tener definido su espacio de nombres de destino en el esquema XML. De lo contrario, el documento XML no será válido.

2.2.3 Definir un modelo de contenido

En la vista general del esquema, ya creó un elemento global llamado `Company`. Este elemento contendrá un elemento llamado `Address` y un número ilimitado de elementos llamados `Person` (que forman su modelo de contenido). Los componentes globales que pueden tener modelos de contenido son los elementos, los tipos complejos y los grupos de elementos.

© 2016-2022 Altova GmbH

En XMLSpy el modelo de contenido de un componente global se muestra en la vista del modelo de contenido de la vista Esquema (*imagen siguiente*). Para ver y editar el modelo de contenido de un componente global, pulse el icono "Mostrar diagrama" situado a la izquierda del componente global.

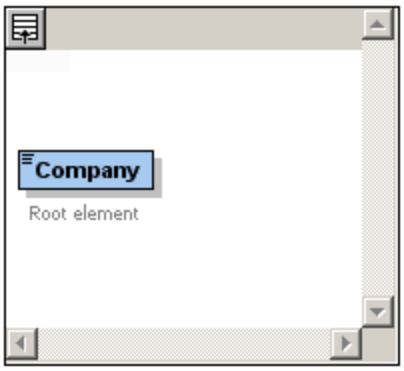


En esta sección creará el modelo de contenido del elemento `Company`.

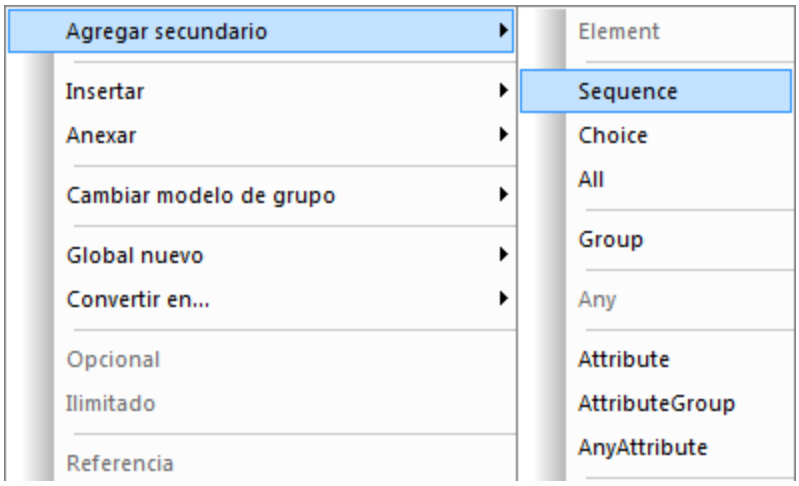
Crear un modelo de contenido básico

Para crear el modelo de contenido del elemento `Company`:

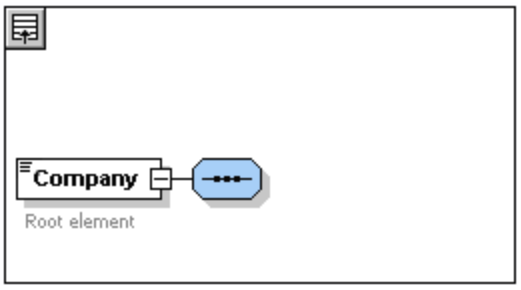
1. En la vista general del esquema, pulse el icono "Mostrar diagrama" situado a la izquierda del elemento `Company`. A continuación aparecerá el modelo de contenido del elemento `Company`, que por ahora está vacío (*imagen siguiente*). Otra opción es hacer doble clic en el nodo `Company` en el ayudante de entrada "Componentes" y aparecerá el modelo de contenido.



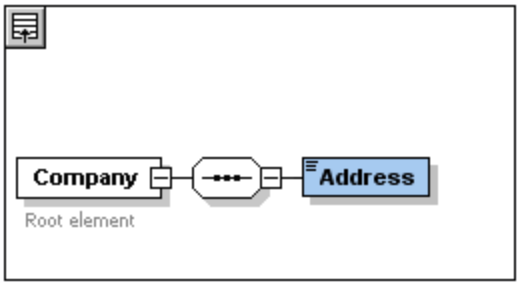
2. Un modelo de contenido está formado por **compositores** y **componentes**. Los compositores indican la relación que existe entre dos componentes. Para poder añadir un elemento secundario en el modelo de contenido de `Company` antes debe añadir un compositor. Para agregar un compositor, haga clic con el botón derecho en el elemento `Company`. En el menú contextual que aparece, seleccione **Agregar secundario | Sequence**. ("Sequence", "Choice" y "All" son los tres compositores que se pueden utilizar en un modelo de contenido.)



A continuación aparece un compositor de secuencia, que requiere que los componentes del grupo aparezcan en la secuencia indicada.

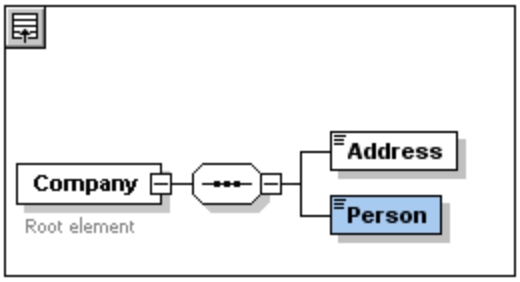


3. Haga clic con el botón derecho en el compositor de secuencia y seleccione **Agregar secundario | Element**. A continuación se añade un componente "element" sin nombre.
4. Escriba el nombre `Address` en el elemento y pulse la tecla **Entrar** para confirmar.



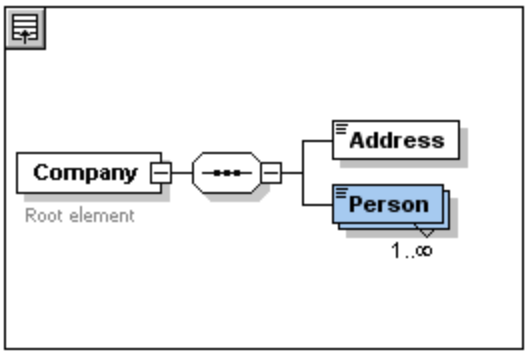
5. Haga clic con el botón derecho en el compositor de secuencia y seleccione **Agregar secundario | Element**. Ahora el elemento `Person` del diagrama muestra el número de repeticiones permitidas: de 1 a infinito.

 © 2016-2022 Altova GmbH



Hasta ahora ha definido un esquema que acepta una dirección (Address) y una persona (Person) por compañía (Company). Ahora debe incrementar el número de elementos `Person`.

6. Haga clic con el botón derecho en el elemento `Person` y seleccione la opción Ilimitado (**unbounded**) del menú contextual. Ahora el elemento `Person` del diagrama muestra el número de repeticiones permitidas: de 1 a infinito.



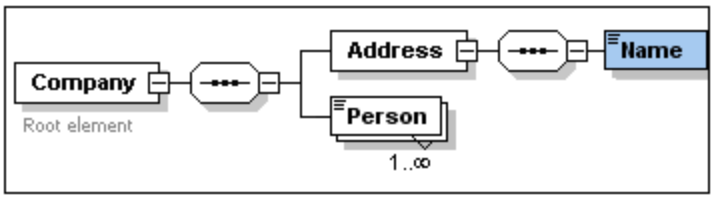
Otra opción es editar los campos `minOcc` y `maxOcc` del ayudante de entrada "Detalles" e indicar el número de repeticiones permitidas, en este caso "1" y "unbounded" respectivamente.

Agregar más niveles a la estructura del modelo de contenido

El modelo de contenido básico que ha creado contiene por ahora un solo nivel: un nivel de elementos secundarios para el elemento `Company` en el que están los elementos secundarios `Address` y Person. A continuación definiremos el contenido del elemento `Address` para que contenga los elementos `Name`, `Street` y `City`. Este será el segundo nivel del modelo de contenido. Para ello debemos añadir un compositor en el elemento `Address` y después los componentes que acabamos de mencionar.

Para ello, siga estos pasos:

1. Haga clic con el botón derecho en el elemento `Address` para abrir el menú contextual y seleccione la opción **Agregar secundario | Sequence**. Esta acción añade el compositor de secuencia.
2. Haga clic con el botón derecho en el compositor de secuencia y seleccione **Agregar secundario | Element**. Ponga el nombre `Name` al componente "element" recién creado.



Tipos complejos, tipos simples y tipos de datos de XML Schema

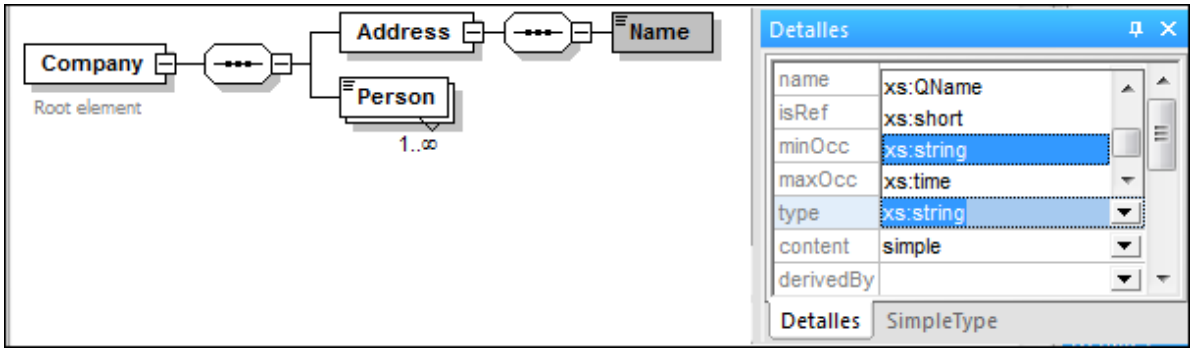
Por ahora no hemos definido ningún tipo de elemento. Haga clic en la pestaña **Texto** para cambiar a la vista Texto del esquema (*ver el código que aparece más adelante*). Observará que cada vez que insertó un compositor de secuencia, el elemento `xs:sequence` se insertó dentro del elemento `xs:complexType`. Es decir, los elementos `Company` y `Address` son tipos complejos porque contienen elementos secundarios. Los elementos de tipo complejo son aquellos que contienen atributos o elementos.

```
<xs:element name="Company">
  <xs:annotation>
    <xs:documentation>Root element</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Address">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Name"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Person"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Los elementos de tipo simple, por el contrario, sólo contienen texto y no tienen atributos. El texto puede ser una cadena de texto, una fecha, números, etc. En nuestro ejemplo queremos que el elemento secundario `Name` de `Address` contenga sólo texto. Es un tipo simple cuyo texto queremos restringir a una cadena de texto. Para ello nos serviremos del tipo de datos de XML Schema `xs:string`.

Para definir el tipo de datos del elemento `Name`:

1. Haga clic en la pestaña **Esquema** para volver a la vista Esquema.
2. Haga clic en el elemento `Name` para seleccionarlo.
3. En el ayudante de entrada "Detalles" seleccione la opción `xs:string` del menú desplegable de `type`.



Observe que tanto `minOcc` como `maxOcc` tienen el valor 1, lo cual indica que este elemento aparece una sola vez.

La representación textual del elemento `Name` es:

 © 2016-2022 Altova GmbH

```
<xs:element name="Name" type="xs:string"/>
```

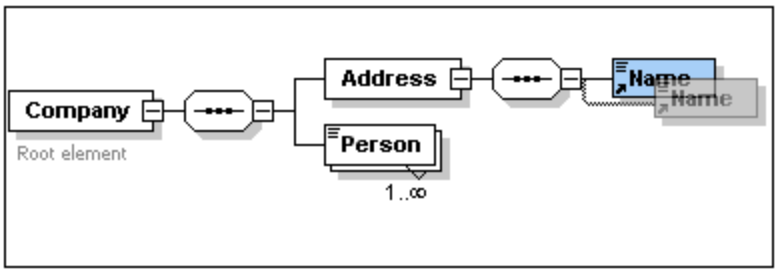
Nota: los elementos de tipo simple pueden tener uno de los muchos tipos de datos de XML Schema. En todos los casos, en el recuadro del elemento aparecerá un icono que indica que el contenido es sólo texto.

2.2.4 Agregar elementos mediante operaciones de arrastrar y colocar

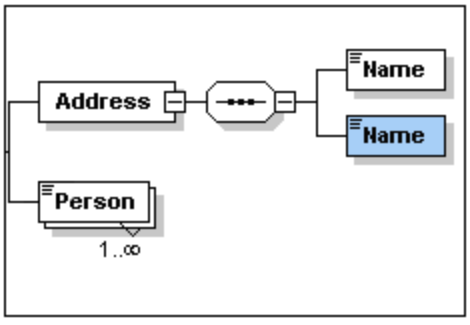
Hasta ahora añadió los elementos con ayuda del menú contextual que aparece al hacer clic con el botón derecho en un elemento o compositor. También puede crear elementos mediante operaciones de arrastrar y colocar, lo cual es mucho más rápido. En esta sección, agregará más elementos a la definición del elemento `Address` mediante operaciones de arrastrar y colocar y terminará esta definición.

Para terminar la definición del elemento `Address` mediante operaciones de arrastrar y colocar:

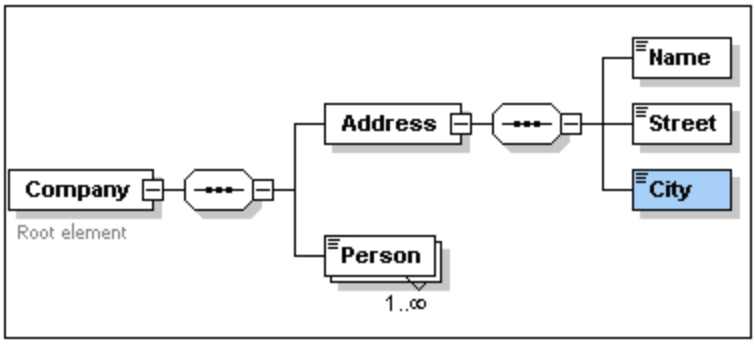
1. Haga clic en el elemento `Name` del elemento `Address`, mantenga pulsada la tecla **Ctrl** y arrastre el recuadro del elemento con el ratón. En el recuadro del elemento aparecerá un pequeño icono con el signo +, que indica que se dispone a copiar el elemento. También aparece una copia del elemento y una línea de conexión, que muestra dónde se creará el elemento.



2. Cuando suelte el botón del ratón se creará el elemento nuevo en la secuencia de `Address`. Si el elemento nuevo no se crea en la posición deseada, colóquelo justo debajo del elemento `Name`.



3. Para cambiar el nombre del elemento, haga doble clic en el recuadro del elemento y escriba `Street`.
4. Utilice el mismo método para crear otro elemento llamado `City`. El modelo de contenido debe ser este:



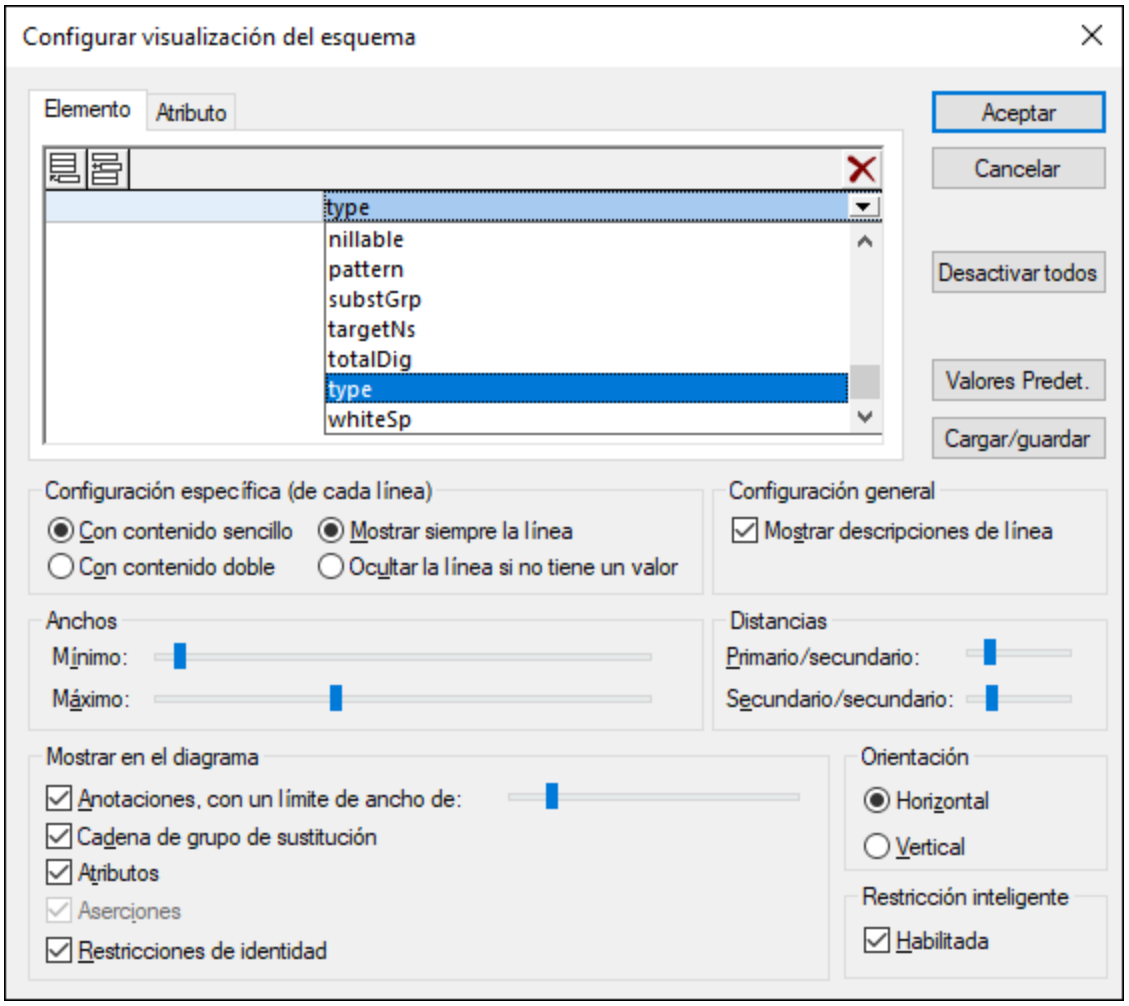
El elemento `Address` tiene una secuencia de elementos `Name`, `Street` y `City`, en ese orden.

2.2.5 Configurar la vista del modelo de contenido

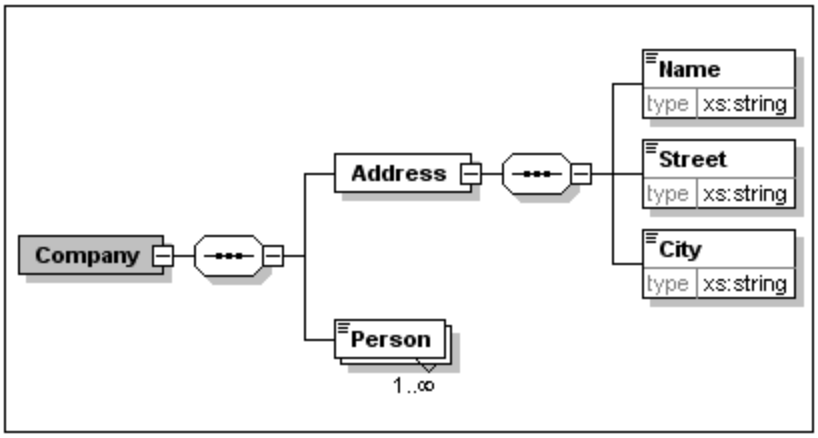
Ahora es el momento ideal para configurar la vista del modelo de contenido. Vamos a configurar la vista de modo que aparezca el tipo de un elemento en el campo correspondiente. Para ello siga estas instrucciones:

1. Pulse el icono de un componente para abrir la vista del modelo de contenido. De lo contrario no estará habilitado el comando para configurar la vista.
2. En la vista del modelo de contenido seleccione el comando **Diseño de esquemas | Configurar visualización del esquema** para ver el cuadro de diálogo "Configurar visualización del esquema".

© 2016-2022 Altova GmbH

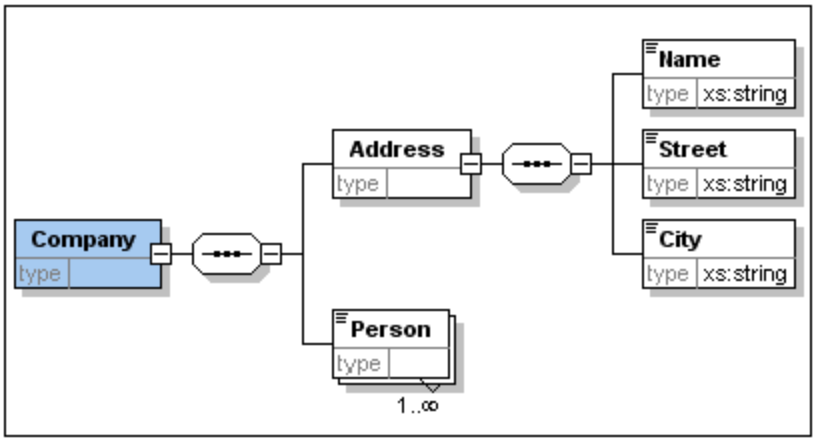


3. En la pestaña *Elemento* pulse el icono **Anexar** y seleccione `type` (*imagen anterior*) para agregar este descriptor de propiedad para cada recuadro de elemento.
4. En el panel "Configuración de línea única", marque la opción *Ocultar la línea si no tiene un valor*. Esto hará que se oculte la descripción del tipo de datos si el elemento no tiene ningún tipo de datos (por ejemplo, si el elemento es de tipo complejo).



Observe que el descriptor de tipo aparece en los elementos `Name`, `Street` y `City`, que son tipos simples del tipo `xs:string`, pero no aparece en los elementos de tipo complejo. Esto es así porque se seleccionó la opción *Ocultar la línea si no tiene un valor*.

6. En el panel "Configuración de línea única" marque la opción *Mostrar siempre la línea*.
7. Para terminar haga clic en **Aceptar**.



Observe que la línea del descriptor del tipo de datos aparece siempre, incluso en los recuadros de elementos de tipo complejo en los que no tiene ningún valor.

Recuerde que:

- las líneas del descriptor de propiedad se pueden editar, de modo que los valores que introduzca en la línea formarán parte de la definición del elemento.
- las opciones de configuración elegidas en el cuadro de diálogo "Configurar visualización del esquema" se aplicarán tanto a la documentación del esquema como a su versión impresa.

2.2.6 Terminar un esquema básico

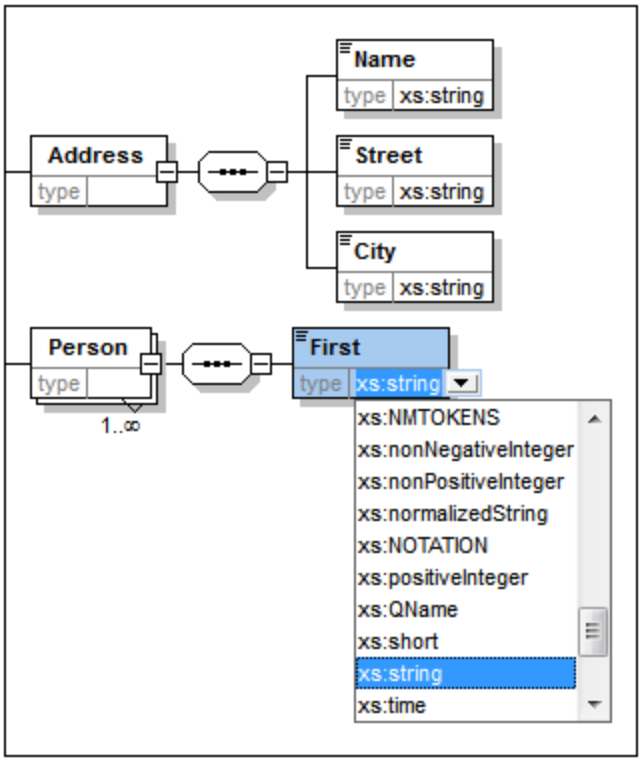
En las secciones anteriores definió el contenido del elemento `Address`. Ahora necesita definir el contenido del elemento `Person`. El elemento `Person` debe contener estos elementos secundarios de tipo simple: `First`, `Last`, `Title`, `PhoneExt` y `Email`. Todos estos elementos son de uso obligatorio, excepto `Title` y todos deben

 © 2016-2022 Altova GmbH

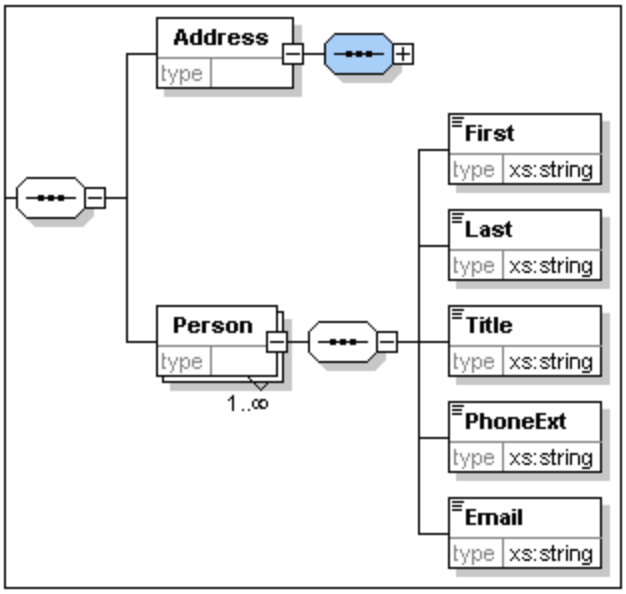
aparecer en ese orden. Todos deben tener el tipo `xs:string` excepto `PhoneExt`, que debe tener el tipo `xs:integer` con un máximo de dos dígitos.

Para crear el modelo de contenido del elemento `Person`:

1. Haga clic con el botón derecho en el elemento `Address` para abrir el menú contextual y seleccione la opción **Agregar secundario | Sequence**. Aparece el compositor de secuencia.
2. Haga clic con el botón derecho en el compositor de secuencia y seleccione **Agregar secundario | Element**.
3. Ponga el nombre `First` al elemento y pulse la tecla **Tabulador**. El cursor pasa automáticamente al campo `type`.



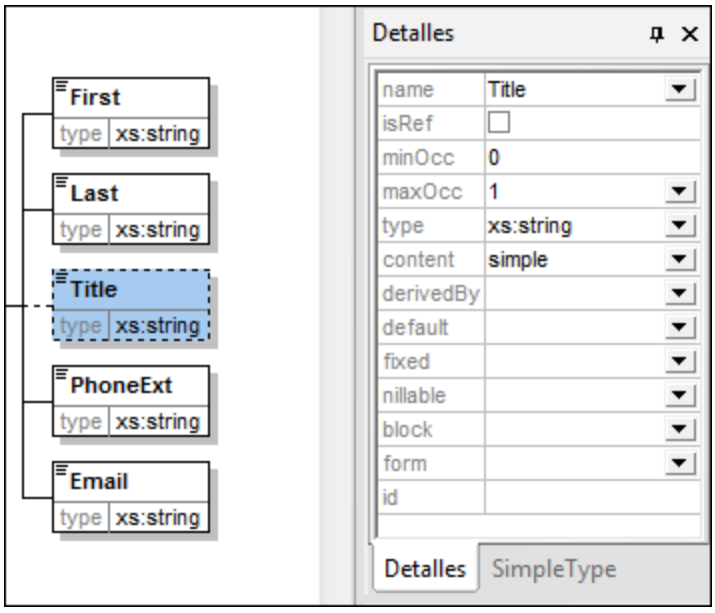
4. Seleccione la opción `xs:string` del menú desplegable o escriba `xs:string` en el campo Type.
5. Cree cuatro elementos más mediante operaciones de arrastrar y colocar. Llámeles `Last`, `Title`, `PhoneExt` y `Email` respectivamente.



Nota: puede seleccionar varios elementos a la vez si mantiene pulsada la tecla **Ctrl** y hace clic en cada uno de los elementos. Esto permite copiar varios elementos a la vez, por ejemplo.

Convertir un elemento en elemento opcional

Haga clic con el botón derecho en el elemento sequence y elija Agregar secundario **| Element** en el menú contextual. El recuadro del elemento pasa de tener un borde sólido a tener un borde discontinuo.



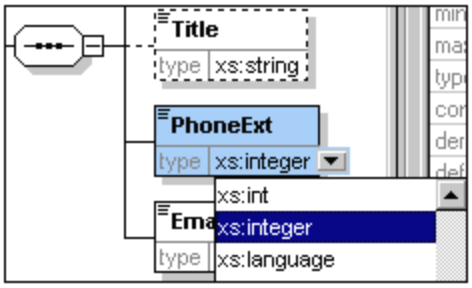
En el ayudante de entrada "Detalles" observará que los campos `minOcc=0` y `maxOcc=1` indican que el elemento es opcional. Es decir, en lugar de usar el menú contextual para convertir un elemento en elemento opcional, puede dar el valor `0` al atributo minOcc en el ayudante de entrada "Detalles".

 © 2016-2022 Altova GmbH

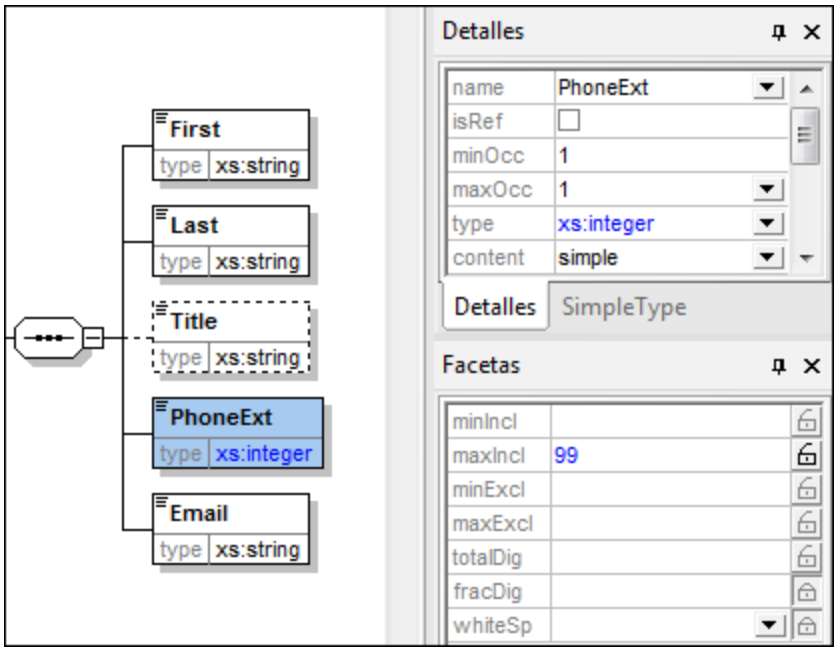
Limitar el contenido de un elemento

Para definir el elemento `PhoneExt` como elemento de tipo `xs:integer` y con un máximo de dos dígitos:

1. Haga doble clic en el campo `type` del elemento `PhoneExt` y seleccione (o escriba) la opción `xs:integer` del menú desplegable (*imagen siguiente*).



   Observe que en este momento cambian las opciones del ayudante de entrada "Facetas".
2. En el ayudante de entrada "Facetas" haga doble clic en el campo `maxIncl` y escriba `99`. Pulse la tecla **Entrar** para confirmar.



   Esto define que las extensiones de teléfono pueden tener valores desde 0 hasta 99.
3. Seleccione **Archivo | Guardar** para guardar los cambios realizados en el esquema.

Tenga en cuenta que:

- al seleccionar un tipo de datos de XML Schema que sea un tipo simple (por ejemplo, `xs:string` o `xs:date`), cambia automáticamente el modelo de contenido en el ayudante de entrada "Detalles" (`content` = simple).
- al agregar un compositor a un elemento (`sequence`, `choice` o `all`) cambia automáticamente el modelo de contenido en el ayudante de entrada "Detalles" (`content` = complex).

- el esquema descrito en esta sección y en las anteriores es el archivo de esquema `AddressFirst.xsd` ubicado en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Tutorial` de XMLSpy.

© 2016-2022 Altova GmbH

2.3 Esquemas XML: aspectos avanzados

Ahora que sabe crear un esquema básico podemos seguir con algunos aspectos más avanzados del desarrollo de esquemas.

Objetivo

En esta sección aprenderá a:

- Trabajar con tipos complejos y tipos simples 67, que se pueden usar como tipos de los elementos del esquema.
- Crear elementos globales 75 y hacerles referencia desde otros elementos.
- Crear atributos 77 y sus propiedades, incluidos valores enumerados.

Empezaremos esta sección con el esquema básico **`AddressFirst.xsd`** que creó en la primera parte de este tutorial.

Comandos utilizados en esta sección

En esta sección del tutorial utilizará sólo la vista Esquema y estos comandos:

	**Mostrar diagrama** (o mostrar la vista del modelo de contenido). Este icono está situado a la izquierda de todos los componentes globales en la vista general del esquema. Al hacer clic en este icono, aparece la vista del modelo de contenido del componente global correspondiente.
	**Mostrar nodos globales**. Este icono está situado en la esquina superior izquierda de la vista del modelo de contenido. Al hacer clic en este icono, la vista cambia a la vista general del esquema, en la que aparecen todos los componentes globales.
	**Anexar**. Este icono está situado en la esquina superior izquierda de la vista general del esquema. Si hace clic en este icono, podrá agregar un componente global.

2.3.1 Tipos complejos y tipos simples

Después de definir el modelo de contenido de un elemento, puede decidir si desea volver a utilizar ese elemento en otra parte del esquema. Esto puede ocurrir, por ejemplo, si quiere definir un modelo de contenido para direcciones en EE UU y en Reino Unido. Algunos de los componentes de estos formatos son comunes, como la calle y la ciudad. Otros componentes, sin embargo, no lo son. Por tanto, una estrategia para solventar esas diferencias sería reutilizar loc componentes comunes en los dos modelos de contenido (EE UU y Reino Unido) y completar después cada modelo con los componentes específicos de este (como el código ZIP en EE UU y el código postal en Reino Unido). Para ello podemos crear los componentes comunes como un tipo complejo global (o crear cada uno de esos componentes comunes como elemento global) y reutilizar el tipo complejo global (o los elementos globales) en el modelo de contenido de cada tipo de dirección.

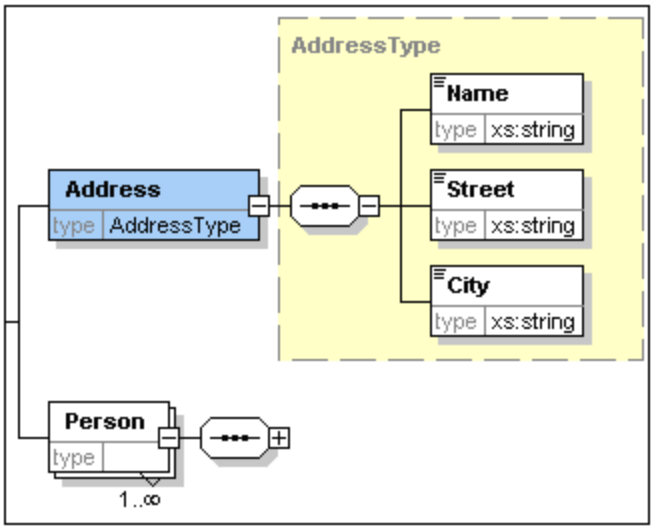
En esta sección aprenderá a trabajar con tipos complejos globales. Un tipo complejo es una definición de tipo para elementos que contienen otros elementos y/o atributos. Primero creará un tipo complejo en el nivel global y después lo extenderá para poder utilizarlo en un modelo de contenido. Más adelante aprenderá a trabajar con elementos globales.

Crear un tipo complejo global

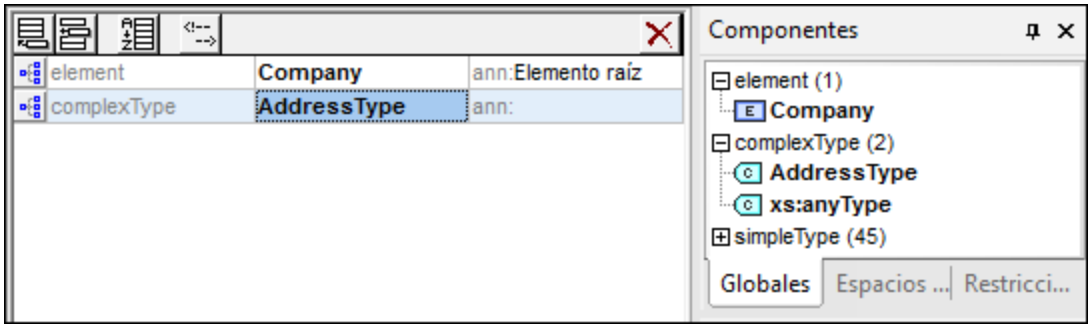
El elemento básico `Address` que definió en la sección anterior (que contiene los elementos `Name`, `Street` y `City`) puede volver a utilizarse pero con varios formatos. Por tanto, crearemos un tipo complejo a partir de esta definición de elemento y así podremos utilizarlo de nuevo.

Para crear un tipo complejo global:

1. En la vista del modelo de contenido, haga clic con el botón derecho en el elemento `Address`.
2. En el menú contextual seleccione **Convertir en... | complexType global**. Se crea un tipo complejo global llamado **`AddressType`** y se le asigna este tipo al elemento `Address` del modelo de contenido de **`Company`**. El contenido del elemento `Address` es el modelo de contenido de **`AddressType`**, que aparece dentro de un recuadro amarillo. Observe que ahora el tipo de datos del elemento `Address` es **`AddressType`**.

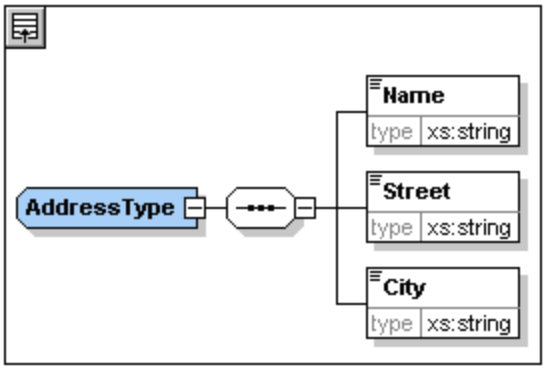


3. Haga clic en el icono "Mostrar nodos globales"  para volver a la vista general del esquema, donde podrá ver todos los componentes globales del esquema.
4. En el ayudante de entrada "Componentes" haga clic en los iconos de expansión de las opciones **`element`** y **`complexType`** para ver las construcciones de esquema correspondientes.
   La vista general del esquema ahora muestra dos componentes globales: el elemento `Company` y el tipo complejo `AddressType`. El ayudante de entrada "Componentes" también muestra el tipo complejo `AddressType`.



 © 2016-2022 Altova GmbH

5. Haga clic en el icono "Vista del modelo de contenido" de `AddressType` para ver su modelo de contenido (*ver imagen siguiente*). Observe qué forma tiene el contenedor del tipo complejo.



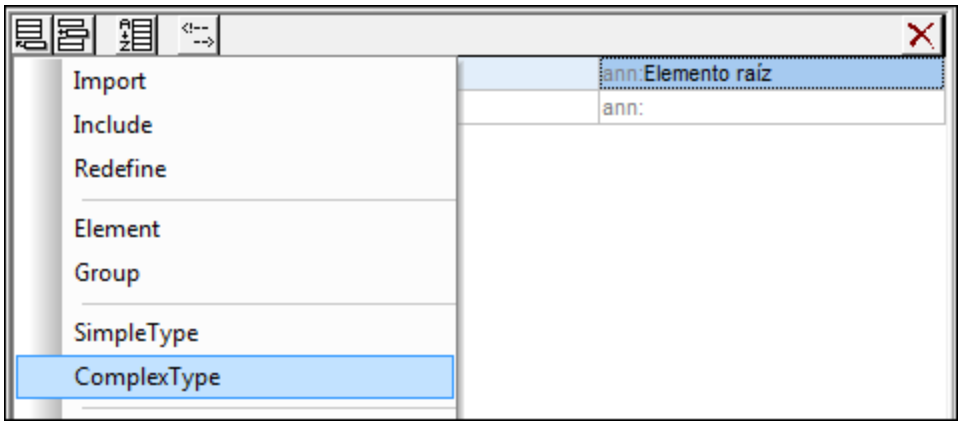
6. Haga clic en el icono "Mostrar nodos globales" para volver a la vista general del esquema.

Extender la definición de un tipo complejo

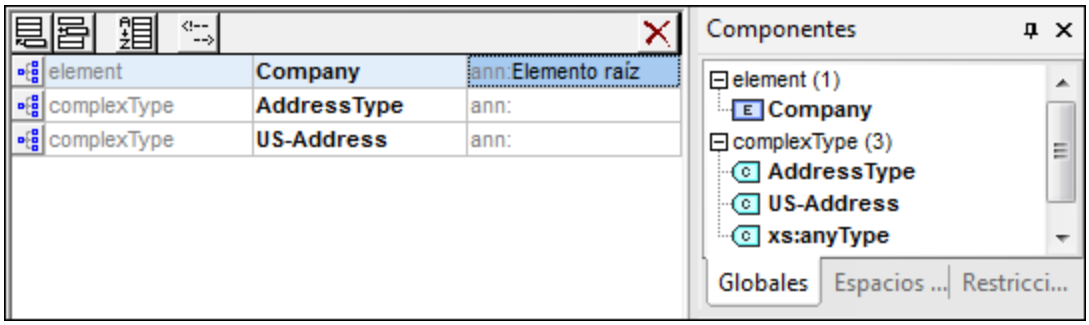
Ahora vamos a usar el componente global `AddressType` para crear dos tipos de direcciones para sendos países. Pero antes debemos definir un tipo complejo nuevo basado en el componente básico `AddressType` y después extenderemos esa definición.

Para ello siga estas instrucciones

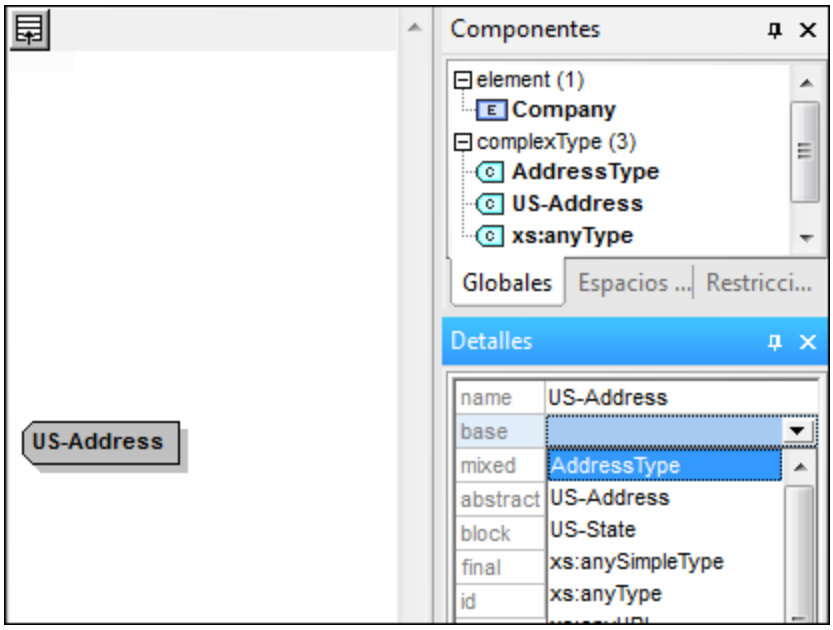
1. Cambie a la vista general del esquema. (Si se encuentra en la vista del modelo de contenido, haga clic en el icono "Mostrar nodos globales".)
2. Haga clic en el icono "Anexar" situado en la esquina superior izquierda de la vista. Aparecerá este menú:



3. Seleccione la opción **ComplexType**. Aparece una línea nueva en la lista de componentes y el cursor se coloca en el campo pertinente para que pueda escribir el nombre del componente.
4. Escriba `US-Address` y pulse la tecla **Entrar**. (Si en lugar de un guión escribe un espacio, el nombre del elemento aparecerá en color rojo. Esto indica que utilizó un carácter no válido.)

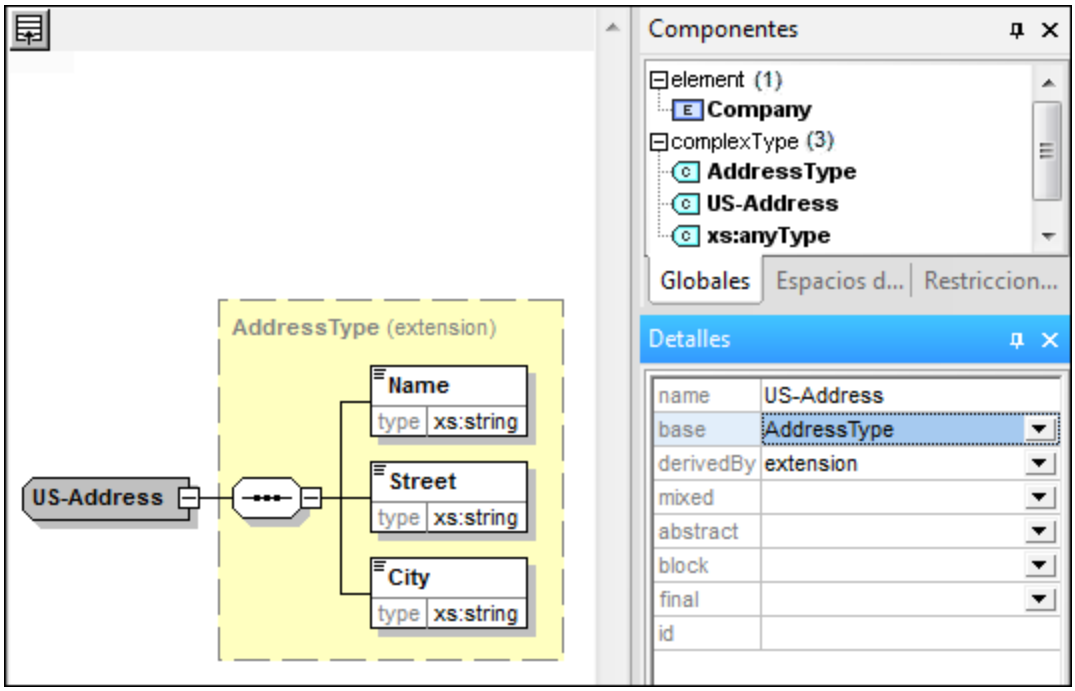


5. Haga clic en el icono "Vista del modelo de contenido" de `US-Address` para ver el modelo de contenido del tipo complejo nuevo. El modelo de contenido está vacío (*ver imagen siguiente*).
6. En el ayudante de entrada "Detalles", haga clic en el cuadro combinado `base` y seleccione la opción `AddressType`.

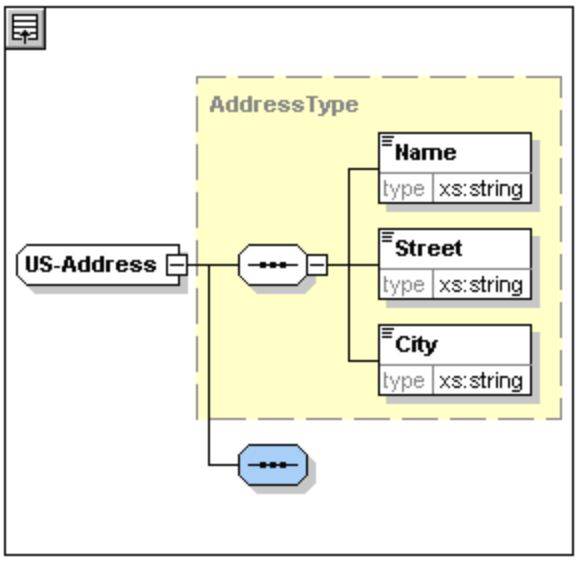


Ahora la vista del modelo de contenido muestra el modelo de contenido de `AddressType` como modelo de contenido de US-Address (*ver imagen siguiente*).

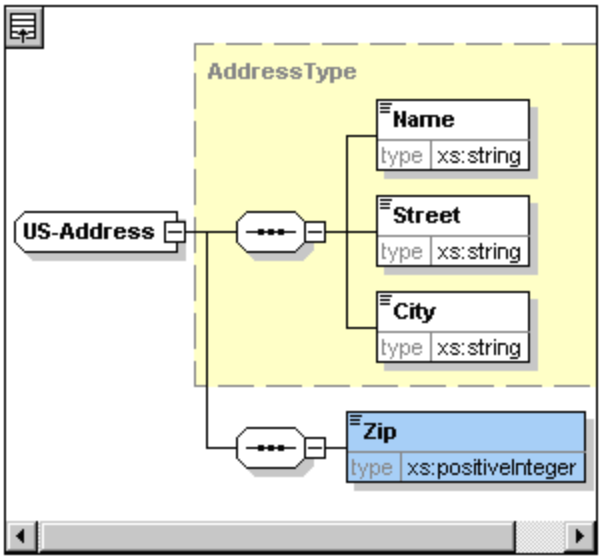
© 2016-2022 Altova GmbH



7. Ya podemos extender el modelo de contenido del tipo complejo **`US-Address`** con un elemento para el código postal. Para ello, haga clic con el botón derecho en el componente **`US-Address`** y en el menú contextual seleccione la opción **Agregar secundario | Sequence**. Aparece un compositor de secuencia nuevo fuera del recuadro de **`AddressType`** (*ver imagen siguiente*). Esto indica que se trata de una extensión del elemento.



8. Haga clic con el botón derecho en el compositor de clasificador secuencia y seleccione **Agregar secundario | Element**.
9. Llame a este elemento **`Zip`** y pulse la tecla **Tabulador**. El cursor aparecerá en el campo de valor del descriptor de tipo.
10. Seleccione la opción **`xs:positiveInteger`** del menú desplegable y pulse la tecla **Entrar** para confirmar.



Ahora tiene un tipo complejo llamado `US-Address`, que está basado en el tipo complejo `AddressType` y que se extiende con un elemento para el código postal.

Tipos simples globales

De la misma manera que el tipo complejo `US-Address` está basado en el tipo complejo `AddressType`, también puede basar elementos en un tipo simple. Las ventajas son las mismas que para los tipos complejos globales: el tipo simple se puede volver a utilizar. Para poder volver a utilizar un tipo simple, antes debe definirlo como tipo simple global. En este tutorial definirá como tipo simple el modelo de contenido de un elemento para los estados de EE UU. Este tipo simple se utilizará como base para otro elemento.

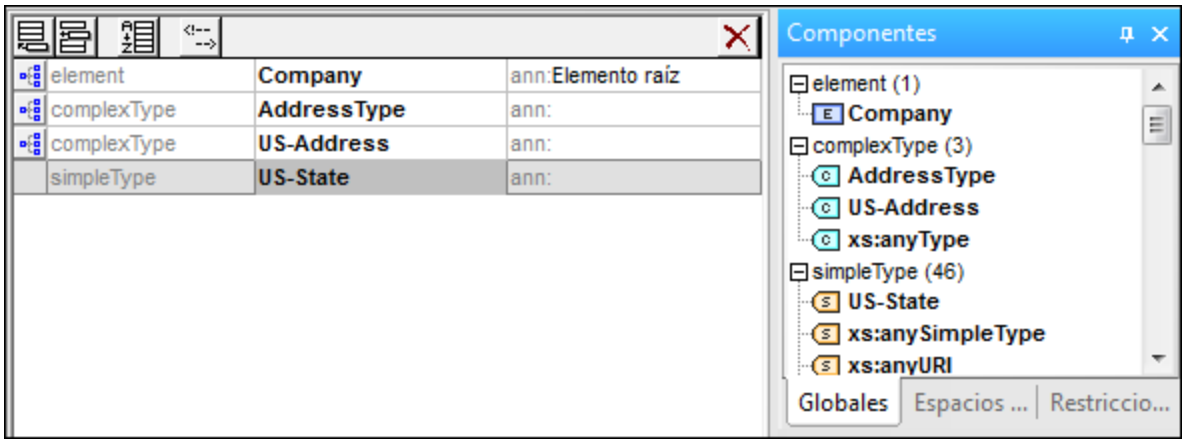
Crear un tipo simple global

Para crear un tipo simple global antes debe anexar un tipo simple nuevo a la lista de componentes globales, darle un nombre y definir su tipo de datos.

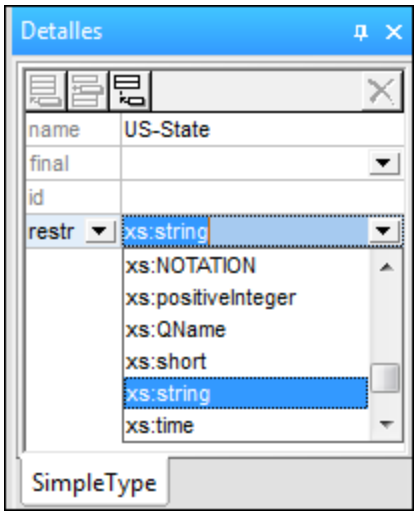
Para crear un tipo simple global:

1. Cambie a la vista general del esquema. (Si se encuentra en la vista del modelo de contenido, haga clic en el icono "Mostrar nodos globales" .)
2. Haga clic en el icono "Anexar" y seleccione la opción **SimpleType** del menú contextual que aparece.
3. Llame `US-State` a este tipo simple que acaba de crear.
4. Pulse la tecla **Entrar** para confirmar. Se crea el tipo simple `US-State` y aparece en la lista de tipos simples del ayudante de entrada "Componentes" (haga clic en el icono de expansión de simpleType para verlo).

© 2016-2022 Altova GmbH



5. En el ayudante de entrada "Detalles" (*ver imagen siguiente*), coloque el cursor en el campo de valor de `restr` y escriba `xs:string` o seleccione `xs:string` del menú desplegable del campo de valor de `restr`.



Esto crea un tipo simple llamado `US-State`, que tiene el tipo de datos `xs:string`. Este componente global ya se puede usar en el modelo de contenido de `US-Address`.

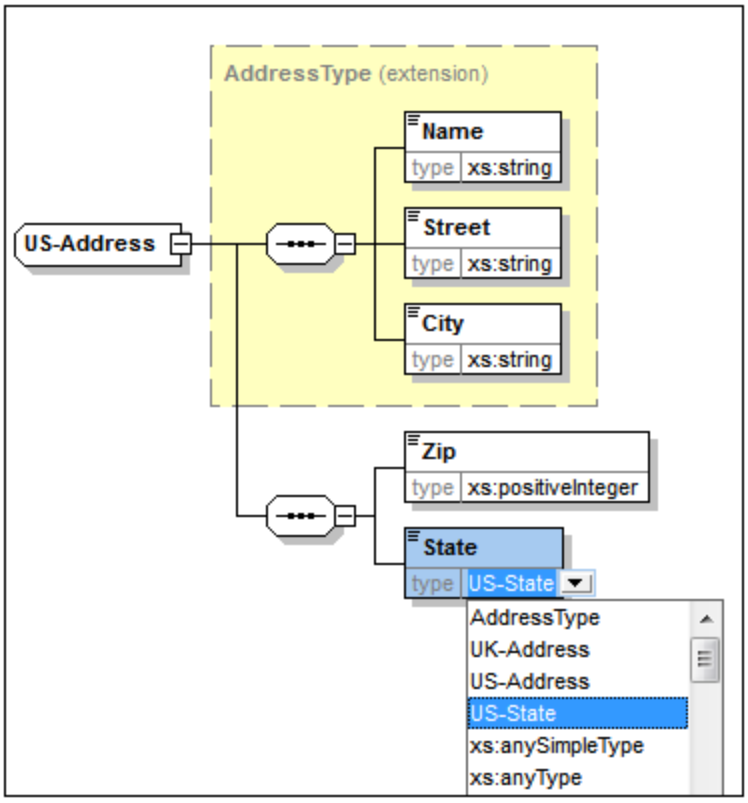
Usar un tipo simple global en un modelo de contenido

Puede usar un tipo simple global en un modelo de contenido para definir el tipo de un componente. En nuestro ejemplo usaremos `US-State` para definir un elemento llamado State en el modelo de contenido de `US-Address`.

Para ello haga lo siguiente:

1. En la vista general del esquema, haga clic en el icono "Vista del modelo de contenido" situado junto al componente `US-Address`.
2. Haga clic con el botón derecho en el compositor de secuencia situado en la parte inferior y seleccione **Agregar secundario | Element**.
3. Llame a este elemento nuevo `State`.
4. Pulse la tecla **Tabulador** y el cursor se coloca en el campo de valor del descriptor de tipo.

5. Seleccione la opción `US-State` en el menú desplegable de este cuadro combinado.



Ahora el elemento `State` está basado en el tipo simple `US-State`.

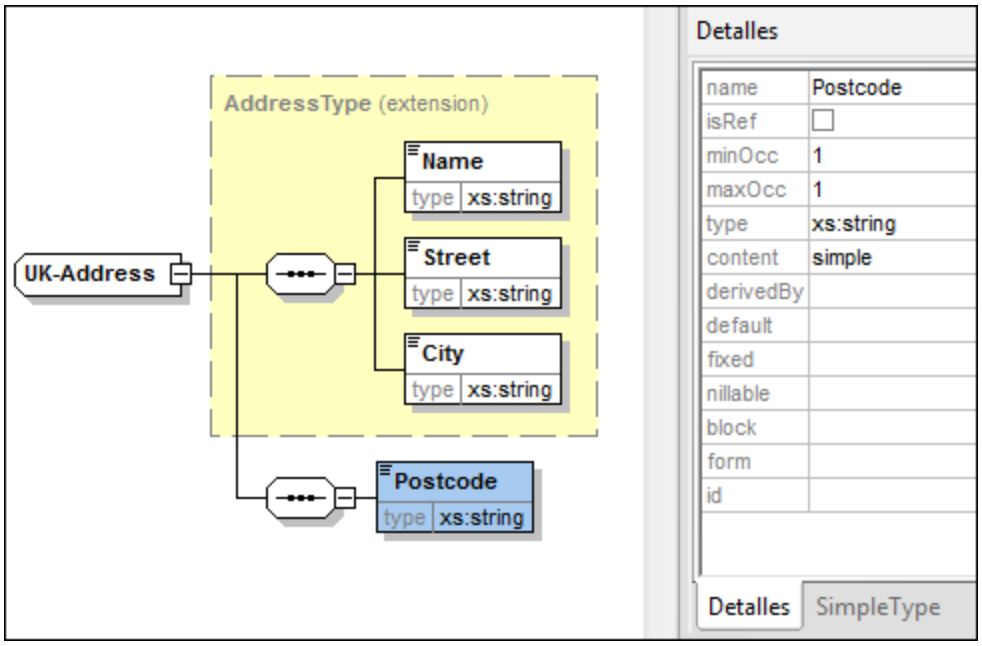
Crear otro tipo complejo basado en AddressType

Ahora vamos a crear un tipo complejo global que contendrá direcciones en el Reino Unido (UK). El tipo complejo se basa en `AddressType` y se extiende para ajustarse al formato de las direcciones postales del Reino Unido.

Para ello haga lo siguiente:

1. En la vista general del esquema cree un tipo complejo global llamado `UK-Address` que debe basarse en `AddressType` (`base=AddressType`).
2. En la vista del modelo de contenido de `UK-Address` añada un elemento llamado Postcode que debe ser de tipo `xs:string`. Así debe ser el modelo de contenido de `UK-Address`:

© 2016-2022 Altova GmbH

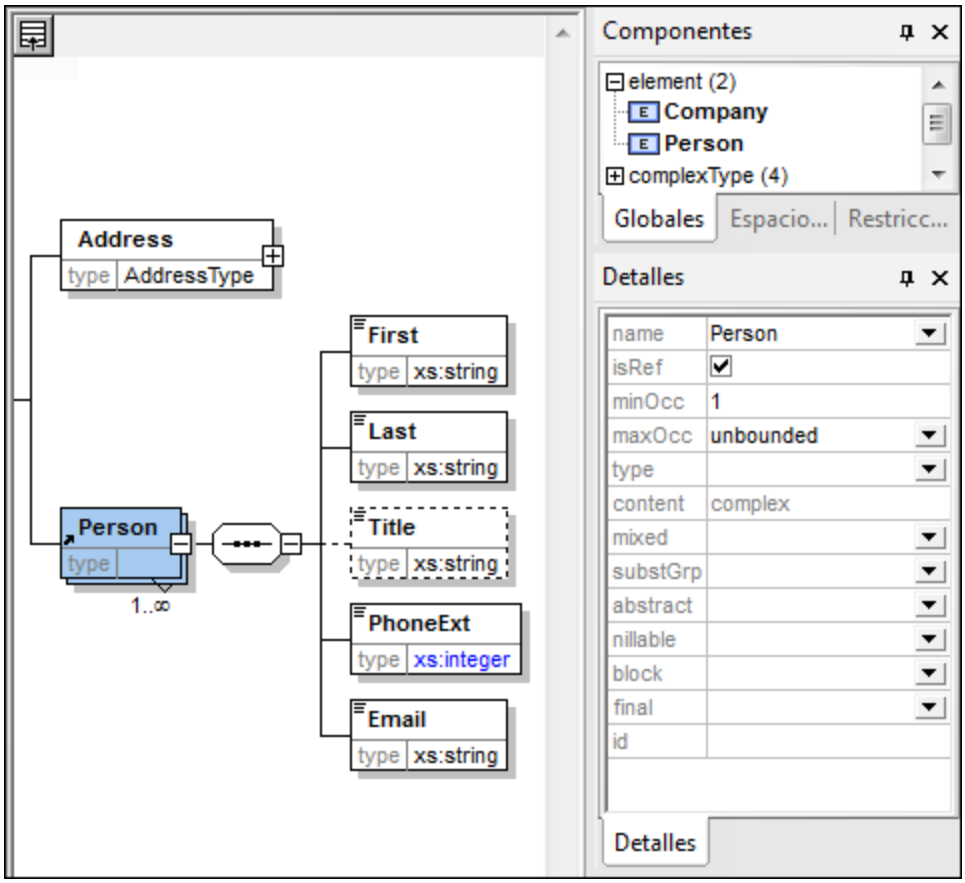


Nota: en esta sección aprendió a crear tipos complejos y tipos simples globales, que se pueden usar en definiciones de modelos de contenido. La ventaja de usar tipos globales es que se pueden volver a utilizar en múltiples definiciones.

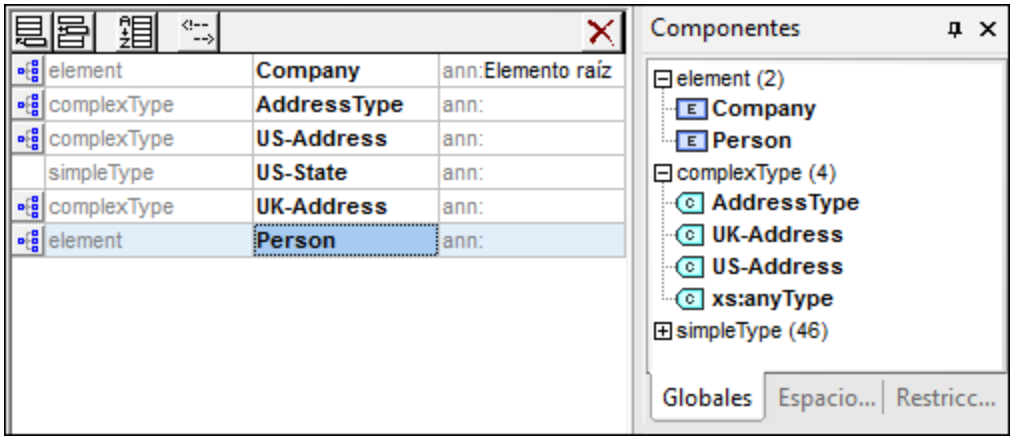
2.3.2 Referencias a elementos globales

En esta sección convertiremos el elemento `Person` definido localmente en un elemento global y haremos referencia a ese elemento global dentro del elemento `Company`.

1. Haga clic en (Mostrar nodos globales) para cambiar a la vista general del esquema.
2. Haga clic en el icono "Mostrar diagrama" del elemento `Company`.
3. Haga clic con el botón derecho en el elemento `Person` y seleccione la opción **Convertir en... | elemento global**. En el elemento `Person` aparece un pequeño icono de vínculo en forma de flecha, indicando que este elemento ahora hace referencia al elemento `Person` declarado globalmente. En el ayudante de entrada "Detalles" la casilla `isRef` aparece marcada.

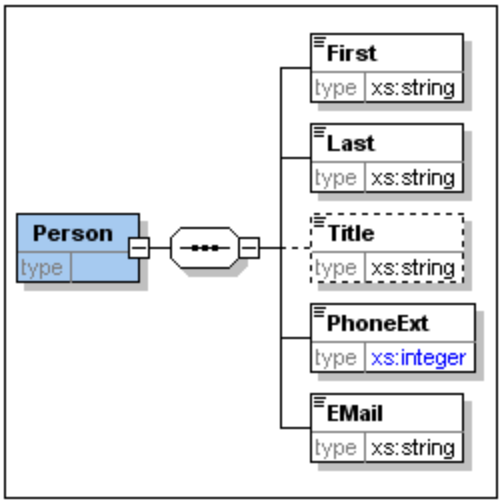


4. Haga clic en el icono "Mostrar nodos globales"  para volver a la vista general del esquema. El elemento `Person` aparece ahora como elemento global. Este elemento también aparece en la lista del ayudante de entrada "Componentes".



5. En el ayudante de entrada "Componentes" haga doble clic en el elemento `Person` para ver el modelo de contenido del elemento global `Person`.

 © 2016-2022 Altova GmbH



Observe que el recuadro del elemento global **no** tiene un icono de vínculo en forma de flecha. El motivo es que éste es el elemento al que se hace referencia. El elemento que hace referencia es el que lleva el icono de vínculo en forma de flecha.

Tenga en cuenta que:

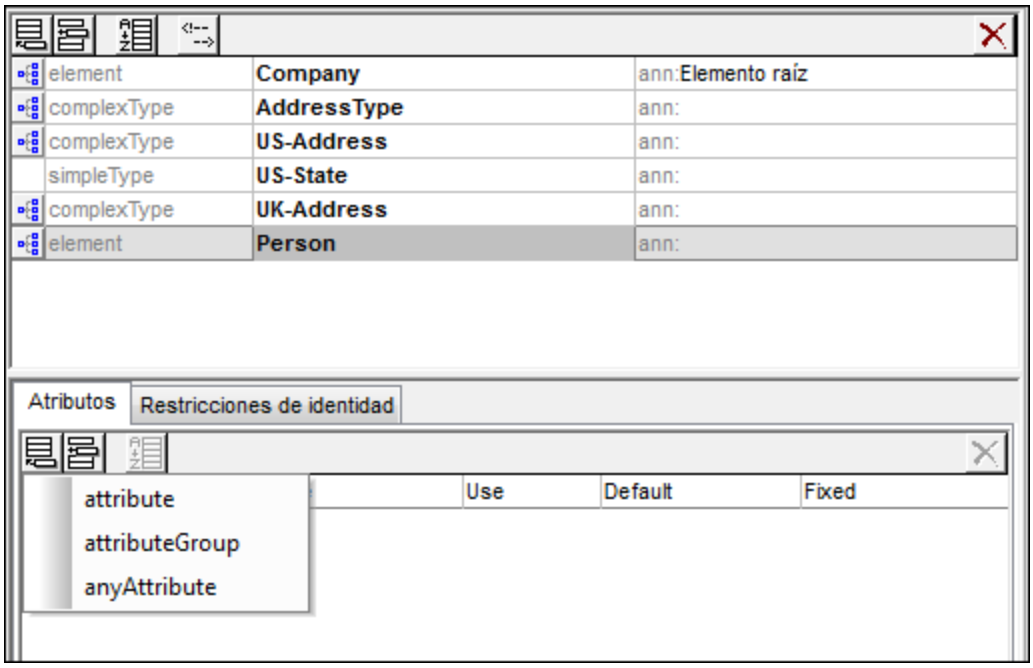
- un elemento que hace referencia a un elemento global debe tener el mismo nombre que el elemento global al que hace referencia.
- una declaración global no describe en qué parte del documento XML se debe usar un componente. Sólo describe su modelo de contenido. La ubicación del componente en el documento XML se especifica al hacer referencia a la declaración global desde otro componente.
- un elemento declarado globalmente se puede volver a utilizar en varias ubicaciones. La diferencia entre un elemento declarado globalmente y un tipo complejo declarado globalmente es que su modelo de contenido no se puede modificar sin modificar también el propio elemento global. Si cambia el modelo de contenido de un elemento que hace referencia a un elemento global, el modelo de contenido del elemento global también se cambiará y, con él, el modelo de contenido de los demás elementos que hagan referencia a ese elemento global.

2.3.3 Atributos y enumeraciones de atributos

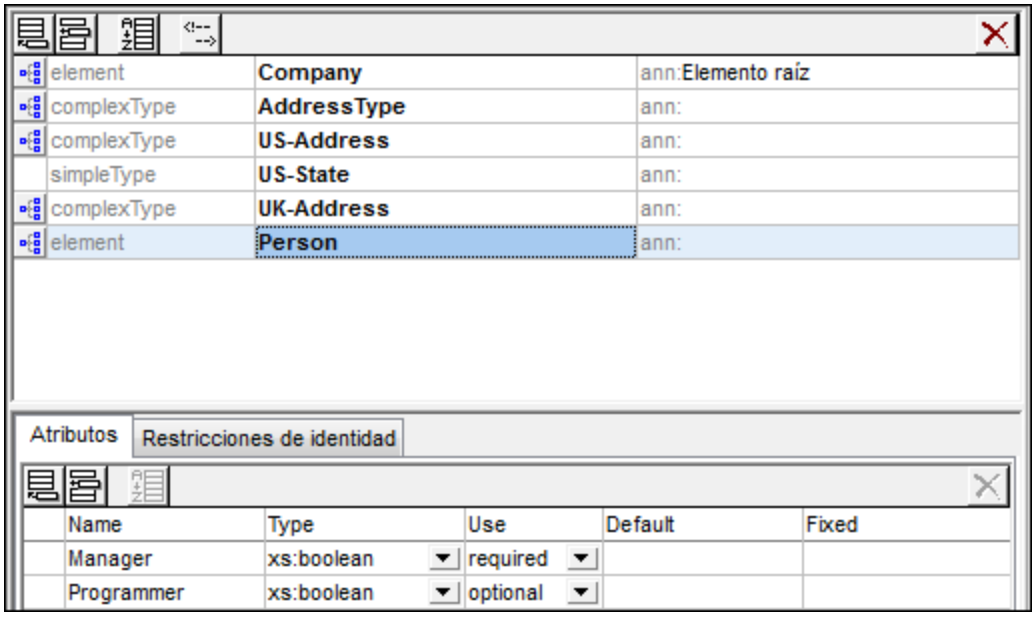
En esta sección aprenderá a crear atributos y enumeraciones para atributos.

Definir los atributos del elemento

1. En la vista general del esquema, haga clic en el elemento `Person`.
2. Justo debajo de la vista general del esquema hay un panel con dos pestañas. En la pestaña "Atributos" del panel inferior, haga clic en el icono "Anexar" , situado en la esquina superior izquierda. Seleccione la opción **attribute**.



3. En el campo `Name` escriba el nombre del atributo, en este caso `Manager`.
4. En el cuadro combinado `Type` seleccione el tipo `xs:boolean`.
5. En el cuadro combinado `Use` seleccione el uso `required` (obligatorio).



6. Siga los mismos pasos para crear un atributo llamado `Programmer` de tipo `xs:boolean` y de uso `optional` (opcional).

Definir enumeraciones para atributos

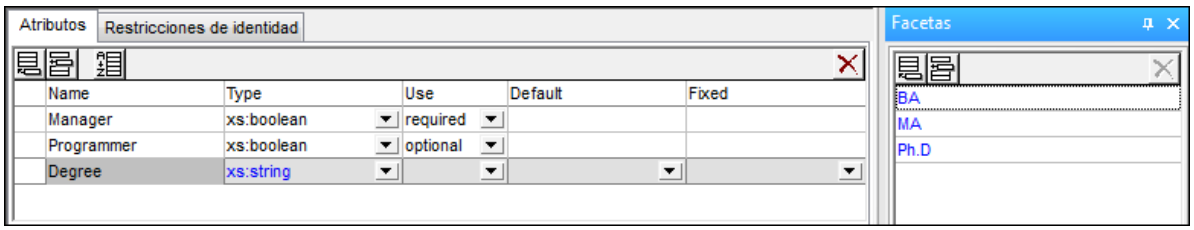
Las enumeraciones son los valores permitidos para un atributo determinado. Si el valor del atributo en el documento de instancia XML no se corresponde con las enumeraciones especificadas en el esquema XML, el

 © 2016-2022 Altova GmbH

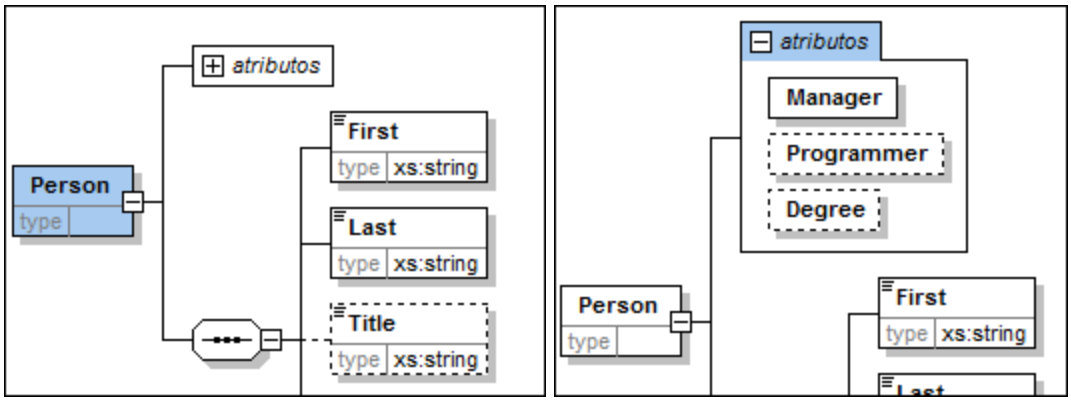
documento no será válido. A continuación aprenderá a crear enumeraciones para el atributo `Degree` del elemento `Person`.

Para ello haga lo siguiente:

1. En la vista general del esquema, haga clic en el elemento `Person`.
2. Pulse el icono "Anexar" situado en la esquina superior izquierda de la pestaña "Atributos" y seleccione la opción **attribute**.
3. Utilice `Degree` como nombre de este nuevo atributo y seleccione el tipo `xs:string`.
4. Haga clic de nuevo en el atributo `Degree` y después haga clic en la pestaña **Enumeraciones**del ayudante de entrada "Facetas" (*ver imagen siguiente*).



5. Ya en la pestaña **Enumeraciones** haga clic en el icono "Anexar" .
6. Escriba `BA` y pulse la tecla **Entrar**.
7. Siga los mismos pasos para agregar dos enumeraciones más: `MA` y `PhD`.
8. Después haga clic en el icono "Vista del modelo de contenido" de `Person`.



Los atributos que acaba de definir pueden verse en la vista del modelo de contenido. Al hacer clic en el icono de expansión podrá ver todos los atributos definidos para ese elemento. Este modo de visualización, así como la pestaña "Atributos", puede configurarse si selecciona la opción de menú **Diseño de esquemas | Configurar opciones de vista**. En el cuadro de diálogo de configuración, active o desactive la casilla **Atributos** del panel **Mostrar en el diagrama**.

9. Haga clic en el icono "Mostrar nodos globales" para volver a la vista general del esquema.

Guardar el esquema

Antes de guardar el archivo de esquema, cambie el nombre del archivo `AddressLast.xsd` que viene con XMLSpy y póngale cualquier otro nombre (p. ej. `AddressLast_original.xsd`) para no sobrescribirlo. Guarde el esquema con el nombre que prefiera (**Archivo | Guardar como**). Le recomendamos que lo guarde con el nombre `AddressLast.xsd` porque el archivo XML que creará en la siguiente parte del tutorial estará basado en el esquema `AddressLast.xsd`.

© 2016-2022 Altova GmbH

2.4 Esquemas XML: características de XMLSpy

Ahora que terminó de diseñar el esquema XML, puede empezar a familiarizarse con algunos accesos directos de navegación 81 y con la función de generación de documentación de esquemas 83 de XMLSpy. Todo ello se describe en esta sección.

Comandos utilizados en esta sección

En esta sección del tutorial utilizará sólo la vista Esquema y estos comandos:

	**Mostrar diagrama** (o mostrar la vista del modelo de contenido). Este icono está situado a la izquierda de cada componente global en la vista general del esquema. Si hace clic en este icono, podrá ver el modelo de contenido del componente global.

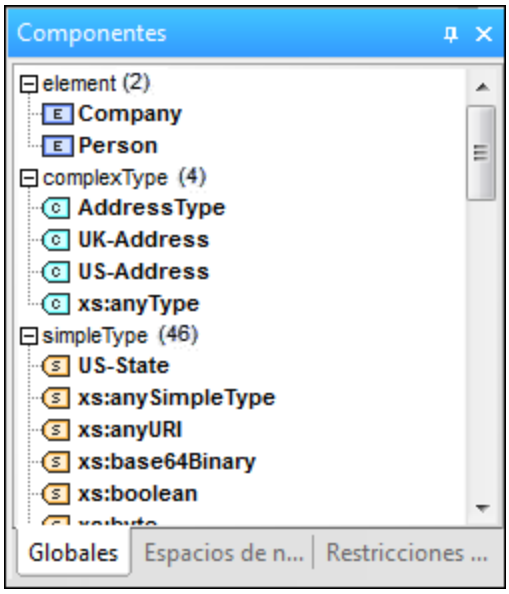
2.4.1 Navegar por el esquema

En esta sección aprenderá a navegar por la vista Esquema con eficacia. Le sugerimos que experimente con estos mecanismos de navegación para familiarizarse con ellos.

Visualizar el modelo de contenido de un componente global

Los componentes globales que pueden tener modelos de contenido son los tipos complejos, los elementos y los grupos de elementos. La vista del modelo de contenido de estos componentes se puede abrir de varias maneras:

- En la vista general del esquema, haga clic en el icono **Mostrar diagrama**  situado a la izquierda del nombre del componente.
- Estando en la vista general del esquema o en la vista del modelo de contenido, haga doble clic en el elemento, tipo complejo o grupo de elementos en el ayudante de entrada "Componentes" (*ver imagen siguiente*). A continuación se abre el modelo de contenido del componente.

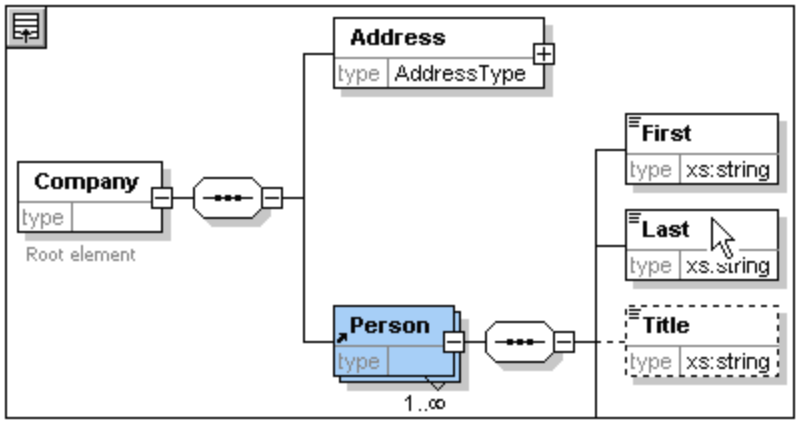


Sin embargo, al hacer doble clic en los demás componentes globales (tipo simple, atributo, grupo de atributos) del ayudante de entrada "Componentes", se resaltará el componente en la vista general del esquema (porque estos componentes no tienen modelo de contenido).

El mecanismo de doble clic descrito en este apartado funciona tanto en la pestaña "Globales" como en la pestaña "Espacios de nombres" del ayudante de entrada "Componentes".

Ir a la definición de un elemento global desde un elemento que le hace referencia

Si un modelo de contenido incluye un elemento que hace referencia a un elemento global, puede ir directamente al modelo de contenido de ese elemento global o a cualquiera de sus componentes. Para ello mantenga pulsada la tecla **Ctrl** y haga doble clic en el elemento que le interesa.

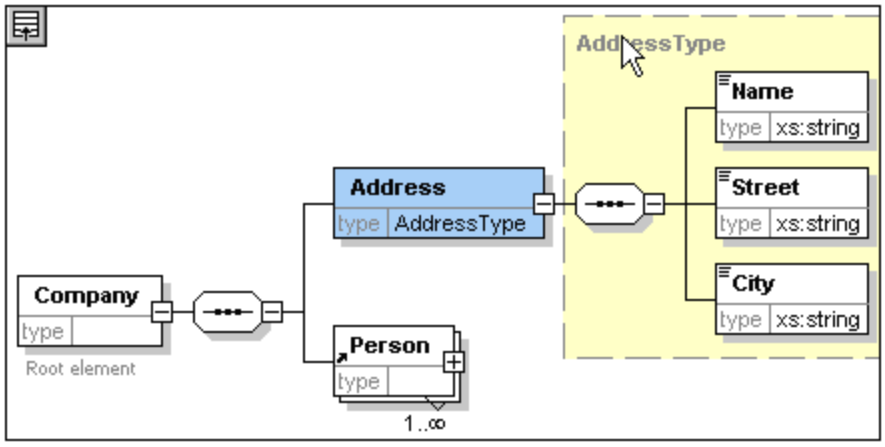


Mientras el elemento `Last` está resaltado, todas sus propiedades aparecen en la ventana de información y en los ayudantes de entrada correspondientes.

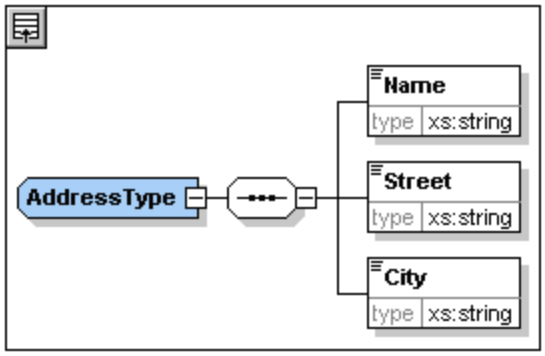
 © 2016-2022 Altova GmbH

Ir a la definición de un tipo complejo

Los tipos complejos se suelen usar como tipo de algún elemento del modelo de contenido. Para ir directamente a la definición de un tipo complejo desde un modelo de contenido, haga doble clic en el **nombre** del tipo complejo en el recuadro amarillo (*vea el puntero del ratón en la imagen siguiente*).



Como resultado se abre la vista del modelo de contenido del tipo complejo.



Nota: al igual que en el caso de elementos globales a los que se hace referencia, puede ir directamente a un elemento de la definición del tipo complejo si mantiene pulsada la tecla Ctrl mientras hace doble clic en el elemento que le interesa en el modelo de contenido que contiene el tipo complejo.

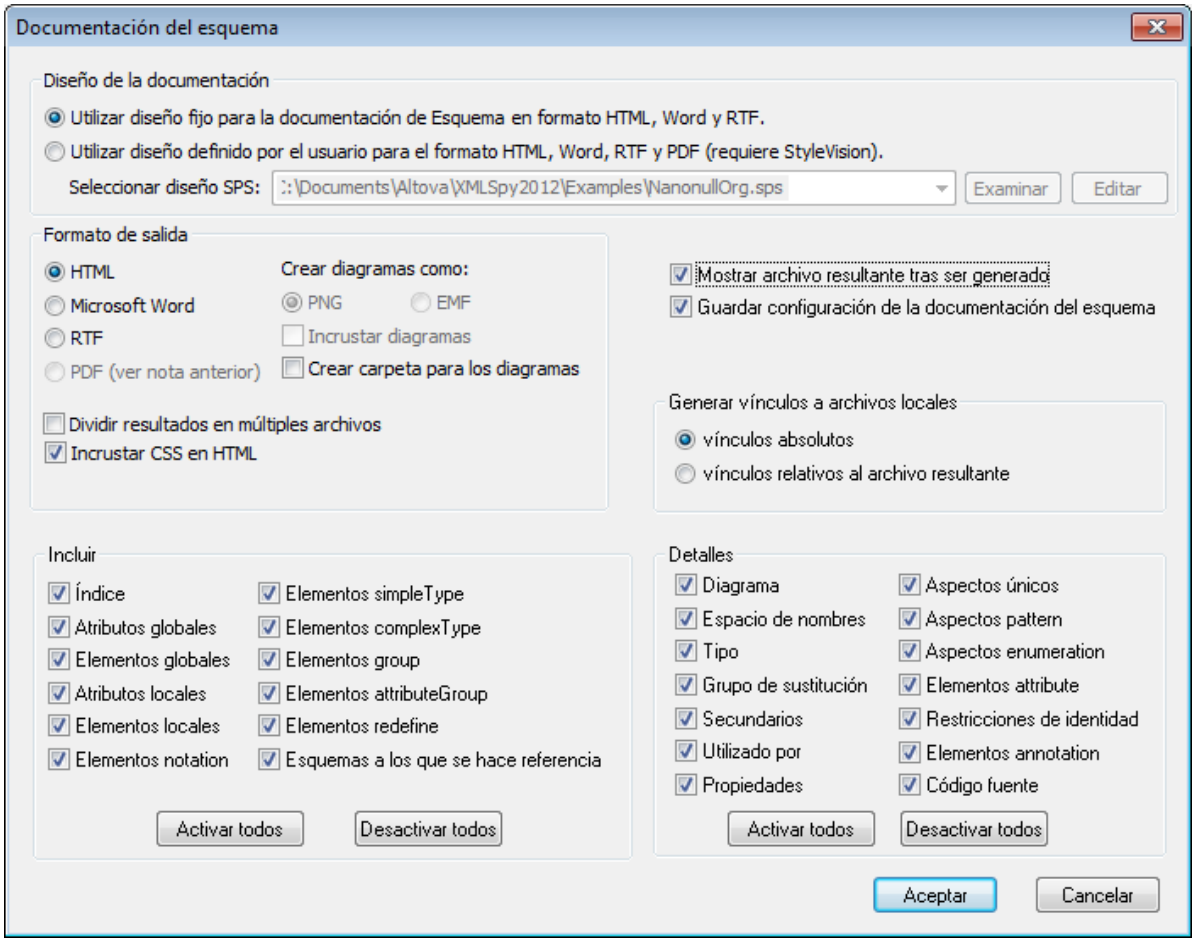
2.4.2 Documentación del esquema

XMLSpy puede generar documentación detallada para sus esquemas XML en formato HTML y Microsoft Word (MS Word). Puede elegir los componentes que se deben documentar y en qué detalle. Los componentes relacionados aparecen hipervinculados tanto en HTML como en MS Word. Para poder generar documentación en formato MS Word es necesario tener instalado Microsoft Word en el equipo (o en la red).

En esta sección aprenderá a generar documentación para el esquema XML `AddressLast.xsd`.

Siga estos pasos:

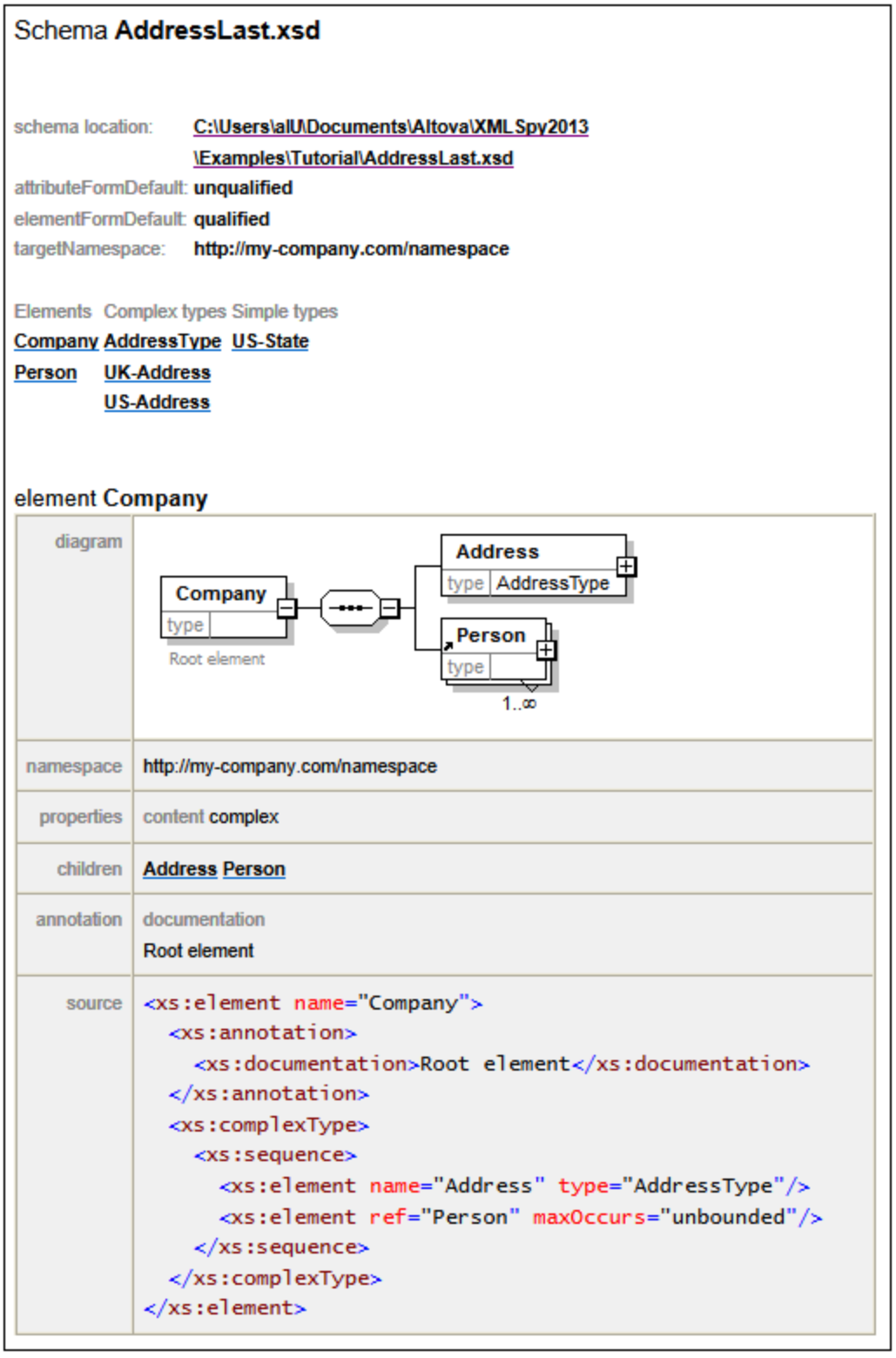
1. Seleccione la opción de menú **Diseño de esquemas | Generar documentación**. Aparece el cuadro de diálogo "Documentación del esquema".



2. En la sección "Formato de salida" seleccione "HTML" y pulse el botón **Aceptar**.
3. En el cuadro de diálogo "Guardar como" seleccione la ubicación donde desea guardar el archivo y asígnele un nombre apropiado (p. ej. `AddressLast.html`). Después haga clic en **Guardar**.

   El documento HTML se abre en la vista Explorador de XMLSpy. Haga clic en cualquier hipervínculo para ir al componente correspondiente.

© 2016-2022 Altova GmbH



Schema **AddressLast.xsd**

schema location: C:\Users\alU\Documents\Altova\XMLSpy2013\Examples\Tutorial\AddressLast.xsd
attributeFormDefault: **unqualified**
elementFormDefault: **qualified**
targetNamespace: **http://my-company.com/namespace**

Elements Complex types Simple types
Company **AddressType** **US-State**
Person **UK-Address**
US-Address

element **Company**

diagram

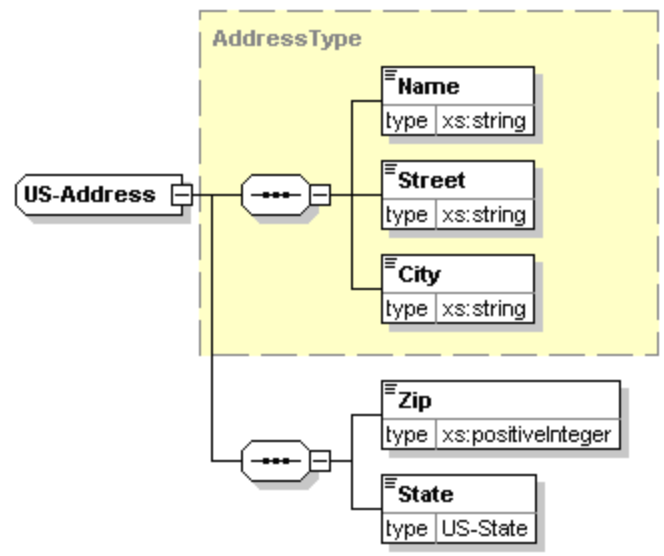
namespace	http://my-company.com/namespace
properties	content complex
children	**Address** **Person**
annotation	documentation Root element

source

```
<xs:element name="Company">
  <xs:annotation>
    <xs:documentation>Root element</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Address" type="AddressType"/>
      <xs:element ref="Person" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

La imagen anterior muestra la **primera página** de la documentación del esquema en formato HTML. Si se incluyen componentes de otros esquemas, también se documentan esos esquemas.

complexType **US-Address**

diagram	
namespace	http://my-company.com/namespace
type	extension of **AddressType**
properties	name US-Address base AddressType
children	**Name Street City Zip State**
source	<xs:complexType name="US-Address"> <xs:complexContent> <xs:extension base="AddressType"> <xs:sequence> <xs:element name="Zip" type="xs:positiveInteger"/> <xs:element name="State" type="US-State"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

La imagen anterior muestra cómo se documentan los tipos complejos.

© 2016-2022 Altova GmbH

element **US-Address/Zip**

diagram	**Zip** type xs:positiveInteger
namespace	http://my-company.com/namespace
type	**xs:positiveInteger**
properties	name Zip isRef 0 content simple
source	<xs:element name="Zip" type="xs:positiveInteger"/>

element **US-Address/State**

diagram	**State** type US-State
namespace	http://my-company.com/namespace
type	**US-State**
properties	name State isRef 0 content simple
source	<xs:element name="State" type="US-State"/>

simpleType **US-State**

namespace	http://my-company.com/namespace
type	**xs:string**
properties	name US-State
used by	element **US-Address/State**
source	<xs:simpleType name="US-State"> <xs:restriction base="xs:string"/> </xs:simpleType>

La imagen anterior muestra cómo se documentan los elementos y los tipos simples.

Ahora puede probar la opción de formato de salida MS Word. El documento Word se abrirá en MS Word. Para usar los hipervínculos en el documento MS Word debe mantener pulsada la tecla **Ctrl** mientras hace clic en el hipervínculo.

2.5 Documentos XML

En esta sección aprenderá a crear documentos XML y trabajar con ellos en XMLSpy. También aprenderá a usar varias funciones de edición inteligente de XMLSpy.

Objetivo

El objetivo de esta sección es aprender a:

- Crear un documento XML nuevo basado en el esquema `AddressLast.xsd`.
- Especificar el tipo de un elemento para hacer un modelo de contenido extendido de ese elemento para que el elemento pueda utilizarlo durante la validación.
- Insertar elementos y atributos y escribir contenido en ellos en la vista Cuadrícula y la vista Texto usando los ayudantes de entrada inteligentes.
- Copiar datos XML de XMLSpy y pegarlos en Microsoft Excel, agregar datos nuevos en MS Excel y copiar los datos modificados en MS Excel para pegarlos en XMLSpy. Esta función está disponible en la vista en forma de tabla de la vista Cuadrícula.
- Ordenar los elementos XML con ayuda de la función de ordenación de la vista en forma de tabla.
- Validar el documento XML.
- Modificar el esquema para que las extensiones de teléfono puedan tener tres dígitos.

Comandos utilizados en esta sección

En esta sección del tutorial utilizará sobre todo la vista Cuadrícula y la vista Texto. También utilizará un poco la vista Esquema. Estos son los comandos utilizados en esta sección:

	**Archivo | Nuevo**. Crea un tipo de archivo XML nuevo.
	**Vista | Vista Texto**. Cambia a la vista Texto.
	**Vista | Vista Cuadrícula**. Cambia a la vista Cuadrícula mejorada.
	**XML | Mostrar como tabla**. Muestra varias repeticiones de un solo tipo de elemento en un solo nivel jerárquico en forma de tabla. Esta vista del elemento se llama vista Tabla. El icono se utiliza para cambiar de la vista Tabla a la vista Cuadrícula normal.
	por tanto, utiliza su propia extensión de archivo (**F7**). Comprueba si el documento tiene un formato XML correcto.
	por tanto, utiliza su propia extensión de archivo (**F8**). Valida el documento XML con el esquema o la DTD asociados.
	Abre el archivo DTD o de esquema XML asociado.

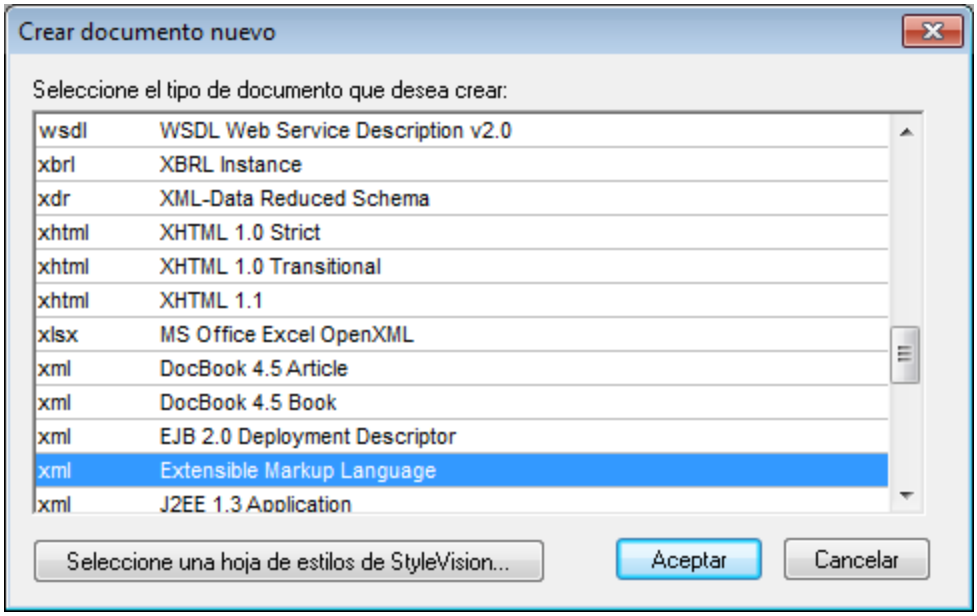
2.5.1 Crear un archivo XML nuevo

Cuando crea un archivo XML nuevo en XMLSpy, tiene la opción de basar el archivo nuevo en un esquema (en una DTD o un esquema XML) o no. En esta sección aprenderá a crear un archivo nuevo basado en el esquema `AddressLast.xsd` que creó en las secciones anteriores del tutorial.

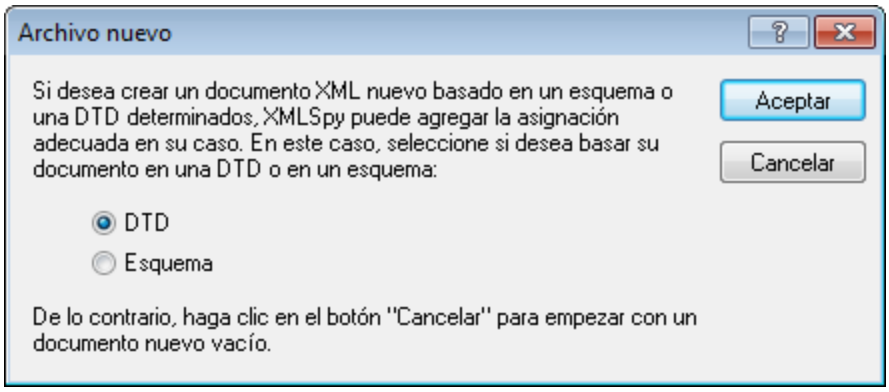
© 2016-2022 Altova GmbH

Para crear el archivo XML nuevo:

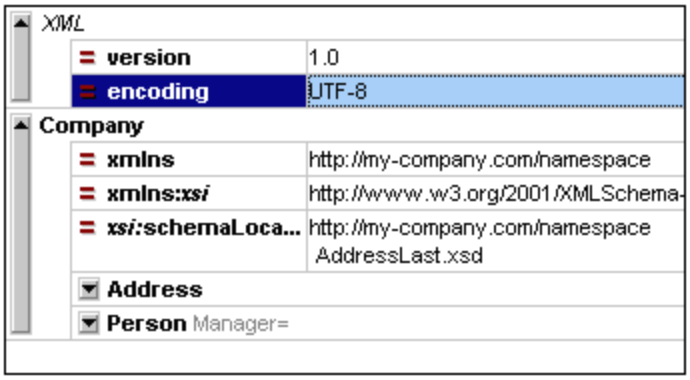
1. Seleccione la opción de menú **Archivo | Nuevo**. Aparece el cuadro de diálogo "Crear documento nuevo".



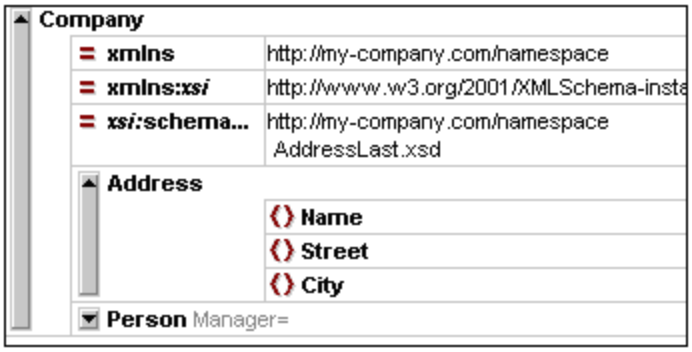
2. Seleccione la opción `Extensible Markup Language` (un documento XML genérico) y haga clic en **Aceptar**. Aparece un aviso que le pregunta si desea basar el documento XML en una DTD o en un esquema.



3. Haga clic en el botón de opción "Esquema" y pulse el botón **Aceptar**. A continuación aparece otro cuadro de diálogo en el que debe seleccionar el archivo de esquema sobre el que se debe basar el documento XML.
4. Utilice los botones **Examinar** o **Ventana** para buscar el archivo de esquema. El botón **Ventana** muestra todos los archivos que estén abiertos en XMLSpy y todos los archivos de proyecto disponibles. Seleccione `AddressLast.xsd` (consulte la introducción del tutorial 38 para ver dónde está ubicado este archivo) y haga clic en **Aceptar**. Como resultado se abre en la ventana principal un documento XML que contiene los elementos definidos en el esquema.
5. Haga clic en la pestaña **Cuadrícula** para cambiar a la vista Cuadrícula.
6. En la vista Cuadrícula observe la estructura del documento. Haga clic en los elementos para expandirlos. El documento debería tener este aspecto:



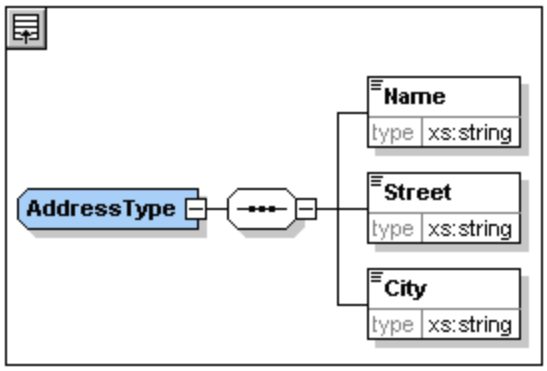
7. Haga clic en el icono situado junto a `Address` para ver los elementos secundarios de `Address`. El documento debería tener este aspecto:



8. Seleccione la opción de menú **Archivo | Guardar**, asígnele un nombre al documento XML y guárdelo en la carpeta Tutorial.

2.5.2 Especificar el tipo de un elemento

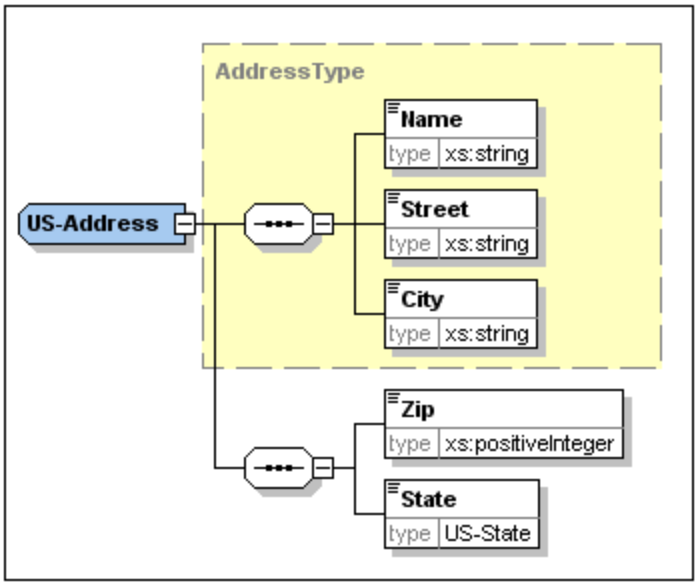
Los elementos secundarios de `Address` son los elementos definidos para el tipo complejo global `AddressType` (cuyo modelo de contenido se define en el esquema XML `AddressLast.xsd` y aparece en la imagen siguiente de la vista Esquema).



Sin embargo, en este ejemplo vamos a usar un tipo de dirección concreta en lugar de un tipo de dirección genérica. Recordará que en el esquema `AddressLast.xsd` creamos tipos complejos globales para `US-`

 © 2016-2022 Altova GmbH

`Address` y `UK-Address` cuando extendimos el tipo complejo `AddressType`. El modelo de contenido de `US-Address` aparece en la imagen siguiente.



En el documento XML vamos a especificar que el elemento `Address` se ajuste a uno de los tipos de `Address` (el tipo `US-Address` o el tipo `UK-Address`) en lugar de ajustarse al tipo genérico `AddressType`. Para ello debemos especificar el tipo complejo extendido pertinente como atributo del elemento `Address`.

Para añadir este atributo del elemento `Address` siga estos pasos:

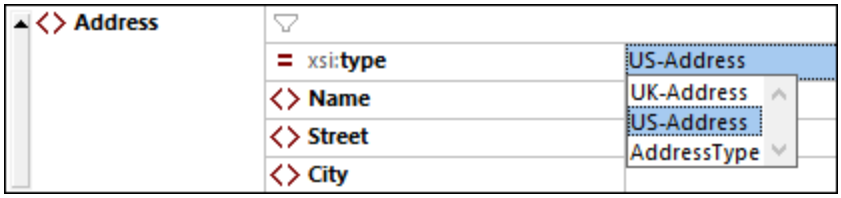
1. En el documento XML, haga clic con el botón en el documento XML derecho en el elemento `Name` y seleccione la opción **Insertar antes** del menú contextual (*imagen siguiente*).



2. Se añade el elemento nodo nuevo **`new_elem`** antes del elemento **`Name`** (*imagen siguiente*). Haga clic en el tipo de elemento a la izquierda del nombre de nodo y seleccione el tipo de nodo *Atributo* el menú que aparece (*imagen siguiente*). Esa acción cambia el tipo de nodo a *Atributo*, aunque el nombre sigue siendo **`new_elem`**.



3. Haga doble clic en el nombre del nodo y, en el ayudante de entrada que aparece, seleccione **xsi:type**.
4. Pulse la tecla **Tabulador** para ir al campo del valor del atributo. Aparece un menú que contiene los valores **xsi:type** disponibles (*imagen siguiente*). Estos valores son los tipos complejos que se han definido para el elemento **Address** en el esquema.



5. Seleccione **US-Address** como el valor del atributo **xsi:type**.

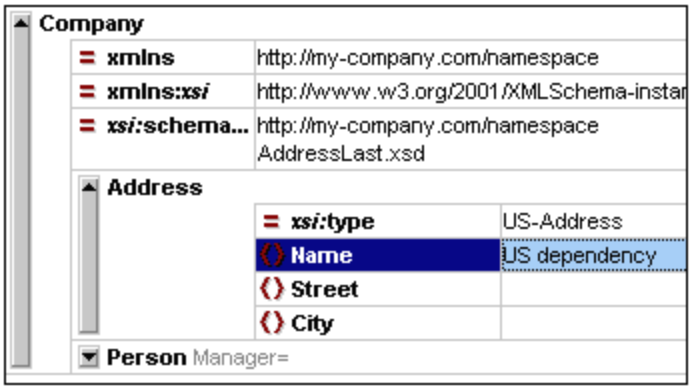
Nota: el prefijo `xsi` permite usar comandos de XML Schema relacionados en su instancia de documento XML. Observe, por ejemplo, que el espacio de nombres para el prefijo `xsi` se añadió automáticamente al elemento cuando asignó el esquema al archivo XML. En esta sección especificó un tipo para el elemento `Address`. Consulte la especificación del estándar XML Schema para obtener más información.

2.5.3 Introducir datos en la vista Cuadrícula

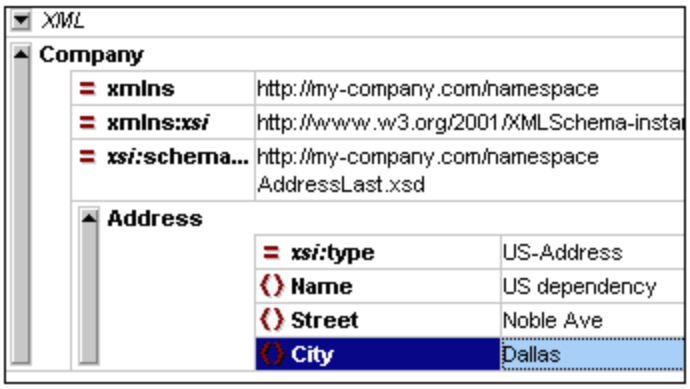
Ahora puede introducir datos en su documento XML. Siga estos pasos:

1. Haga doble clic en el campo de valor del elemento **Name** (o use las teclas de dirección para desplazarse hasta el campo) y escriba *US dependency*. Pulse la tecla **Entrar**.

© 2016-2022 Altova GmbH



2. Utilice el mismo método para escribir el nombre de los elementos **`Street`** y **`City`** (por ejemplo, *Noble Ave* y *Dallas*).
3. Haga clic en el elemento **`Person`** y pulse la tecla **Suprimir** para eliminar este elemento. (Volveremos a añadir este elemento en la siguiente sección del tutorial.) A continuación, aparece resaltado todo el elemento **`Address`**.
4. Haga clic en uno de los elementos secundarios del elemento **`Address`** para que no esté seleccionado todo el elemento. Este es el aspecto que debe tener su documento XML:



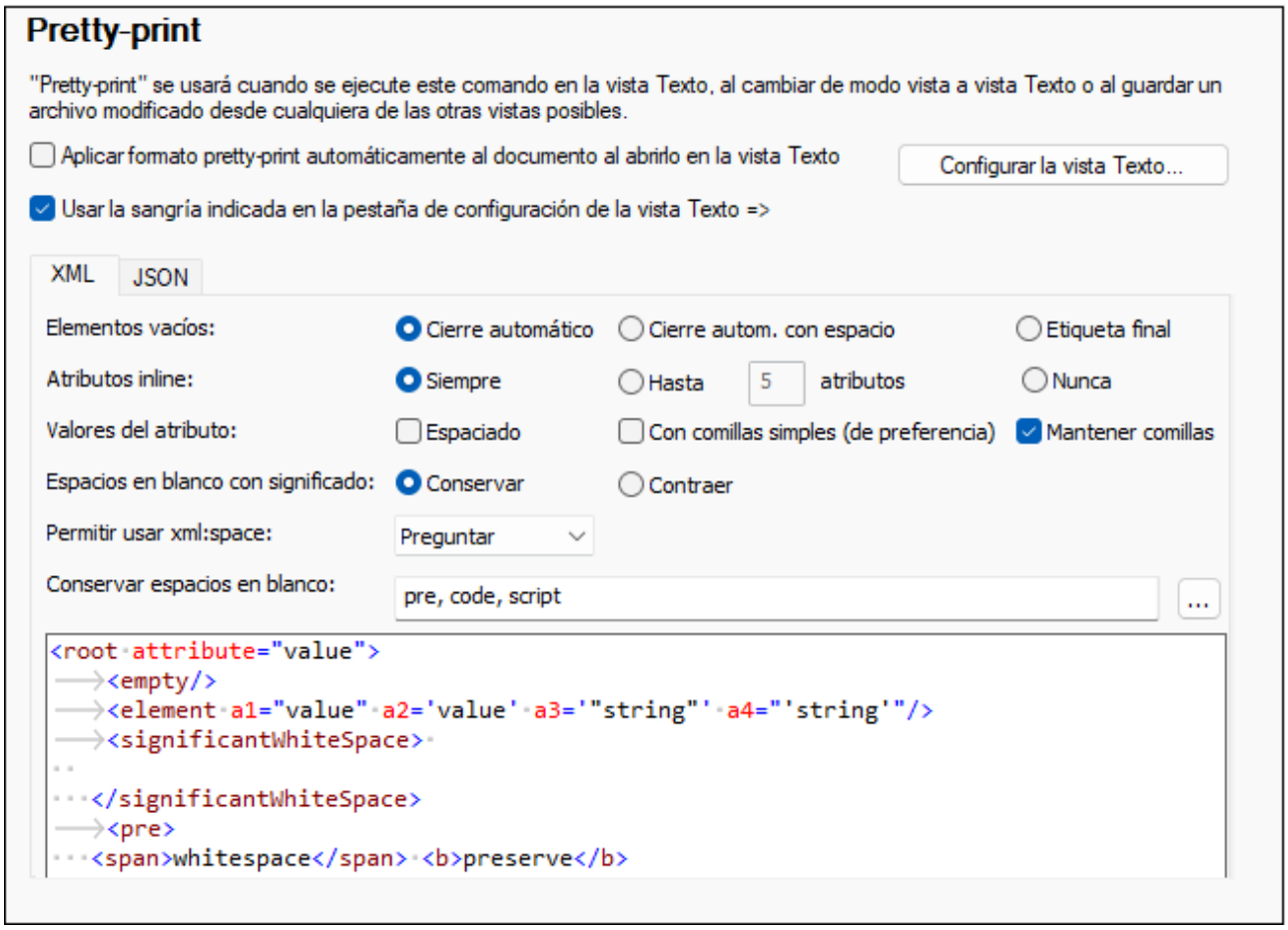
2.5.4 Introducir datos en la vista Texto

La vista Texto presenta los datos y el marcado de los archivos XML con un diseño estructurado fácil de leer y ofrece funciones de edición inteligentes basadas en la información del esquema asociado al archivo XML.

Diseño del documento

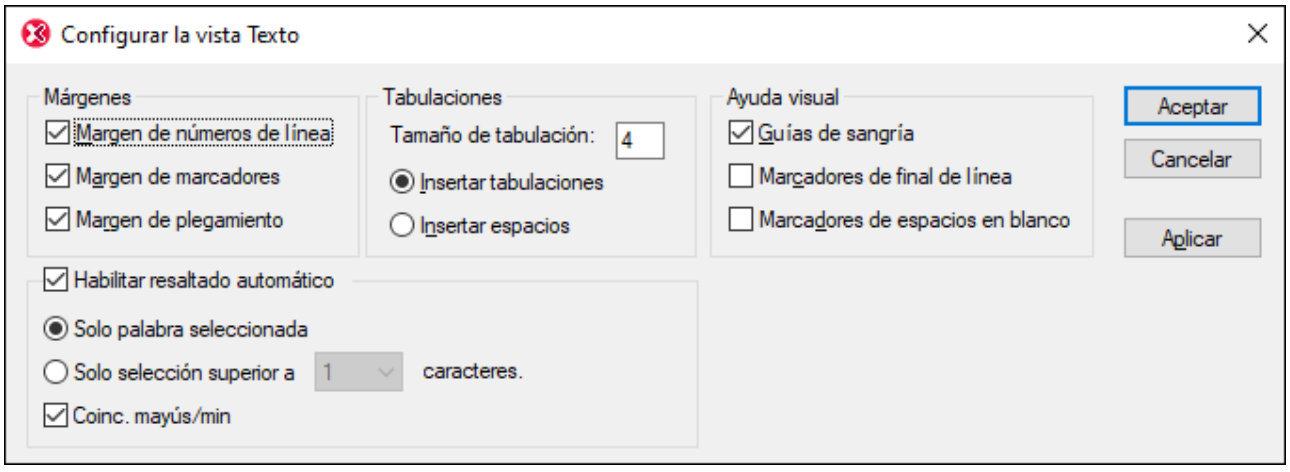
Puede configurar el diseño del documento para la vista Texto en dos sitios:

Configurar el formato Pretty-print
Estas opciones se encuentran en la sección Pretty-print del cuadro de diálogo "Opciones" (*imagen siguiente*). Al cambiar una opción puede ver inmediatamente cómo afectaría ese cambio al formato en el panel de vista previa que hay en la parte inferior. Puede configurar estas opciones como quiera. Al editar el documento en la vista Texto puede ocurrir que el texto pierda parte de su estructura, sobre todo si copia y pega bloques de texto. Para volver a obtener un diseño jerárquico solo tiene que hacer clic en el comando **Editar | Pretty Print**.



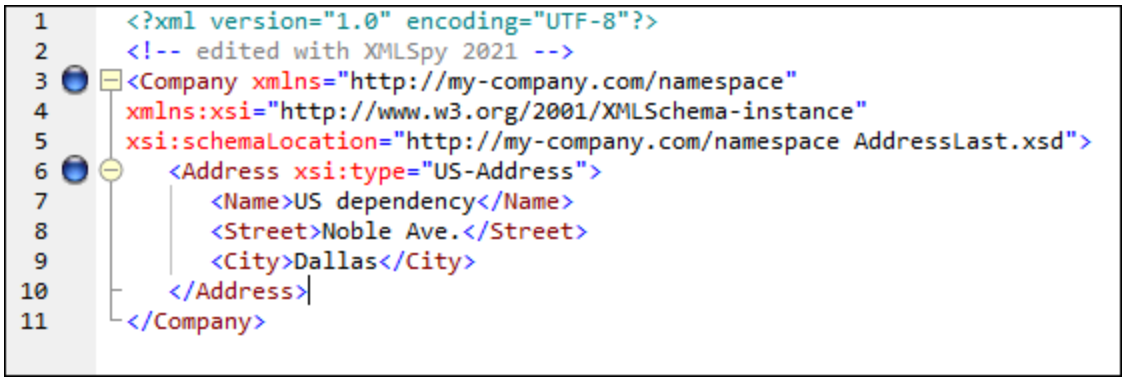
Configurar la vista Texto

El cuadro de diálogo "Configurar la vista Texto" (*imagen siguiente*) no solo permite editar otros aspectos del diseño, sino también activar y desactivar opciones de la vista Texto como la numeración de líneas o el margen de plegamiento. Para acceder a este cuadro de diálogo use el comando **Vista | Configurar la vista Texto**.



 © 2016-2022 Altova GmbH

En la imagen siguiente puede ver un archivo XML en la vista Texto con las opciones que se han definido en el cuadro de diálogo de la imagen anterior.



A la izquierda se aprecian tres márgenes: (i) el margen de la numeración de línea, (ii) el margen de marcadores (que contiene dos marcadores azules) y (iii) el margen de plegamiento del código fuente (que permite expandir o contraer los elementos XML). Las líneas verticales de color gris claro son las guías de indentación de las etiquetas de los distintos niveles jerárquicos. También hay otras ayudas visuales, como los marcadores de final de línea y los marcadores de espacios en blanco, que se pueden activar o desactivar en el panel *Ayuda visual* (*imagen anterior*).

Nota: las opciones de configuración de marcadores y del formato pretty-print se describen en el apartado Configuración de la vista Texto 46 de este tutorial.

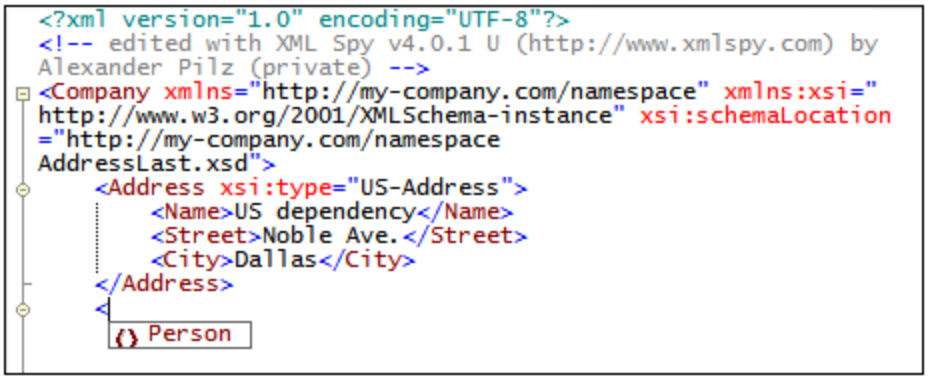
Edición en la vista Texto

En esta sección aprenderá a introducir y editar datos en la vista Texto a fin de familiarizarse con las distintas funciones de la vista Texto.

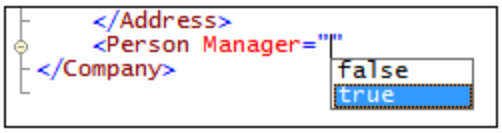
Nota: la opción *Validar al modificar* está activada por defecto, por lo que cualquier error de validación que surja durante la edición aparecerá marcado inmediatamente y en la ventana Mensajes aparecerá el mensaje de error correspondiente. Ignore todo esto por ahora. Si prefiere no usar esta opción puede desactivarla en el opciones de validación del cuadro de diálogo "Opciones" 1584. Si la desactiva, recuerde que siempre puede validar el documento en otro momento (consulte la sección siguiente de este tutorial).

Para ello haga lo siguiente:

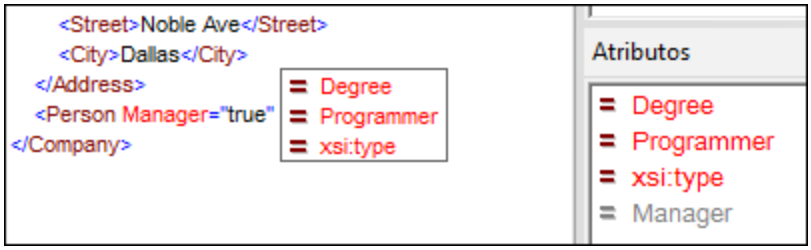
1. Seleccione la opción de menú **Vista | Vista Texto** o haga clic en la pestaña *Texto*. Verá el documento XML en formato texto y con color de sintaxis.
2. Coloque el cursor después de la etiqueta final del elemento Address y pulse la tecla **Entrar** para insertar una línea nueva.
3. Escriba el corchete angular < en esa posición. En ese momento aparecerá una lista desplegable con todos los elementos permitidos en esa posición (de acuerdo con el esquema). Puesto que en esta posición el único elemento permitido es el elemento `Person` la lista contiene solo ese elemento.



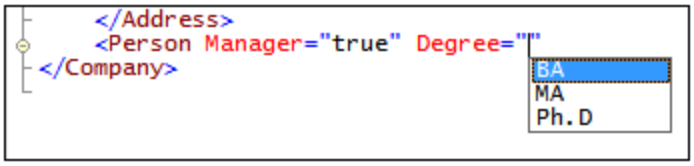
4. Seleccione la opción `Person`. Como resultado se insertan el elemento `Person` y su atributo `Manager` y el cursor aparece dentro del campo de valor del atributo **`Manager`**.
5. Seleccione la opción `true` de la lista emergente para el atributo `Manager`.



6. Mueva el cursor al final de la línea (usando si lo desea la tecla **Fin**) y pulse la barra espaciadora. Como resultado aparece una lista desplegable que contiene todos los atributos permitidos en esta posición. Los atributos permitidos también aparecen en color ojo en la ventana del ayudante de entrada Atributos. El atributo `Manager` aparece atenuado porque ya se ha utilizado.

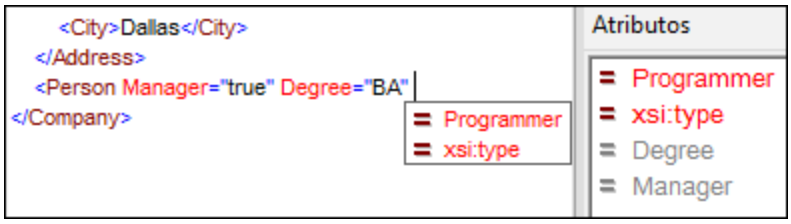


7. Seleccione la opción `BA` con la tecla de dirección **Abajo** y pulse la tecla **Entrar**. Como resultado se abre otra lista, en la que puede seleccionar una de las enumeraciones predefinidas (`BA`, `MA` o `PhD`). Las enumeraciones son valores permitidos por el esquema XML.

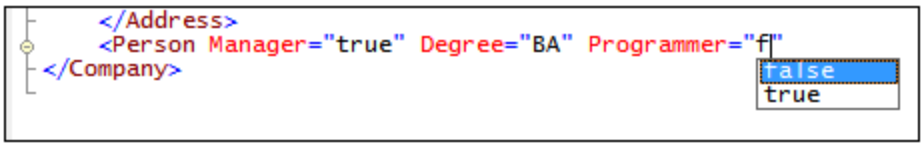


8. Seleccione la opción `BA` con la tecla de dirección **Abajo** y pulse la tecla **Entrar**. Después mueva el cursor al final de la línea (con la tecla **Fin**) y pulse la barra espaciadora. Observe que los atributos `Manager` y `Degree` aparecen atenuados en el ayudante de entrada Atributos.

 © 2016-2022 Altova GmbH

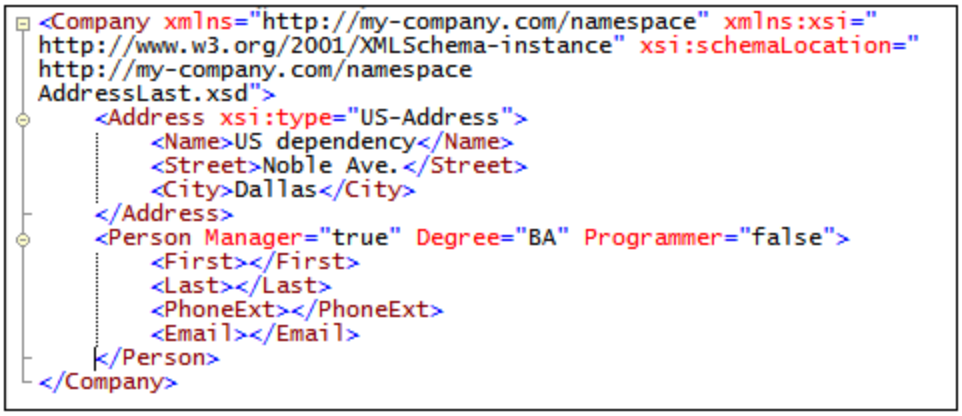


9. Seleccione la opción `Programmer` con la tecla de dirección **Abajo** y pulse la tecla **Entrar**.



10. Escriba la letra "`f`" y pulse la tecla **Entrar**.

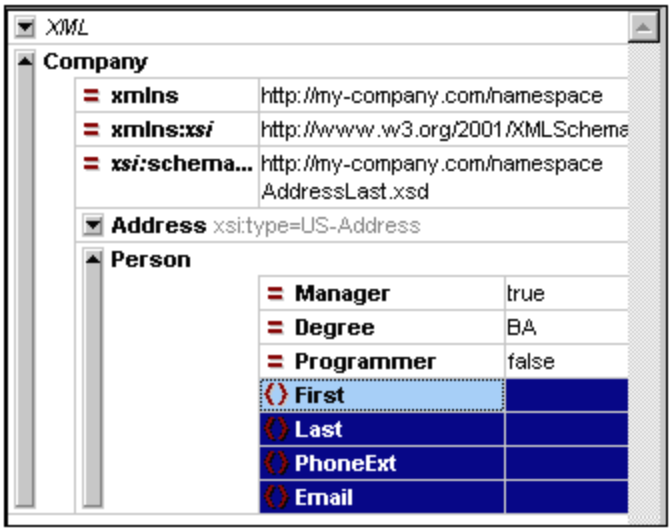
11. Mueva el cursor al final de la línea (con la tecla **Fin**) y escriba el corchete angular **>**. XMLSpy inserta automáticamente todos los elementos secundarios necesarios del elemento `Person`. Observe que no se inserta el elemento opcional `Title`. Cada elemento incluye etiquetas iniciales y finales, pero no tienen contenido.



Ahora podría escribir los datos del elemento `Person` en la vista Texto, pero vamos a pasar a la vista Cuadrícula para que aprecie la flexibilidad de cambiar de una vista a otra mientras edita un documento.

Cambiar a la vista Cuadrícula

Para cambiar a la vista Cuadrícula, seleccione la opción de menú **Vista | Vista Cuadrícula mejorada** o haga clic en la pestaña **Cuadrícula**. Los elementos secundarios de `Person` recién añadidos ahora aparecen en la vista Cuadrícula.



Y ahora ya podemos validar el documento y corregir los errores que se encuentren durante el proceso de validación.

2.5.5 Validar el documento

XMLSpy ofrece dos maneras de evaluar la calidad del documento XML:

- comprobación de formato XML
- validación del documento

	Comprobación de formato XML
	Validación del documento

Si ambos métodos de evaluación encuentran errores, deberá corregir el documento.

La opción *Validar al modificar* está activada por defecto, por lo que cualquier error de validación que surja durante la edición aparecerá marcado inmediatamente y en la ventana Mensajes aparecerá el mensaje de error correspondiente. Ignore todo esto por ahora. Si prefiere no usar esta opción puede desactivarla en el opciones de validación del cuadro de diálogo "Opciones" 1584. Si la desactiva, recuerde que siempre puede comprobar el formato del documento en otro momento. Más abajo explicamos cómo hacerlo.

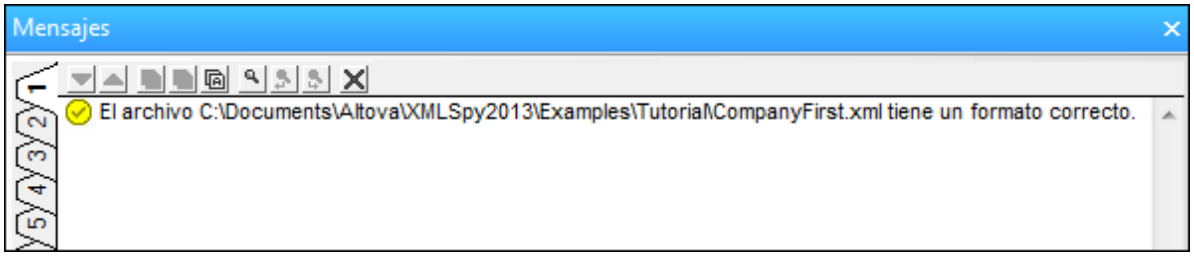
Comprobación de formato XML

Un documento XML tiene un formato XML correcto si las etiquetas iniciales coinciden con las etiquetas finales, si los elementos están anidados correctamente, si no hay caracteres injustificados y si no faltan caracteres (como en una entidad en la que falta el delimitador de punto y coma, etc.).
La comprobación de formato XML se puede llevar a cabo en cualquier vista de edición. Para hacer la comprobación de un documento:

1. Seleccione la vista Texto, por ejemplo.

 © 2016-2022 Altova GmbH

2. Seleccione la opción de menú **XML | Comprobar formato XML** o pulse la tecla **F7**. (También puede usar el icono correspondiente de la barra de herramientas). A continuación aparece un mensaje en la ventana "Mensajes" situada en la parte inferior de la ventana principal.



Observe que la ventana "Mensajes" cuenta con 9 pestañas. El resultado de la comprobación o validación siempre aparece en la pestaña activa. Por tanto, puede comprobar si el formato XML de un archivo es correcto en la pestaña 1 y cambiar a la pestaña 2 para validar el siguiente documento (de lo contrario el resultado de la nueva validación sobrescribirá los resultados de la pestaña 1).

Nota: este método de evaluación no revisa la estructura del archivo XML para comprobar si se ajusta al esquema. De ese tipo de comprobación se encarga la función de validación.

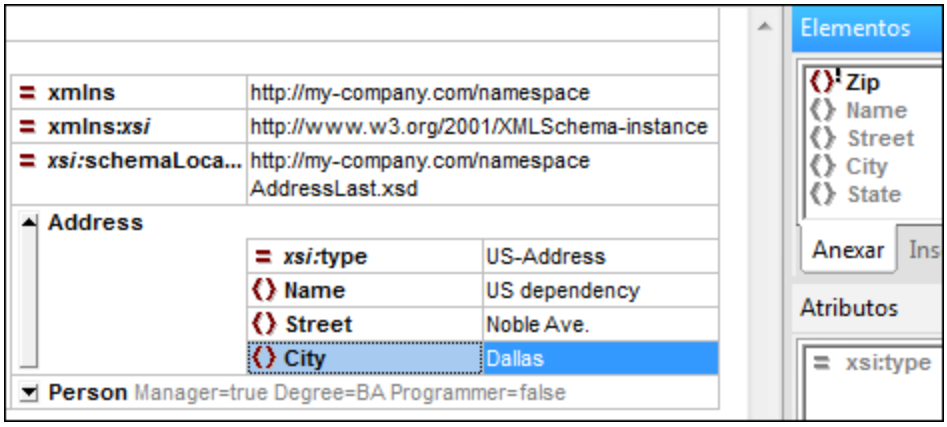
Validación del documento

Un documento XML es válido conforme a un esquema si se ajusta a la estructura y al contenido especificados en dicho esquema. Puede comprobar si un documento es válido en cualquiera de las vistas de edición. Para validar un documento:

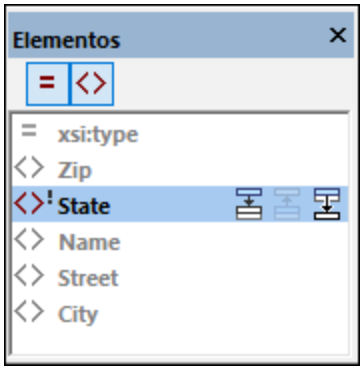
1. Seleccione la vista Cuadrícula, por ejemplo.
2. Seleccione la opción de menú **XML | Validar el documento XML** o pulse la tecla **F8.** (También puede usar el icono correspondiente de la barra de herramientas). La ventana "Mensajes" muestra un mensaje de error que dice que el archivo no es válido. En el elemento `Address` faltan elementos obligatorios después del elemento `City`. Si volvemos al esquema veremos que el tipo complejo `US-Address` (que se asignó al elemento `Address` al definirse como valor del atributo `xsi:type`) tiene un modelo de contenido en el que el elemento City debe estar seguido de un elemento `Zip` y un elemento `State`.

Corregir el documento no válido

La ubicación del error aparece resaltada en la vista, junto con un indicador de error y una solución inteligente 335. El elemento no válido en este caso es **`Address`**. Si hace clic en el icono de la solución inteligente aparece esta información emergente: *Agregar el elemento <Zip> que falta con contenido de muestra*. En el esquema verá que el elemento `Address/City` debe ir seguido por el elemento obligatorio `Zip`. Para comprobar si esto se cumple seleccione el elemento `City` y busque el ayudante de entrada Elementos del mismo nivel. Verá que el elemento `Zip` viene precedido de un signo de exclamación, lo que indica que es un elemento obligatorio en este contexto.

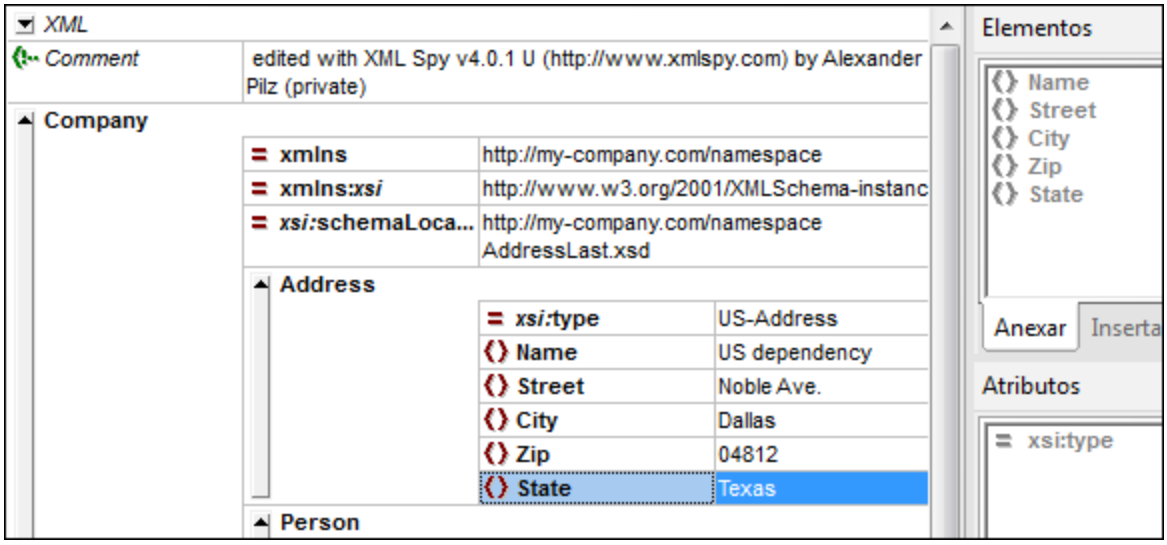


Ahora haga clic en la solución inteligente (*imagen anterior*). Así se añade el elemento `Zip`, que contendrá contenido de muestra para que sea válido. Introduzca el código postal correcto (por ejemplo, **`04812`** para Dallas). Vuelva a observar el ayudante de entrada Elementos del mismo nivel. Verá que ahora muestra que el elemento `State` es obligatorio (como marca el signo de exclamación). Si selecciona el elemento `State` se habilitan las opciones correspondientes del ayudante de entrada (*imagen siguiente*). Estas opciones son las acciones que puede insertar en el elemento `State` que está seleccionado actualmente en la ventana principal (que es `City`) o anexar a `State` después de los elementos del mismo nivel que `City`.



Como en este caso las dos acciones tienen el mismo efecto puede seleccionar cualquiera de ellas. Así se añade un elemento `State` después de `City`. Haga doble clic dentro del campo de contenidos de `State` e introduzca el nombre del estado, `Texas` (*imagen siguiente*). Observe que ahora todos los elementos que contiene el ayudante de entrada Elementos del mismo nivel aparecen atenuados, lo que indica que no quedan elementos obligatorios que añadir.

© 2016-2022 Altova GmbH

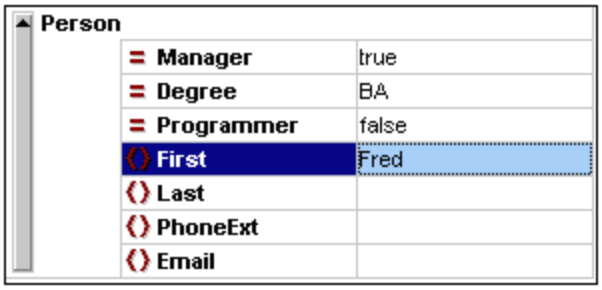


Terminar el documento y volver a validarlo

Ahora vamos a terminar el documento (introducir los datos para el elemento `Person`) antes de volver a validarlo.

Siga estos pasos:

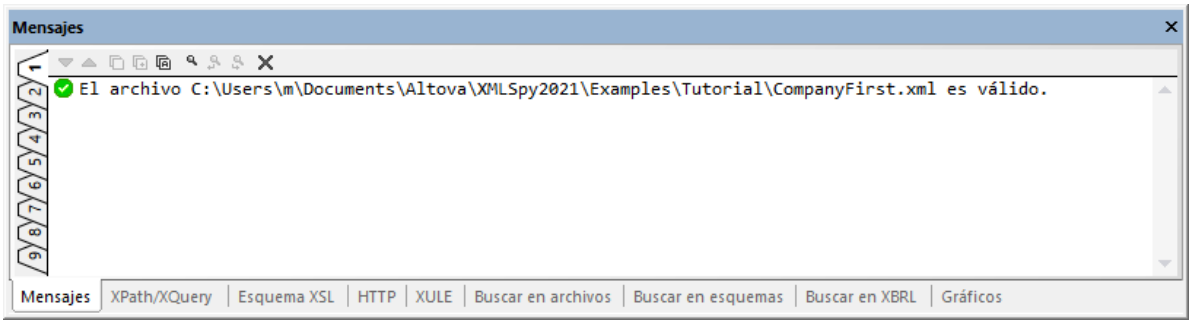
1. Haga clic en el campo de valor de `First` y escriba un nombre propio (p. ej. `Fred`). Después pulse la tecla **Entrar**.



2. Siga los mismos pasos para introducir los datos de todos los elementos secundarios de `Person`, es decir, de los elementos `Last`, `PhoneExt` y `Email`. Puede usar el tabulador para pasar de una celda a otra. Recuerde que el valor del elemento `PhoneExt` debe ser un número entero con un valor máximo de 99 (este es el intervalo permitido para los valores de `PhoneExt` que definimos en el esquema). Después de introducir los datos este es el aspecto que tendrá el documento XML en la vista Cuadrícula:

Company		
	xmlns	http://my-company.com/namespace
	xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
	xsi:schemaLoca...	http://my-company.com/namespace C:\PROGRA~1\Altova\XMLSpy2006\Examples\Tutorial\AddressLast.xsd
Address		
	xsi:type	US-Address
	Name	US Dependency
	Street	Noble Ave.
	City	Dallas
	Zip	04812
	State	Texas
Person		
	Manager	true
	Degree	BA
	Programmer	false
	First	Fred
	Last	Smith
	PhoneExt	22
	Email	Smith@work.com

3. Haga clic en el icono  otra vez para comprobar si el documento es válido. En la ventana "Mensajes" aparece un mensaje que dice que el archivo es válido. Ahora el documento XML es válido con respecto al esquema.



4. Guarde el documento con **Archivo | Guardar**.

Nota: para guardar un documento XML no es necesario que sea válido. Si guarda un documento no válido aparece un mensaje de aviso advirtiéndole de que se dispone a guardar un documento no válido. Si desea guardar el documento en su estado no válido, seleccione la

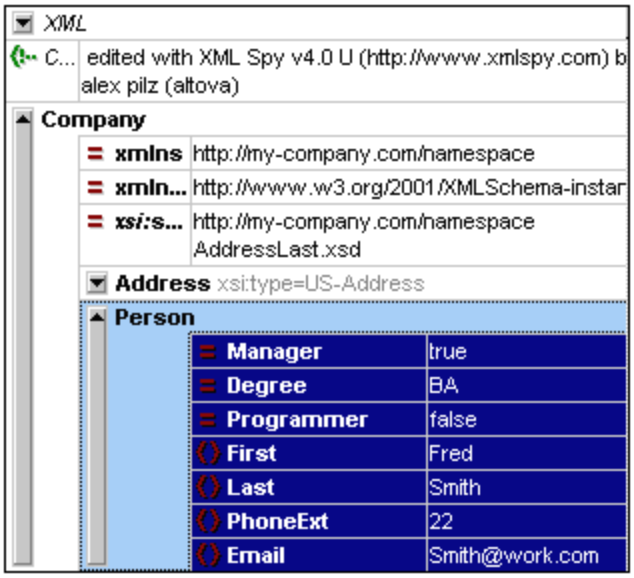
2.5.6 Agregar elementos y atributos

Llegados a este punto el documento contiene un solo elemento `Person`.

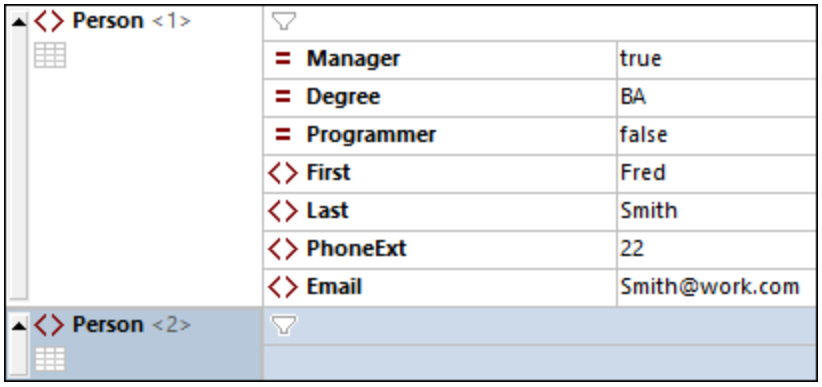
Para agregar un elemento `Person` nuevo:

© 2016-2022 Altova GmbH

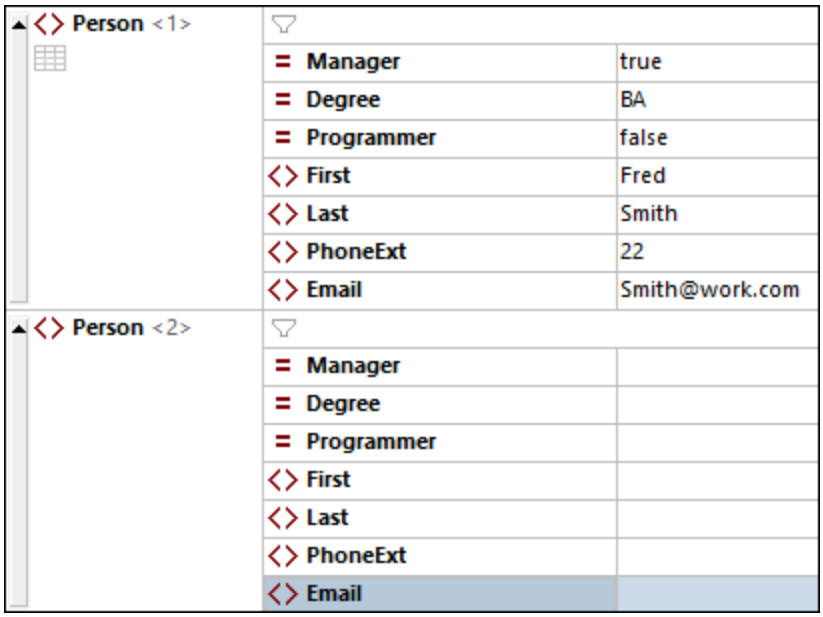
1. Haga clic en la barra lateral gris situada a la izquierda del elemento `Address` para contraer el elemento `Address`. Así despejamos un poco la vista.
2. En la vista Cuadrícula haga clic en el nombre del elemento `Person` para seleccionar todo el elemento. Observe que el elemento `Person` está disponible en el ayudante de entrada "Elementos del mismo nivel".



3. Seleccione el elemento **`Person`** en el ayudante de entrada Elementos del mismo nivel y haga clic en **Insertar después** o en **Anexar** para anexar un elemento **`Person`** nuevo (*imagen siguiente*).



4. Al seleccionar un elemento **`Person`** verá los atributos y elementos secundarios disponibles en el ayudante de entrada Elementos secundarios. Haga doble clic en esos atributos y elementos secundarios para añadir los mismos nodos secundarios que en el elemento **`Person`**. Si, en la ventana principal, selecciona uno de los elementos secundarios añadidos, puede añadir más elementos secundarios del elemento **`Person`** de dos maneras: (i) seleccione el elemento **`Person`** y añada un elemento secundario nuevo desde el ayudante de entrada Elementos secundarios, (ii) seleccione el elemento secundario que ha añadido añada un elemento del mismo nivel desde el ayudante de entrada Elementos del mismo nivel. Los nodos secundarios de **`Person`** que ya se han añadido aparecerán atenuados en los dos ayudantes de entrada.

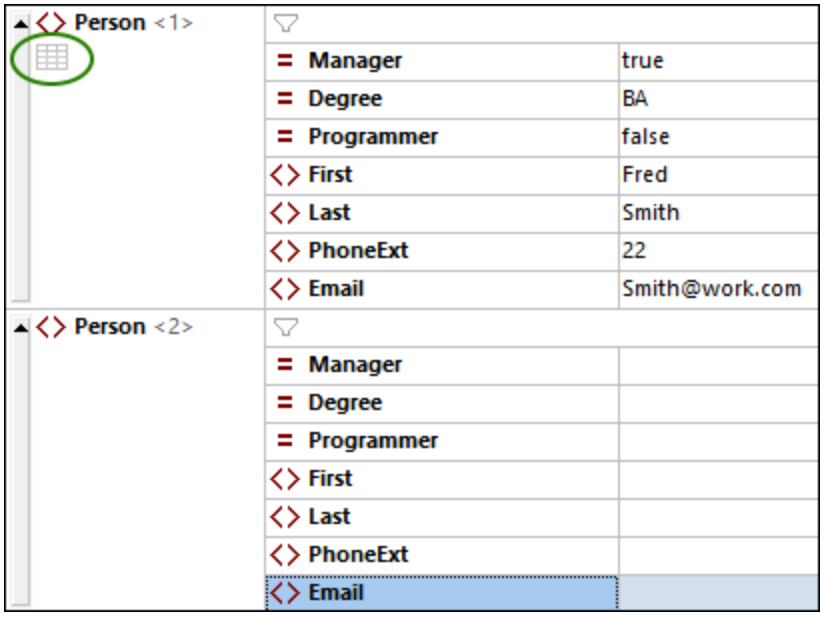


Podríamos añadir contenido para los nodos secundarios del elemento **Person** en la vista Cuadrícula, pero vamos a cambiar a la vista en forma de tabla de la vista Cuadrícula porque es más adecuada para editar una estructura con varias ocurrencias, como es el caso de **Person**.

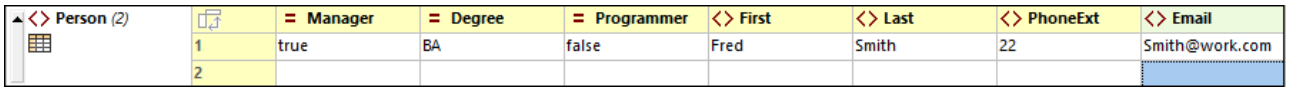
2.5.7 Edición en la vista tabular y de base de datos

La vista Cuadrícula contiene la vista llamada vista Diseño de tabla, que se usa para editar elementos que aparecen varias veces. Por ejemplo, el elemento **Person** aparece varias veces (*imagen siguiente*), por lo que se puede mostrar como tabla. Para mostrar este tipo de elementos como tabla haga clic en el icono **Vista Diseño de tabla** de la primer aparición del elemento. Por ejemplo, en la imagen siguiente el icono **Vista Diseño de tabla** del elemento **Person** aparece en verde. (También puede usar el comando de menú **XML | Mostrar como tabla** o el icono correspondiente de la barra de herramientas de la vista Cuadrícula 293.)

 © 2016-2022 Altova GmbH



Al hacer clic en el icono de la vista Diseño de tabla el elemento **Person** aparece como tabla. En la vista Diseño de tabla los nodos secundarios del elemento (sus atributos y elementos) aparecen como columnas, mientras que los elementos **Person** aparecen como filas (*imagen siguiente*).



Ventajas de la vista Diseño de tabla
La vista Diseño de tabla ofrece estas ventajas:

- Puede arrastrar y colocar encabezados de columna para cambiar de posición las columnas relacionadas entre sí. En el documento XML esto se traduce en un cambio de la posición relativa de los nodos secundarios de todas las apariciones de un elemento (que se corresponden con las filas de la tabla).
- Las tablas, así como las apariciones del elemento que representan, se pueden ordenar (en orden ascendente o descendente) en función del contenido de cualquiera de las columnas. Para ello use el comando de menú **XML | Orden ascendente/descendente**.
- Las demás filas (es decir, apariciones del elemento) se pueden anexar o insertar rápidamente con los comandos del menú **XML**. La ventaja es que no solo se añade un elemento nuevo, sino que todos sus elementos secundarios aparecen representados como columnas en la tabla.
- Puede copiar y pegar *datos estructurados* en y desde otros productos, como Microsoft Excel.
- Las funciones de edición inteligente de XMLSpy también están disponibles en la vista Diseño de tabla.

Mostrar como tabla un elemento que aparece varias veces

Para visualizar el tipo de elemento **Person** en forma de tabla siga estos pasos:

1. Haga clic en el icono **Vista Diseño de tabla** de la primera aparición del elemento **Person**, como se describe más arriba.

2. Seleccione la opción de menú **Vista | Ancho de tabla óptimo** o use el icono **Ancho de tabla óptimo** de la barra de herramientas de la vista Cuadrícula[293].

Nota: la vista Diseño de tabla se puede activar y desactivar para todos los elementos que aparezcan varias veces. Sin embargo, tenga en cuenta que los elementos secundarios que estén representados en forma de tabla seguirán representados de esa manera.

Introducir contenidos en la vista Diseño de tabla

Para introducir contenidos en el segundo elemento **Person**, haga doble clic en cada una de las celdas de la segunda fila y escriba los datos. Las funciones de edición inteligente también se pueden usar dentro de las celdas de una tabla, por lo que puede seleccionar opciones de las listas desplegables donde las haya (por ejemplo, en el contenido booleano y las enumeraciones del atributo **Degree**).

	= Manager	= Degree	= Programmer	<> First	<> Last	<> PhoneExt	<> Email
1	true	BA	false	Fred	Smith	22	Smith@work.com
2	false	MA	true	Alfred	Aldrich	33	Aldrich@work.com

Validación dinámica

Observe que, tal y como viene definido en el esquema, **PhoneExt** debe ser un número entero (hasta 99) para que el archivo sea válido. Puede activar y desactivar la función de XMLSpy que valida el documento mientras lo edita. Cuando esta opción está activada, el archivo se valida cada vez que se selecciona un nodo nuevo. Si quiere probar la función de validación dinámica:

1. Active el comando de menú **XML | Validar al editar**.
2. Introduzca un valor **PhoneExt** válido (cualquier valor mayor que **99**), como se ve en la imagen siguiente.
3. Pulse la tecla **Tabulador**. Verá que en la celda **PhoneExt** aparecen un icono de error y un icono de solución inteligente (*imagen siguiente*).
4. Pase el cursor del ratón sobre el icono de error para ver el mensaje de error de validación (*imagen siguiente*).



5. Haga clic en el icono de solución inteligente y después en la opción que aparece como información emergente. El valor no válido se sustituye por uno válido y el indicador de error desaparece.

Copiar y pegar datos XML en hojas de estilo

En la vista Diseño de tabla puede copiar datos como texto separado por tabulador para poder usarlo también en aplicaciones de hojas de estilos como MS Excel. Siga estos pasos para copiar datos desde un archivo XML:

1. Haga clic en el elemento **Person** (*imagen siguiente*). Esta acción selecciona tanto los encabezados de columna como las dos filas de la tabla.

 © 2016-2022 Altova GmbH

Address	<Address xsi:type="US-Address"><Name>US dependency</Name><Street>Noble Ave.</Street><City>Dallas</City>						
Person (2)	= Manager	= Degree	= Programmer	First	Last	PhoneExt	Email
1	true	BA	false	Fred	Smith	22	Smith@work.com
2	false	MA	true	Alfred	Aldrich	33	Aldrich@work

2. Haga clic con el botón derecho dentro de la selección y seleccione el comando **Copiar | Copiar como texto separado por tabulador** en el menú contextual que aparece. También puede pulsar **Ctrl+C**.
3. Ahora cambie a la hoja de estilos de Excel, seleccione la celda A1 y pegue (**Ctrl+V**) los datos XML. Al pegarse, los datos se introducen como filas que se corresponden con la estructura de tabla de la vista Diseño de tabla (*imagen siguiente*).

	A	B	C	D	E	F	G	H
1	Manager	Degree	Programm	First	Last	PhoneExt	Email	
2	TRUE	BA	FALSE	Fred	Smith	22	Smith@work.com	
3	FALSE	MA	TRUE	Alfred	Aldrich	33	Aldrich@work	
4								

4. Añada una fila nueva de datos en Excel, como se ve en la imagen siguiente. Asegúrese de escribir un número de tres dígitos para el elemento `PhoneExt` (`444`, por ejemplo).
5. Seleccione los datos de la tabla en Excel, sin incluir los encabezados de las columnas (*en verde en la imagen siguiente*), y cópielos con **Ctrl+C**.

	A	B	C	D	E	F	G	H
1	Manager	Degree	Programm	First	Last	PhoneExt	Email	
2	FALSE	MA	TRUE	Alfred	Aldrich	33	Aldrich@work	
3	TRUE	BA	FALSE	Fred	Smith	22	Smith@work.com	
4	TRUE	Ph.D	FALSE	Colin	Coletti	444	Coletti@work.com	
5								

6. En XMLSpy asegúrese de que tiene activado el comando **XML | Validar al editar**.
7. En XMLSpy, en la vista Diseño de tabla del documento XML, seleccione la celda `Manager` de la primera fila y pegue el contenido del portapapeles con **Ctrl+V**. La tabla nueva debería tener un aspecto parecido al de las imágenes siguientes.

Address	<Address xsi:type="US-Address"><Name>US dependency</Name><Street>Noble Ave.</Street><City>Dallas</City>						
Person (3)	= Manager	= Degree	= Programmer	First	Last	PhoneExt	Email
1	FALSE	MA	TRUE	Alfred	Aldrich	33	Aldrich@work
2	TRUE	BA	FALSE	Fred	Smith	22	Smith@work.com
3	TRUE	Ph.D	FALSE	Colin	Coletti	444	Coletti@work.com

8. Los errores de validación de los valores booleanos se deben a las diferencias entre mayúsculas y minúsculas en XML y Excel. Para solucionar este problema aplique las soluciones inteligentes de las celdas de tabla correspondientes.

Ordenar la tabla en base al contenido de una columna

Puede ordenar una tabla de la vista Diseño de tabla en orden ascendente o descendente según el contenido de las columnas. En este caso, ordenaremos la tabla `Person` según los apellidos. Para ello siga estas instrucciones:

1. Seleccione la columna **Last** (haciendo clic en el encabezado).

Person (3)	Manager	Degree	Programmer	First	Last	PhoneExt	Email
1	true	BA	false	Fred	Smith	22	Smith@work.com
2	false	MA	true	Alfred	Aldrich	33	Aldrich@work
3	true	Ph.D	false	Colin	Coletti	444	Coletti@work.com

2. Seleccione la opción de menú **XML | Orden ascendente** o use el icono **Orden ascendente** de la barra de herramientas de la vista Cuadrícula 293. La columna, y con ella la tabla entera, se ordenarán por orden alfabético. La columna sigue estando resaltada.

Person (3)	Manager	Degree	Programmer	First	Last	PhoneExt	Email
1	false	MA	true	Alfred	Aldrich	33	Aldrich@work
2	true	Ph.D	false	Colin	Coletti	444	Coletti@work.com
3	true	BA	false	Fred	Smith	22	Smith@work.com

Como la extensión de teléfono **444** es correcta pero no válida, debe modificar el esquema XML para que este número sea válido. En el apartado siguiente explicamos cómo hacerlo.

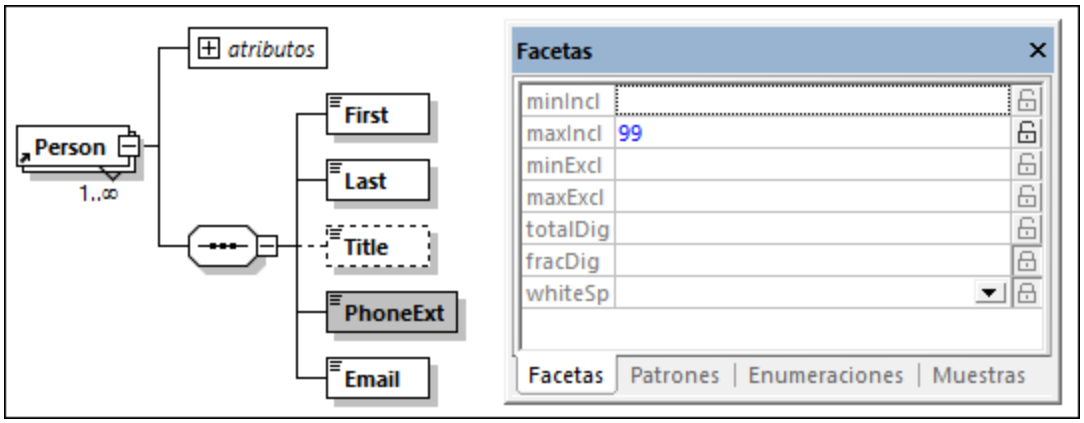
2.5.8 Modificar el esquema

Puesto que la extensión de teléfono que tenemos está fuera del rango definido en el esquema XML (**0** a **999**) vamos a ampliar ese rango a **999**. Para ello siga estas instrucciones:

1. En la vista Cuadrícula seleccione cualquiera de las celdas **PhoneExt** (*imagen siguiente*).

Person (3)	Manager	Degree	Programmer	First	Last	PhoneExt	Email
1	false	MA	true	Alfred	Aldrich	33	Aldrich@work
2	true	Ph.D	false	Colin	Coletti	444	Coletti@work.com
3	true	BA	false	Fred	Smith	22	Smith@work.com

2. Seleccione la opción de menú **DTD/Esquema | Ir a la definición** o haga clic en el icono "Ir a la definición" de la barra de herramientas de la vista Cuadrícula 293. El esquema asociado, en este caso **AddressLast.xsd**, está abierto y la definición **PhoneExt** aparece resaltada (*imagen siguiente*).



 © 2016-2022 Altova GmbH

3. La faceta **maxIncl** del elemento es **99**, como puede ver en la imagen. Cambie este valor a **999** y guarde el esquema.
4. Vuelva al documento XML e intente validarlo. Ahora será válido.
5. Guarde el archivo como **CompanyLast.xml**.

Nota: la carpeta Tutorial de XMLSpy ya contiene un archivo llamado **CompanyLast.xml** con los mismos datos que el que guardará al terminar este tutorial.

2.6 Transformaciones XSLT

Objetivo

Generar un archivo HTML a partir del archivo XML usando una hoja de estilos XSL para transformar el archivo XML. Tenga en cuenta que una *transformación* no convierte el archivo XML en otro tipo de archivo, sino que se genera un archivo de salida nuevo. La palabra *transformación* es simplemente un término usado por consenso.

Método

El método utilizado para realizar la transformación es este:

- Asignar al documento XML un archivo XSL predefinido (`Company.xsl`).
- Ejecutar la transformación en la interfaz de XMLSpy con ayuda de uno de los dos motores XSLT integrados de Altova. (*Ver nota al final de esta página.*)

Comandos utilizados en esta sección

En esta sección utilizaremos estos comandos de XMLSpy:

	**XSL/XQuery	Asignar XSL**, que asigna un archivo XSL al documento XML activo.
	**XSL/XQuery	Ir al archivo XSL**, que abre el archivo XSL al que hace referencia el documento XML activo.
	**XSL/XQuery	Transformación XSL (F10)**, o su icono de la barra de herramientas, que transforma el documento XML activo utilizando la hoja de estilos XSL asignada al archivo XML. Si no se asignó un archivo XSL, cuando seleccione este comando se le pedirá que asigne uno.

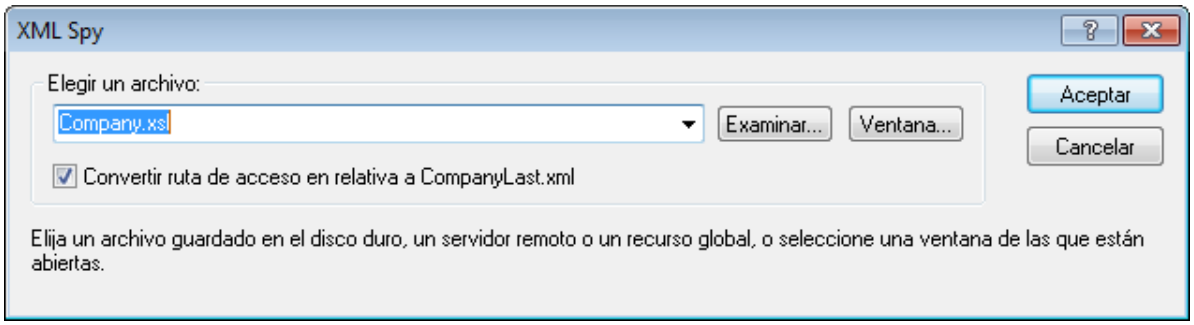
Nota: XMLSpy incluye motores XSLT integrados para XSLT 1.0, 2.0 y 3.0. El motor adecuado se selecciona automáticamente en función del atributo `version` del elemento `xsl:stylesheet` o `xsl:transform`. Para la transformación del tutorial utilizamos hojas de estilos XSLT 3.0. El motor XSLT 3.0 se seleccionará automáticamente para las transformaciones que usen estas hojas de estilos cuando se invoca el comando **Transformación XSL**.

2.6.1 Asignar un archivo XSLT

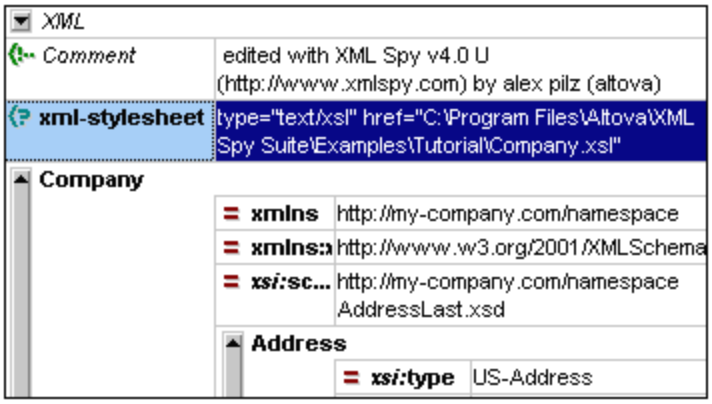
Para asignar un archivo XSLT al archivo `CompanyLast.xml`:

1. Haga clic en la pestaña `CompanyLast.xml` de la ventana principal para que este sea el documento activo y cambie a la vista Texto.
2. Seleccione el comando **XSL/XQuery | Asignar XSL**.
3. Haga clic en el botón **Examinar** y seleccione el archivo `Company.xsl` ubicado en la carpeta Tutorial. En el cuadro de diálogo puede activar la opción Convertir esta ruta de acceso en relativa a `CompanyLast.xml` si desea convertir la ruta al archivo XSL (en el documento XML) en relativa.

© 2016-2022 Altova GmbH



4. Haga clic en **Aceptar** y el archivo XSL se asignará al documento XML.
5. Cambie a la vista Cuadrícula para ver la asignación (*ver imagen siguiente*).

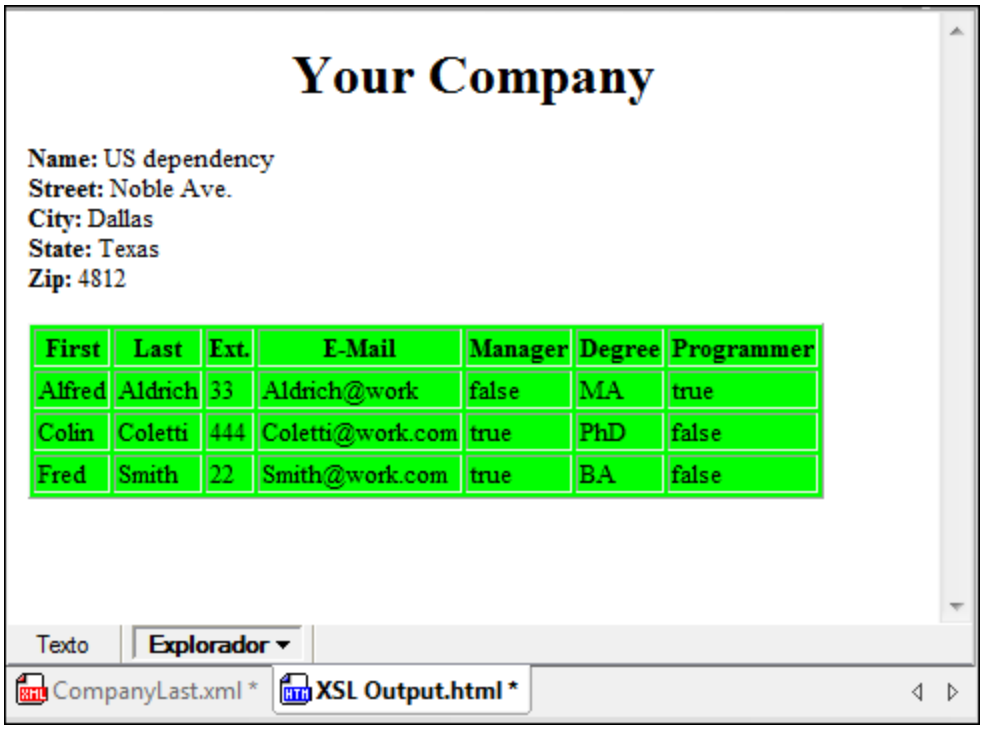


En el documento XML se insertó una instrucción de procesamiento `XML-stylesheet` que hace referencia al archivo XSL. Si marcó la casilla *Convertir esta ruta de acceso en relativa a* `CompanyLast.xml`, la ruta de acceso será relativa; de lo contrario será absoluta.

2.6.2 Transformar el archivo XML

Para transformar el documento XML usando el archivo XSL que le asignó, siga estos pasos:

1. Asegúrese de que el archivo XML es el documento activo.
2. Seleccione la opción de menú **XSL/XQuery | Transformación XSL (F10)** o pulse el icono correspondiente en la barra de herramientas. Inmediatamente después se inicia la transformación con la hoja de estilos XSLT a la que hace referencia el documento XML. El documento de salida se muestra en la vista Explorador y lleva el nombre XSL `Output.html`. El documento HTML muestra los datos de `Company` en un bloque de texto a la izquierda de la página y los datos de `Person` en forma de tabla justo debajo del bloque de texto.



Nota: Puesto que el archivo `Company.xsl` es un documento XSLT 3.0, se selecciona automáticamente el motor integrado XSLT 3.0 de Altova para procesar esta transformación. Si el archivo de salida HTML no se genera, asegúrese de que en la sección *XSL* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**), la extensión de archivo predeterminada del archivo de salida es `.html`. De esta forma se asegura de que el explorador lee el documento de salida correctamente, es decir, como archivo HTML.

2.6.3 Modificar el archivo XSL

Si lo desea, puede cambiar el diseño de salida con sólo cambiar el documento XSL. Por ejemplo, vamos a cambiar el color de fondo verde lima de la tabla a amarillo.

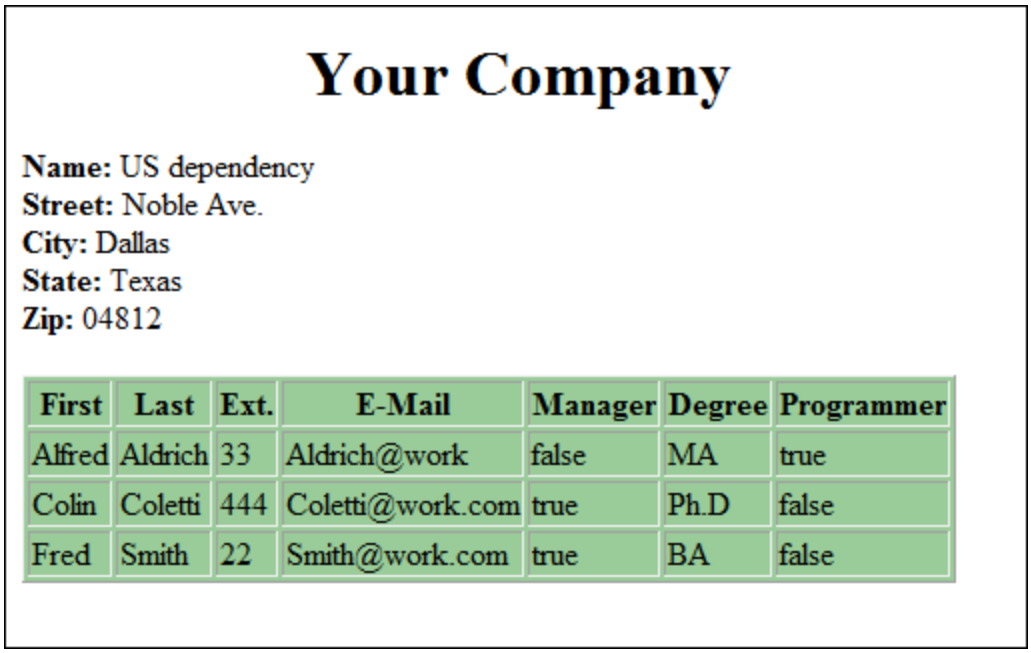
Siga estos pasos:

1. Haga clic en la pestaña `CompanyLast.xml` para que éste sea el documento activo y cambie a la vista Cuadrícula.
2. Seleccione la opción de menú **XSL/XQuery | Ir al archivo XSL**. Este comando abre el archivo `Company.xsl` al que se hace referencia en el documento XML.
3. Busque la etiqueta del elemento **`table`** y después el atributo **`bgcolor`** del elemento (*resaltado en la imagen siguiente*). Cambie el valor del atributo de **`#ccccff`** a **`#99cc99`**.

 © 2016-2022 Altova GmbH

```
<xsl:template match="/">
   <html>
      <head><title>Your company</title></head>
         <body>
            <h1><center>Your Company</center></h1>
            <xsl:apply-templates select="//my:Address"/>
               <table border="1" bgcolor="#ccccff">
                  <thead align="center">
                     <td><strong>First</strong></td>
                     <td><strong>Last</strong></td>
                     <td><strong>Ext.</strong></td>
                     <td><strong>E-Mail</strong></td>
                     <td><strong>Manager</strong></td>
                     <td><strong>Degree</strong></td>
                     <td><strong>Programmer</strong></td>
                  </thead>
                  <xsl:apply-templates select="//my:Person"/>
               </table>
         </body>
   </html>
</xsl:template>
```

4. Seleccione la opción de menú **Archivo | Save** para guardar los cambios.
5. Haga clic en la pestaña **CompanyLast.xml** para que el archivo activo sea el archivo XML.
6. Ejecute el comando de menú **XSL/XQuery | Transformación XSL** o pulse la tecla **F10**. En la vista Explorador aparece un archivo **XSL Output.html** nuevo con la tabla con el color de fondo cambiado (*imagen siguiente*).



Your Company

Name: US dependency
Street: Noble Ave.
City: Dallas
State: Texas
Zip: 04812

First	Last	Ext.	E-Mail	Manager	Degree	Programmer
Alfred	Aldrich	33	Aldrich@work	false	MA	true
Colin	Coletti	444	Coletti@work.com	true	Ph.D	false
Fred	Smith	22	Smith@work.com	true	BA	false

7. Seleccione la opción de menú **Archivo | Guardar** y guarde el documento de salida como **Company.html**.

2.7 Gestión de proyectos

Esta sección ofrece una introducción a las funciones de gestión de proyectos de XMLSpy. Empezaremos por explicar las ventajas que supone organizar sus archivos XML en proyectos y después aprenderá a organizar los archivos que creó en las secciones anteriores en un sencillo proyecto.

2.7.1 Ventajas de trabajar con proyectos

Estas son algunas de las ventajas de organizar sus archivos XML en proyectos:

- Puede agrupar archivos en carpetas en base a un criterio en común. Por ejemplo, puede separar los archivos XML y XSD en dos carpetas distintas. Puede crear distintas jerarquías.
- Puede definir distintas propiedades para cada carpeta. Por ejemplo, puede asignar un esquema de validación a una carpeta que contiene archivos XML. Eso permite validar en lote todos los archivos de esta carpeta con respecto al esquema. También puede asignar un esquema distinto a esa misma carpeta y realizar una segunda validación en lote. Otras propiedades permiten, por ejemplo, realizar transformaciones en lote con un único archivo XSLT.
- Puede aplicar procesamiento por lotes a las distintas carpetas o al proyecto entero.
- Puede asignar una DTD o un esquema XML a cada carpeta para validar los archivos de la carpeta.
- Puede asignar archivos XSLT a cada carpeta para transformar los archivos XML de la carpeta con el archivo XSLT asignado.
- Se pueden especificar carpetas de destino de los archivos de transformación XSLT para una carpeta entera.

Puede acceder a todas las opciones de configuración del proyecto desde la opción de menú **Proyecto | Propiedades**. Estos comandos también se encuentran en los menús contextuales del proyecto y de las carpetas del proyecto. En la siguiente sección aprenderá a crear un proyecto desde el menú **Proyecto**.

Además, XMLSpy ofrece estas funciones avanzadas de proyecto:

- Los archivos XML se pueden poner bajo control de código fuente gracias a la opción de menú **Proyecto | Control de código fuente | Agregar al control de código fuente**. (Para más información consulte la sección sobre Control de código fuente[1302] de la ayuda en línea.)
- Puede agregar carpetas externas[1319] y carpetas web[1322] a sus proyectos. Esto permite aplicar todas las características de las carpetas del proyecto, como la validación o las transformaciones, a carpetas de su red o de Internet.

2.7.2 Crear un proyecto

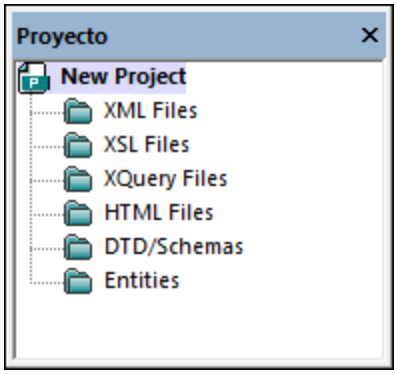
Llegados a este punto del tutorial ya tendrá varios archivos abiertos en la ventana principal. Puede agrupar todos estos archivos en un proyecto de tutorial. Para eso debe crear primero un proyecto nuevo y después añadir los archivos del tutorial a las subcarpetas correspondientes.

Crear un proyecto nuevo

Estos son los pasos para crear el proyecto:

© 2016-2022 Altova GmbH

1. Seleccione la opción de menú **Proyecto | Proyecto nuevo**. A continuación se crea una carpeta de proyecto nueva llamada **`New Project`** en la ventana "Proyecto" (*imagen siguiente*). El proyecto nuevo contiene carpetas vacías para los tipos de archivos XML que se suelen usar en un proyecto.



2. Haga clic en la pestaña **`CompanyLast.xml`** para que el archivo **`CompanyLast.xml`** sea el archivo activo de la ventana principal.
3. Seleccione la opción de menú **Proyecto | Agregar archivos activos y relacionados** al proyecto. Se añaden dos archivos al proyecto: **`CompanyLast.xml`** y **`AddressLast.xsd`**. El archivo XML es el archivo activo, por lo que se añade a la subcarpeta *XML*. El archivo del esquema se añade a la carpeta *DTD/Schemas* porque es un archivo relacionado, ya que se hace referencia a él desde el archivo XML. Tenga en cuenta que los archivos a los que se hace referencia desde archivos de instrucciones de procesamiento, como los archivos XSLT, no cuentan como archivos relacionados.
4. Seleccione la opción de menú **Proyecto | Guardar proyecto** y guárdelo con el nombre **`Tutorial`**.

Nota: todas las carpetas (pero no el proyecto) tienen una propiedad llamada *Extensiones de archivo*. Esta propiedad consiste en una lista de propiedades separadas por punto y coma (por ejemplo **`xml;svg;wml`**). Esa lista determina qué archivos se añaden a qué carpetas al añadir archivos al proyecto. Por ejemplo, si añade archivos activos y relacionados a un proyecto, como acabamos de hacer, las propiedades *Extensiones de archivo* de las carpetas determinan en qué carpeta va cada uno de los archivos.

El proyecto y las propiedades de las carpetas

Las propiedades (como el esquema de validación y el archivo XSLT de las transformaciones) no solo se pueden definir a nivel del proyecto, sino también para carpetas individuales. Puede ejecutar acciones como la validación o la transformación a nivel del proyecto o de una sola carpeta. Para ello haga clic con el botón derecho en el proyecto o la carpeta y seleccione la acción que quiere ejecutar en el menú contextual que aparece.

Tenga en cuenta que:

- La propiedad que defina a nivel de la carpeta sobrescribe esa misma propiedad a nivel del proyecto.
- Si define una propiedad a nivel del proyecto, esta se aplica a todas las carpetas para las que no se haya definido esa propiedad.
- Si lleva a cabo una acción en un proyecto, esa acción se aplica a todos los tipos de archivos pertinentes en todas las carpetas del proyecto. Por ejemplo, si ejecuta una validación a nivel de un proyecto, esta se aplica a todos los archivos XML de todas las carpetas del proyecto. En este caso, el esquema que definió para el proyecto se usa para todas las validaciones excepto para los archivos XML en cuyas carpetas se haya definido un esquema de validación distinto.

Agregar archivos al proyecto

Si lo desea, puede agregar más archivos al proyecto. Para ello siga estas instrucciones:

1. Haga clic en uno de los archivos XML que están abiertos (que tengan la extensión de archivo **.xml**) y que no sea el archivo **CompanyLast.xml** para que ese archivo XML sea el archivo activo. (Si no tiene abiertos más archivos XML, abra uno o cree uno nuevo.)
2. Seleccione la opción de menú **Proyecto | Agregar archivo activo al proyecto**. El archivo XML se añade a la carpeta *XML Files* en base a su extensión de archivo **.xml**.
3. Ahora repita este paso para añadir un archivo HTML y uno XSD (por ejemplo los archivos **Company.html** y **AddressFirst.xsd**) al proyecto. Estos archivos se agregarán a las carpetas *HTML Files* y *DTD/Schemas* respectivamente.
4. Guarde el proyecto. Para ello seleccione la opción de menú **Proyecto | Guardar proyecto** o seleccione cualquier archivo o carpeta en la ventana Proyecto y haga clic en el icono **Guardar** de la barra de herramientas (o bien use **Archivo | Guardar**)

Nota: otra opción es hacer clic con el botón derecho en una carpeta del proyecto y seleccionar **Agregar archivo activo** para añadir el archivo activo a esta carpeta en concreto.

Otros comandos útiles

- Para agregar una carpeta nueva al proyecto, seleccione la opción de menú **Proyecto | Agregar carpeta de proyecto al proyecto** e inserte el nombre de la carpeta de proyecto.
- Para eliminar una carpeta de un proyecto, haga clic con el botón derecho en la carpeta y seleccione la opción **Eliminar** del menú contextual.
- Para eliminar un archivo de un proyecto, seleccione el archivo y pulse la tecla **Suprimir**.

© 2016-2022 Altova GmbH

2.8 ¡Eso es todo!

¡Enhorabuena y muchas gracias por llegar hasta aquí!

Esperamos que este tutorial le sea útil y le sirva para comprender los aspectos básicos de XMLSpy. Ahora ya puede seguir trabajando con XML usando XMLSpy como editor. Si busca información sobre alguna funcionalidad en concreto puede usar el Índice o la función de búsqueda de este manual. También puede utilizar el sistema de ayuda contextual o imprimir la versión en PDF de este tutorial (el archivo `Tutorial.pdf` ubicado en la carpeta de aplicación 36 de XMLSpy).

3 Interfaz y entorno

Esta introducción describe:

- la interfaz gráfica del usuario (IGU) 119 y
- el entorno de aplicación 136.

La sección dedicada a la IGU 119 empieza con una introducción sobre la IGU y después describe en detalle las diferentes ventanas de la interfaz. También explica como cambiar el tamaño de las ventanas, moverlas y cómo trabajar con ellas en general.

La sección dedicada al entorno de aplicación 136 describe las opciones de configuración con las que puede controlar cómo se visualizan y editan los diferentes tipos de archivo. También explica cómo se puede personalizar la aplicación. Además aprenderá dónde se guardan los archivos de ejemplo importantes y el tutorial. Al final de la sección, encontrará enlaces al sitio web de Altova, donde podrá consultar una lista completa con todas las funciones y características de la aplicación. En el sitio web también encontrará el *Manual del usuario* en diferentes formatos, información sobre las opciones de soporte técnico disponibles y detalles sobre los demás productos de Altova.

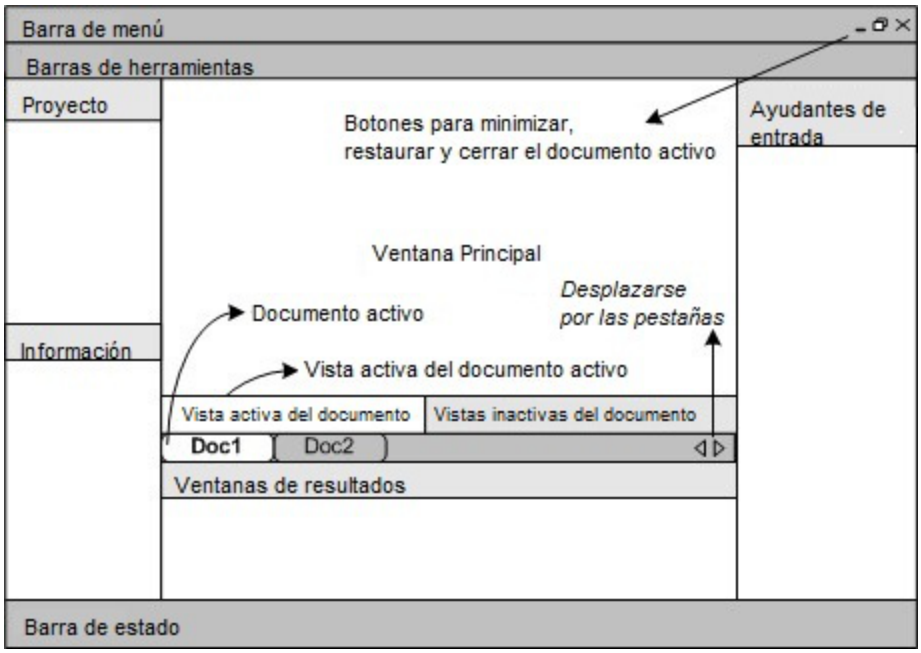
© 2016-2022 Altova GmbH

3.1 La interfaz gráfica de usuario (IGU)

La interfaz gráfica de usuario (IGU) incluye una ventana principal y varias barras laterales (*imagen siguiente*). Las barras laterales están situadas por defecto alrededor de la ventana principal y están ordenadas en varios grupos:

- Ventana Proyecto
- Ventana Información
- Ayudantes de entrada: Elementos, Atributos, Entidades, etc. (dependiendo del tipo de documento que esté activo)
- Ventanas de resultados: Mensajes, XPath/XQuery, Esquema XSL, HTTP, Buscar en archivos, Buscar en esquemas, Buscar en XBRL y Gráficos

Los diferentes apartados de esta sección describen en detalle la ventana principal y las barras laterales.



La IGU también incluye una barra de menú, una barra de estado y barras de herramientas, todas ellas explicadas en Barra de menú, barras de herramientas y barra de estado 134..

Activar o desactivar las barras laterales

Los grupos de barras laterales (ventana Proyecto, ventana Información, ayudantes de entrada y ventanas de resultados) se pueden mostrar u ocultar con solo pulsar los comandos correspondientes del menú **Ventanas**. Además puede ocultar las barras laterales (o un grupo de barras laterales organizadas en pestañas) si hace clic con el botón derecho sobre el título de la barra y selecciona el comando **Ocultar**.

Si cierra una de las ventanas de resultados puede volver a abrirla haciendo clic en el menú contextual **Ventanas | Ventanas de resultados**.

Desacoplar y acoplar las barras laterales

Las ventanas de las barras laterales pueden desacoplarse de la IGU o acoplarse en la IGU. Cuando se acopla una ventana flotante, se acopla en la última posición donde estuvo acoplada. También puede acoplar una ventana dentro de otra ventana en forma de pestaña.

Hay varios métodos para acoplar o desacoplar una ventana:

- Haga clic con el botón derecho en la barra de título de una ventana y seleccione el comando pertinente (**Flotante** o **Acoplada**).
- Haga doble clic en la barra de título de la ventana. Si la ventana está acoplada, se desacoplará. Si está desacoplada, la ventana se acoplará en la última posición donde estuvo acoplada.
- Arrastre la ventana (usando su barra de título como controlador) y desacóplela de su posición. Arrastre una ventana flotante (por su barra de título) hasta la posición donde desee acoplarla. Aparecerán dos grupos de flechas azules. El grupo de flechas exterior permite acoplar la ventana en relación a la ventana de la aplicación (en relación al borde superior, inferior, izquierdo o derecho de la IGU). El grupo de flechas interior permite acoplar la ventana en relación a la ventana sobre la que está el puntero del ratón. Si coloca la ventana sobre el botón central del conjunto de flechas interior (o sobre la barra de título de una ventana), la ventana se acopla dentro de la ventana en forma de pestaña.

Para desacoplar una pestaña de una ventana, haga clic con el botón derecho sobre la pestaña. Para arrastrar una pestaña fuera de su ventana, simplemente haga clic sobre la pestaña y arrástrela.

Ocultar automáticamente las barras laterales

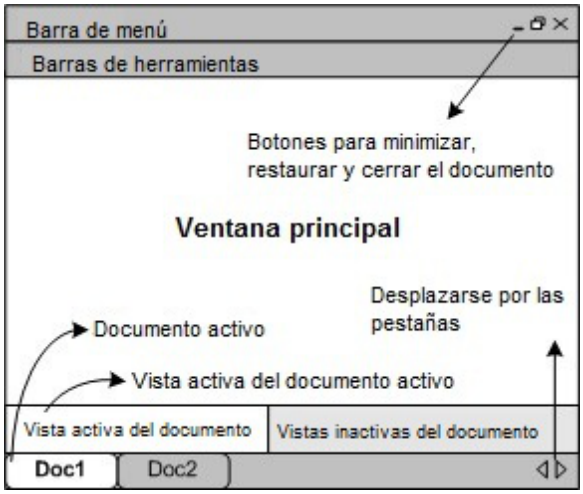
Las barras laterales acopladas se pueden minimizar en forma de botones que se sitúan en los bordes de la ventana de la aplicación. De esta manera tendrá más espacio para trabajar en la ventana principal y en las demás barras laterales. Al pasar el cursor sobre la barra minimizada, ésta se abrirá.

Para ocultar automáticamente las barras laterales o restaurarlas, haga clic en el icono en forma de chincheta que aparece en la barra de título de la barra lateral (o haga clic con el botón derecho y seleccione **Ocultar automáticamente**).

3.1.1 Ventana principal

La ventana principal (*imagen siguiente*) es la ventana donde se visualizan y editan los documentos.

© 2016-2022 Altova GmbH



Archivos en la ventana principal

- Se pueden abrir y editar a la vez cualquier cantidad de archivos
- Cada documento que esté abierto tiene su propia ventana y pestaña (que tiene el nombre de archivo del documento) en la parte inferior de la ventana principal. Para activar un documento, haga clic en su pestaña.
- Si hay varios archivos abiertos, puede que algunas pestañas no sean visibles por falta de espacio en la barra de pestañas. Para ver todas las pestañas: (i) utilice los botones de desplazamiento situados a la derecha de la barra de pestañas o (ii) seleccione el documento pertinente en el menú **Ventanas**[1629].
- Cuando se maximiza el documento activo, los botones **Minimizar**, **Restaurar** y **Cerrar** están situados en el lado derecho de la barra de menú. Cuando un documento está en cascada, en mosaico o minimizado, los botones **Maximizar**, **Restaurar** y **Cerrar** están situados en la barra de título de la ventana del documento.
- Cuando se maximiza un archivo, se maximizan también todos los archivos que estén abiertos.
- Puede organizar los archivos abiertos en cascada o en mosaico con los comandos del menú **Ventanas**[1629].
- También puede activar los archivos abiertos en la misma secuencia en la que se abrieron si pulsa las teclas **Ctrl+Tabulador** o **Ctrl+F6**.
- Al hacer clic con el botón derecho en la pestaña de un documento aparece un menú contextual con varios comandos del menú **Archivo**, como **Imprimir** y **Cerrar**.

Vistas en la ventana principal

El documento activo se puede visualizar y editar en varias vistas. Las vistas disponibles aparecen en una barra situada sobre las pestañas de los documentos (*ver imagen anterior*) y la vista activa aparece resaltada. Para seleccionar una vista, haga clic en el botón de la vista o seleccione la vista pertinente en el menú **Vista**[1484].

Las vistas disponibles pueden ser vistas de edición o vistas de explorador:

- **Vista Texto**[145]**:** una vista de edición con color de sintaxis para trabajar directamente con el código del documento.
- **Vista Cuadrícula**[169]**:** una vista de edición estructurada. El documento se representa en forma de cuadrícula estructurada que se puede manipular de forma gráfica. Esta vista contiene también una vista Tabla incrustada, que muestra los elementos que se repiten en formato de tabla.
- **Vista Esquema**[141]**:** para ver y editar esquemas XML.

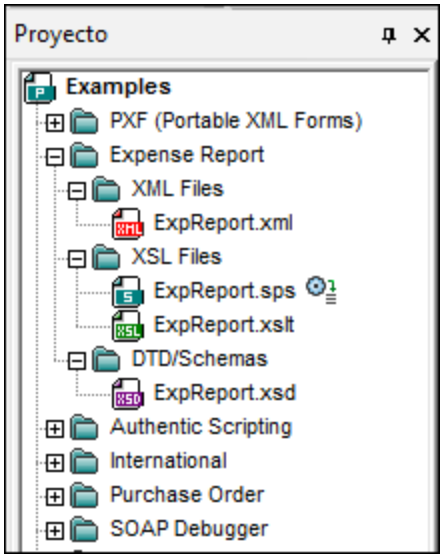
- **Vista WSDL**[141]: para ver y editar documentos WSDL.
- **Vista Authentic**[613]: para editar documentos XML basados en hojas de estilos StyleVision Power Stylesheet en una interfaz gráfica.
- **Vista Explorador**[279]: una vista de explorador integrada compatible con CSS y hojas de estilos XSL.

Nota: la vista predeterminada para cada tipo de archivo se puede personalizar en el cuadro de diálogo **Herramientas | Opciones**[1583] (en el panel *Vista predeterminada* de la sección *Tipos de archivo*).

3.1.2 Ventana Proyecto

Un proyecto es un conjunto de archivos relacionados entre sí. Por ejemplo, en la imagen siguiente, el proyecto llamado `Examples` agrupa varios archivos de ejemplo en diferentes carpetas, cada una de ellas con sus propias subcarpetas. Por ejemplo, en el proyecto `Examples` la carpeta de ejemplo `Expense report` tiene subcarpetas para archivos XML, archivos XSL y archivos de esquema.

Nota: el proyecto de ejemplos *Examples* se abre por defecto en la ventna Proyecto. Si lo ha cerrado y necesita abrirlo vaya a la carpeta **`Examples`** de la aplicación, en la *carpeta (Mis) Documentos*[36], y haga dobre clic en el archivo **`Examples.spp`**.



Por tanto, los proyectos son una manera de reunir archivos que se usan juntos y también de tener acceso a ellos más rápido. Además, puede definir archivos de esquema y XSLT para cada carpeta para procesar los archivos de cada carpeta por lotes.

Operaciones de proyecto

Los comandos para las operaciones de carpeta están en el menú **Proyecto** y algunos están también disponibles en los menús contextuales del proyecto y sus carpetas (al hacer clic con el botón derecho). Hay un subconjunto de comandos de menú que también están disponibles en la barra de herramientas de la ventana Proyecto (*imagen siguiente*) porque se usan muy a menudo.

© 2016-2022 Altova GmbH



Los comandos de la barra de herramientas son, de izquierda a derecha: *Proyecto nuevo*, *Abrir proyecto*, *Volver a cargar el proyecto*, *Guardar el proyecto*, *Agregar archivo activo al proyecto*, *Seleccionar archivo activo*, *Expandir todo*, *Contraer todo*, *Buscar*. Los nombres de estos comandos son autoexplicativos pero también se describen en el menú Proyecto[1298].

A continuación enumeramos las operaciones clave relacionadas con la ventana Proyecto.

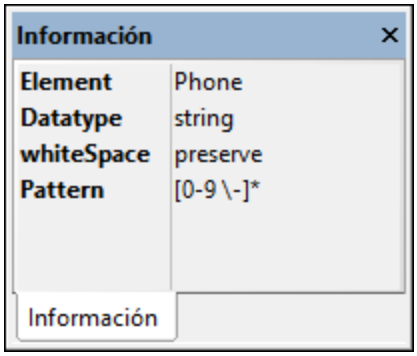
- En la ventana Proyecto se puede abrir sólo un proyecto. Cuando se crea un proyecto nuevo o se abre un proyecto ya existente, el proyecto de la ventana Proyecto se sustituye con el proyecto nuevo.
- Después de realizar cambios en un proyecto, es necesario guardar los cambios (clic en el comando **Proyecto | Guardar proyecto**). Si en un proyecto hay cambios sin guardar, junto al nombre del proyecto aparece un asterisco.
- El proyecto se representa en forma de árbol, con carpetas, archivos y otros recursos. Los recursos se pueden agregar en cualquier nivel y en cualquier subcarpeta.
- Las carpetas de proyecto son carpetas *semánticas* que representan una agrupación lógica de archivos. **No se corresponden** con la organización jerárquica de los archivos en su disco duro.
- Las carpetas pueden coincidir y estar relacionadas con directorios de su sistema de archivo. A esas carpetas las llamamos *carpetas externas* y en la ventana Proyecto se representan con un icono de carpeta amarilla (en contraste con las carpetas normales, que son verdes). Las carpetas externas de proyecto deben sincronizarse expresamente con el comando **Actualizar**.
- Cada carpeta puede contener varios tipos de archivo diferentes. También puede definir extensiones de archivo para cada carpeta (en el cuadro de diálogo "Propiedades" de la carpeta) para guardar los archivos más comunes en varias carpetas. Cuando se añade un archivo a la carpeta primaria, el archivo se añade automáticamente a la subcarpeta a la que se asignaron archivos con ese tipo de extensión.
- Al pasar el cursor del ratón sobre un archivo de imagen de una de las carpetas del proyecto aparece una vista previa de esa imagen (formatos `.png`, `.jpeg`, `.gif`, `.bmp`, `.tiff` y `.ico`). Haga doble clic en la imagen para abrirla en el visor/editor de imágenes predeterminado del sistema.
- En la ventana Proyecto puede arrastrar carpetas y colocarlas en otra carpeta o en otra posición de la misma carpeta. Los archivos, sin embargo, sólo se pueden arrastrar a otra carpeta, pero no se pueden mover a otra posición de la misma carpeta (en la que los archivos están organizados alfabéticamente). Además, puede arrastrar archivos y carpetas desde el explorador de archivos de Windows hasta la ventana Proyecto.
- Cada carpeta tiene un conjunto de propiedades que se definen en el cuadro de diálogo "Propiedades" de cada carpeta. Estas propiedades incluyen extensiones de archivos asignadas a la carpeta, el esquema con el que validar los archivos XML, el archivo XSLT con el que transformar los archivos XML, etc.
- Para procesar por lotes los archivos de una carpeta, haga clic con el botón derecho en la carpeta y seleccione el comando pertinente en el menú contextual (p. ej. **Validar el documento XML** o **Comprobar formato XML**).

Para más información sobre los proyectos, consulte la sección Proyectos[1060].

Nota: **p**ara habilitar o deshabilitar la ventana Proyecto, utilice el menú **Ventanas**.

3.1.3 Ventana Información

La ventana Información (*imagen siguiente*) muestra información sobre el elemento o el atributo donde está situado el cursor. La ventana Información está disponible en las vistas Texto, Cuadrícula, XBRL y Authentic.



Esta ventana se puede ocultar/mostrar desde el menú **Ventanas**.

También es conveniente tener en cuenta estos aspectos de la ventana Información:

- Cuando el documento activo es un documento XSLT, la ventana incluye una pestaña más llamada *XSLT* que ofrece información específica sobre el código XSLT. Para más información sobre la pestaña *XSLT* de la ventana Información consulte el apartado XSLT y XQuery | XSLT | Esquema XSL [496].
- Cuando un documento JSON (instancia o esquema) está activo, las opciones de validación y edición inteligente están en la pestaña *JSON* de la ventana Información. Para más información consulte Validar documentos JSON [754].
- Cuando el documento activo es un documento de esquema XML (archivo `.xsd`), la ventana Información incluye una pestaña más llamada Esquema con opciones para habilitar la validación extendida de esquemas. Para más información sobre la pestaña *Esquema* de la ventana Información consulte el apartado Reglas de esquema [449].
- Cuando el documento activo es un documento XULE, la pestaña *XULE* de la ventana Información permite seleccionar una taxonomía XBRL como referencia sobre la estructura de una instancia XBRL. La información de la taxonomía se usa para las funciones de finalización automática del editor XULE [922].

3.1.4 Ayudantes de entrada

Los ayudantes de entrada son una función de edición inteligente que ayuda a crear documentos XML válidos con rapidez. Mientras edita un documento, los ayudantes de entrada muestran opciones de edición estructural en base a la posición del cursor. Los ayudantes de entrada obtienen la información necesaria del esquema XML, del archivo DTD o de la hoja de estilos StyleVision Power Stylesheet subyacente. Por ejemplo, si está editando un documento XML, los elementos, atributos y entidades que se pueden insertar en la posición actual del cursor aparecerán en las ventanas de los ayudantes de entrada.

Los ayudantes de entrada disponibles en cada momento dependen de:

1. *La clase de documento que se esté editando*. Por ejemplo, los documentos XML tienen ayudantes de entrada para elementos, atributos y entidades. Sin embargo, los documentos XQuery tienen

ayudantes de entrada para palabras clave XQuery, variables y funciones. Los ayudantes de entrada disponibles para cada tipo de documento se detallan en esta documentación, en la sección correspondiente a cada documento.

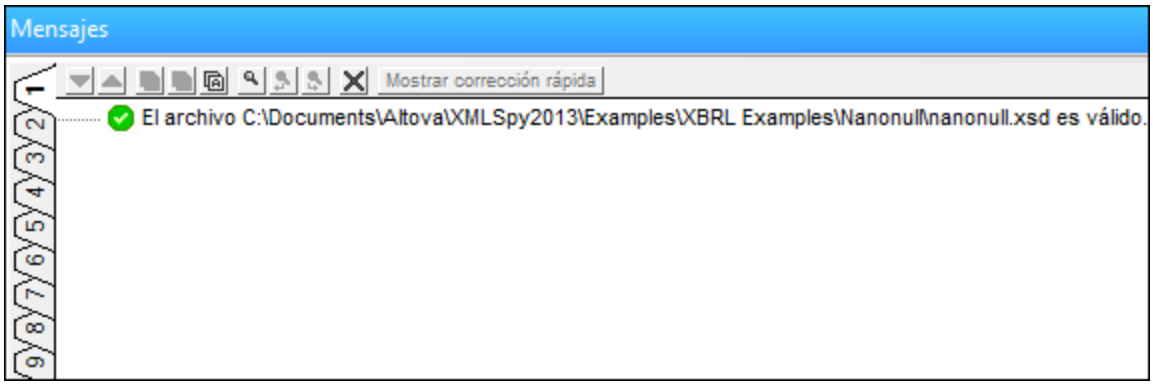
2. *La vista actual*. Puesto que los mecanismos de edición son diferentes en cada vista, los ayudantes de entrada están diseñados para ser compatibles con el mecanismo de edición de la vista activa. Por ejemplo, en la vista Texto sólo se puede insertar un elemento en la posición del cursor, así que el ayudante de entrada está diseñado para insertar un elemento cuando se haga doble clic sobre el elemento. Sin embargo, en la vista Cuadrícula se puede insertar un elemento antes del nodo seleccionado, después o como nodo secundario. En este caso el ayudante de entrada tiene tres pestañas para insertar, anexar o agregar el elemento como secundario. Cada pestaña contiene los elementos disponibles para cada una de esas operaciones.

La sección Vistas de edición[141] ofrece una descripción general de los ayudantes de entrada disponibles en cada tipo de vista. También encontrará más detalles sobre las diferencias según el tipo de documento en la descripción de los distintos tipos de documento. Consulte por ejemplo las secciones sobre ayudantes de entrada XML[333] y ayudantes de entrada XQuery[509].

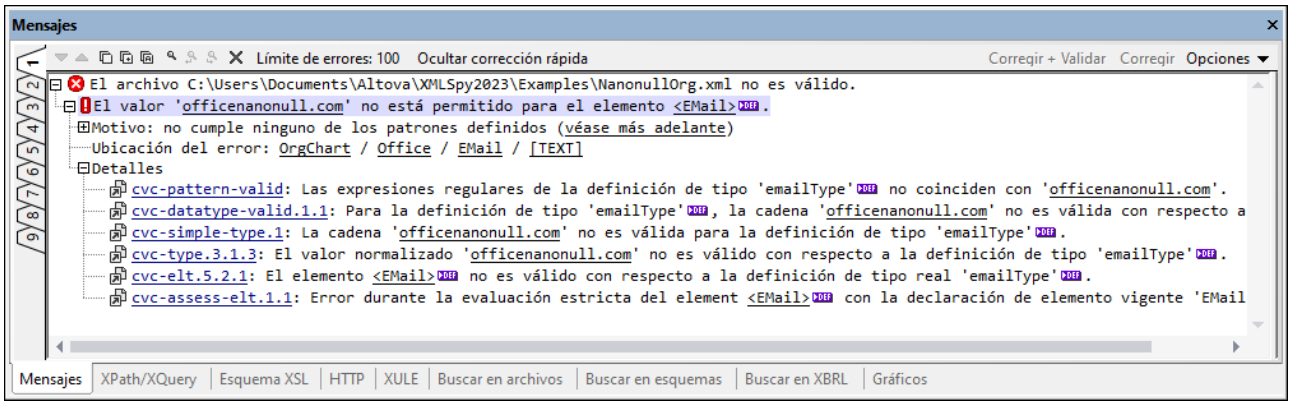
Tenga en cuenta que puede habilitar o deshabilitar los ayudantes de entrada con la opción de menú **Ventanas | Ayudantes de entrada**. Además en Visual Studio .NET los ayudantes de entrada tienen un prefijo que corresponde al nombre de la aplicación.

3.1.5 Ventana de resultados: Mensajes

La ventana Mensajes muestra mensajes (como errores y otros resultados) sobre acciones llevadas a cabo en XMLSpy. Por ejemplo, si se valida un documento XML, esquema XML, DTD o XQuery y el documento es válido, aparece un mensaje de validación correcta (*ver imagen siguiente*) en la ventana Mensajes:



De lo contrario, aparece un mensaje que describe el error (*ver imagen siguiente*). Observe que el mensaje de la imagen anterior incluye hipervínculos a los nodos del documento XML (texto subrayado en negro), así como hipervínculos a las secciones de la especificación pertinente en Internet que describe la regla en cuestión (texto subrayado en azul). Al hacer clic en los botones `Def` color púrpura, la definición de esquema correspondiente se abre en la vista Esquema.

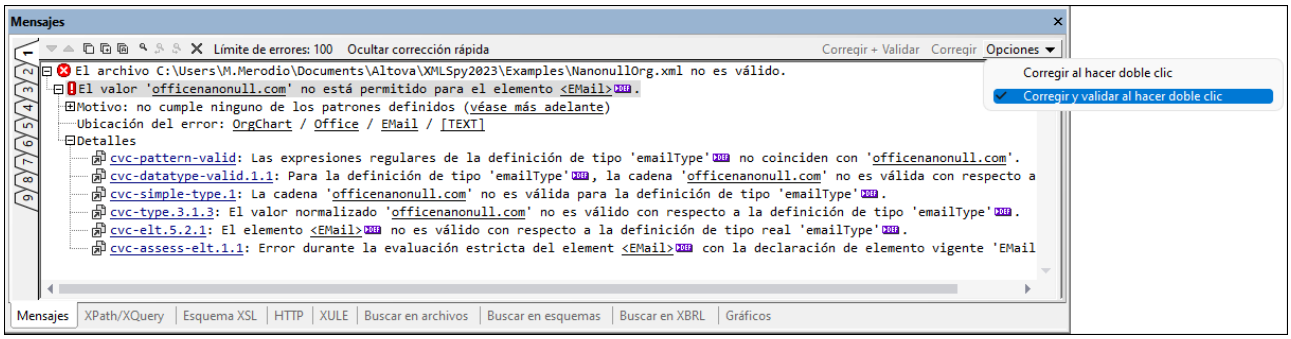


La ventana Mensajes está habilitada en todas las vistas, pero si hace clic en un hipervínculo al contenido del documento XML, dicho nodo se resaltará en el documento XML en la vista Texto. Sin embargo, si se valida un esquema XML en la vista Esquema, al hacer clic en el botón `Def` color púrpura, la vista no cambiará.

Opciones de corrección rápida de la validación XML

Si el documento XML se valida en la **vista Texto** o **Cuadrícula**, el mensaje de validación incluye opciones de corrección rápida basadas en la información disponible en el esquema. Para ver una lista de opciones de corrección rápida, haga clic en el botón **Mostrar corrección rápida** (*ver imagen anterior*). Como resultado aparece un panel con las opciones de corrección rápida en la misma ventana de mensajes (*ver imagen siguiente*).

Recuerde que los errores se van mostrando en la ventana de mensajes de uno en uno. Además, si existen errores de formato XML (como cuando falta una etiqueta inicial o final de un elemento), estos se muestran antes de los errores de validación. Por tanto, el botón **Mostrar corrección rápida** se habilitará cuando se llegue al primer error de validación (una vez corregidos los errores de formato XML).



En el panel de corrección rápida (*ver panel situado a la derecha en la imagen anterior*), seleccione una de las opciones de corrección rápida y después haga clic en el botón **Corregir + Validar** o en el botón **Corregir** (*ver imagen anterior*). El texto no válido del documento XML se sustituye con la opción de corrección rápida seleccionada. Otra opción es hacer doble clic en la opción de corrección rápida que desee utilizar. Al hacer doble clic en la opción, se corrige o se corrige y valida el documento (dependiendo de la opción seleccionada en la lista desplegable *Opciones,* tal y como muestra la imagen anterior). El comando **Corregir y validar** es práctico para comprobar si hay más errores de validación.

Para ocultar el panel de corrección rápida, haga clic en el botón **Ocultar corrección rápida** (*ver imagen anterior*). Para más información consulte el apartado Vistas de edición | Vista Esquema | Validación y corrección rápida 238.

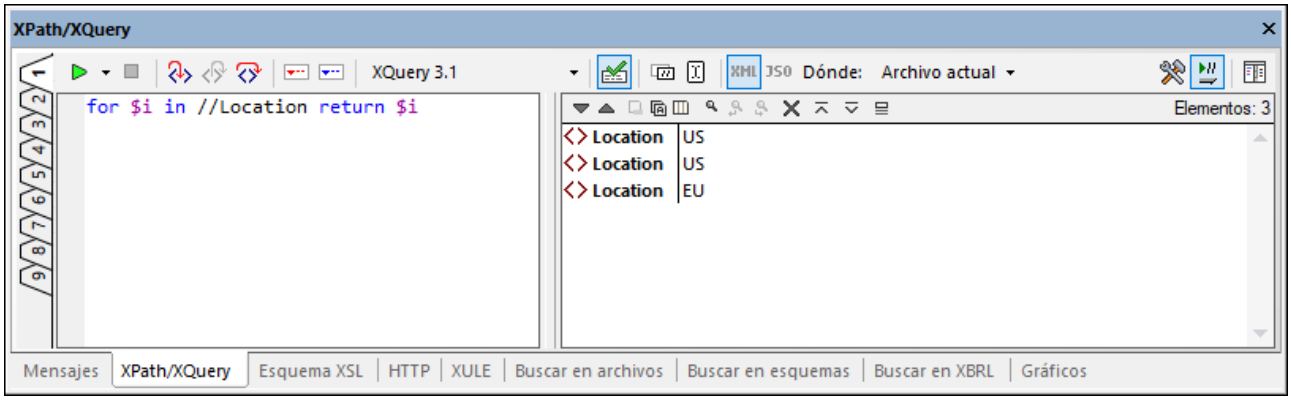
Validar carpetas y archivos en la ventana de proyecto

El comando **Validar** (del menú **XML**) se suele aplicar al documento activo. Sin embargo, también puede aplicar este comando a un archivo, una carpeta o un grupo de archivos del proyecto activo. Seleccione el archivo o la carpeta que desea validar en la ventana Proyecto y después haga clic en **XML | Validar el documento XML**[1337] o pulse la tecla **F8**. Los archivos del proyecto que no sean válidos se abren en la ventana principal y aparece un mensaje de error de tipo *El archivo no es válido*.

Nota: también puede realizar comprobaciones de formato XML (haciendo clic en el comando **Comprobar formato XML**[1335] o pulsando **F7**) en la ventana Proyecto.

3.1.6 Ventana de resultados: XPath/XQuery

La ventana XPath/XQuery (*imagen siguiente*) permite construir, evaluar y depurar expresiones XPath y XQuery con respecto a documentos XML o JSON. (La compatibilidad con las consultas JSON no existe hasta la versión XPath/XQuery 3.1. *Consulte Transformaciones JSON con XSLT/XQuery*[758].)



En este apartado encontrará un resumen de las características principales de la ventana XPath/XQuery. Para una descripción más detallada de cómo trabajar con la ventana XPath/XQuery consulte la sección expresiones XPath/XQuery[573].

Características principales

Las características principales de la ventana XPath/XQuery son:

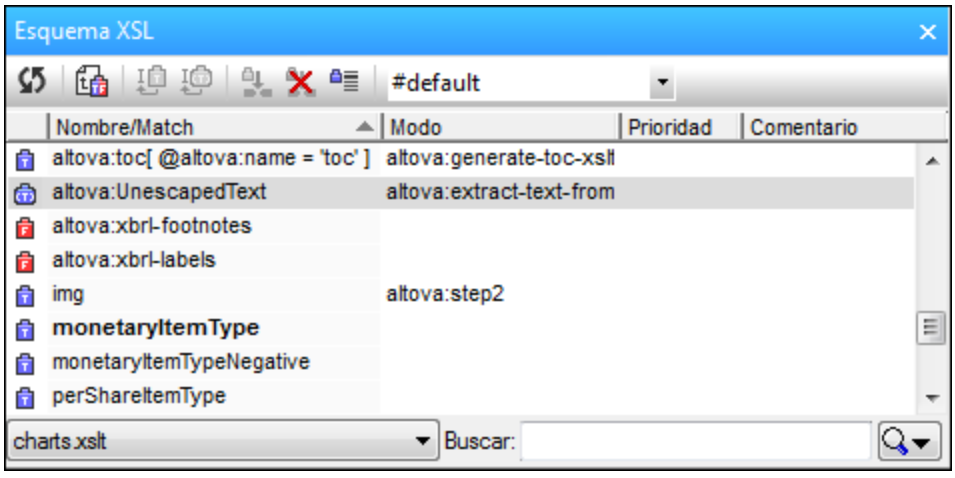
- *Modo Evaluador:* en este modo se evalúa una expresión XPath o XQuery con respecto a uno o más documentos XML/JSON. La expresión se introduce en el panel *Expresión* y el resultado se muestra en el panel adyacente *Resultados*. Puede hacer clic en los nodos del resultado para ir a ese mismo nodo en el documento XML/JSON. Para más detalles consulte Expresiones XPath/XQuery[573].
- *Modo Depurador:* en este modo puede depurar expresiones XPath/XQuery conforme estas se aplican al documento XML activo. Puede definir puntos de interrupción y puntos de seguimiento y avanzar paso a paso por la evaluación. En cada paso puede ver el contenido de las variables, así como definir expresiones de inspección personalizadas para comprobar distintos aspectos de la evaluación. Para más detalles consulte Modo Depurador[582].

- *Generador de expresiones:* cuenta con ayudantes de entrada y diálogos emergentes informativos que le ayudan a construir expresiones con sintaxis correcta. Para más detalles consulte Generador de expresiones 590.
- *Compatibilidad con varios lenguajes:* puede escoger qué versión del lenguaje quiere usar; puede elegir entre XPath 1.0, XPath 3.1 y XQuery 3.1. La expresión que introduzca se evalúa según las normas del lenguaje seleccionado.
- *Abrir y guardar expresiones desde/en archivos:* puede guardar expresiones XPath/XQuery, junto con la configuración actual de la ventana, en un archivo XQuery, y puede cargar expresiones desde un archivo XQuery.
- *Detección automática del tipo de archivo:* el tipo de archivo del documento (XML o JSON) actual se detecta automáticamente y se selecciona automáticamente el documento de destino apropiado. Si el alcance es un conjunto de documentos puede elegir manualmente el tipo de documento de destino.
- *Alcance flexible para documentos de destino:* puede probar la expresión en el documento XML o JSON activo o en un conjunto de documentos elegido por usted.
- *Paneles Resultados y Expresión:* puede definir cómo aparecen los elementos de los paneles Resultados y Expresión 582.
- *Resultados vinculados directamente a documentos y nodos:* las líneas del panel Resultados contienen enlaces a los documentos o nodos de documento relevantes en cada caso. Eso permite comprobar los datos directamente en nodos específicos.
- Análisis de depuración: en el modo Depurador se muestra una gran cantidad de información analítica. Además, puede definir expresiones personalizadas para comprobar otros aspectos de la evaluación.

Para una descripción detallada de las características de la ventana XPath/XQuery consulte la sección expresiones XPath/XQuery 573.

3.1.7 Ventana de resultados: Esquema XSL

La ventana Esquema XSL (*ver imagen siguiente*) muestra todas las plantillas y funciones de una hoja de estilos XSLT y, si lo desea, también las de las hojas de estilos XSLT importadas e incluidas. La ventana Esquema XSL está acoplada por defecto en las ventanas de resultados, en la parte inferior de la ventana de XMLSpy. La ventana Esquema XSLT se puede desacoplar o acoplar en uno de los bordes de la ventana de XMLSpy.



La ventana Esquema XSL ofrece información sobre las plantillas y funciones de la hoja de estilos. Esta información se puede ordenar y en ella puede realizar búsquedas. La barra de herramientas de la ventana

 © 2016-2022 Altova GmbH

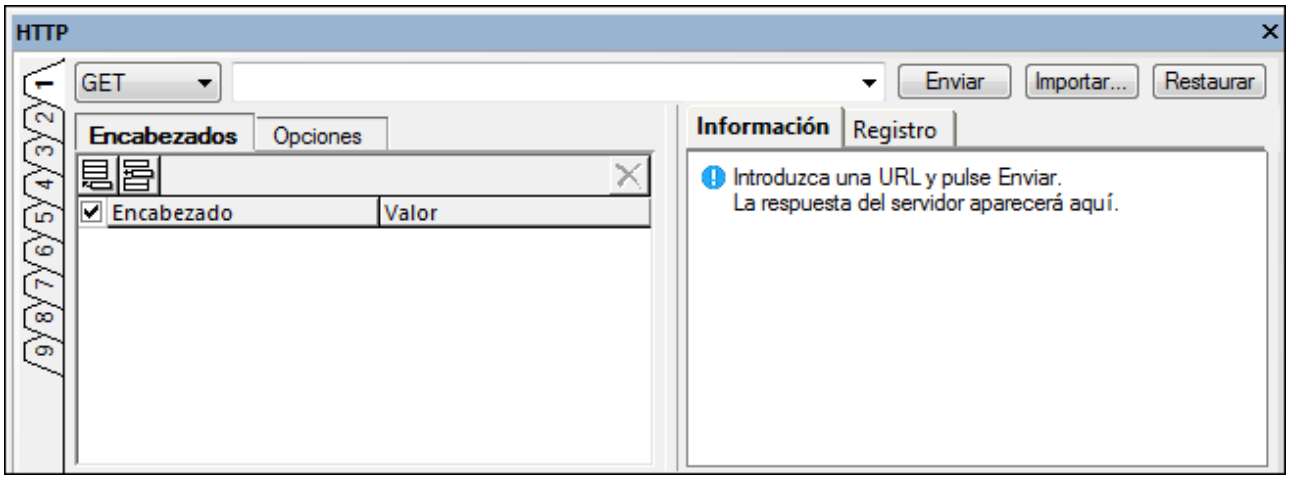
contiene comandos con los que podrá insertar llamadas a plantillas con nombres y configurar plantillas con nombre como punto inicial de la transformación. Para más información sobre cómo trabajar con la ventana Esquema XSL, consulte la sección XSLT y XQuery | XSLT | Esquema XSL | Ventana Esquema XSL 497.

Nota: la pestaña *XSLT* de la ventana Información contiene información sobre el archivo de la hoja de estilos y comandos relacionados con el archivo. Para más información sobre cómo usar estos comandos, consulte la sección XSLT y XQuery | XSLT | Esquema XSL | Ventana Información 500.

3.1.8 Ventana de resultados: HTTP

En la ventana de resultados HTTP (imagen siguiente) puede probar comandos HTTP: creando y enviando solicitudes HTTP a un servidor web y recibiendo y revisando las respuestas.

La ventana HTTP cuenta con nueve pestañas (*imagen siguiente*). En cada pestaña puede almacenar una solicitud y después pasar de una pestaña a otra. Tras crear una solicitud en la ventana, podrá enviarla con solo hacer clic en el botón **Enviar**. La respuesta aparece directamente en la ventana.



La ventana se divide en varios paneles:

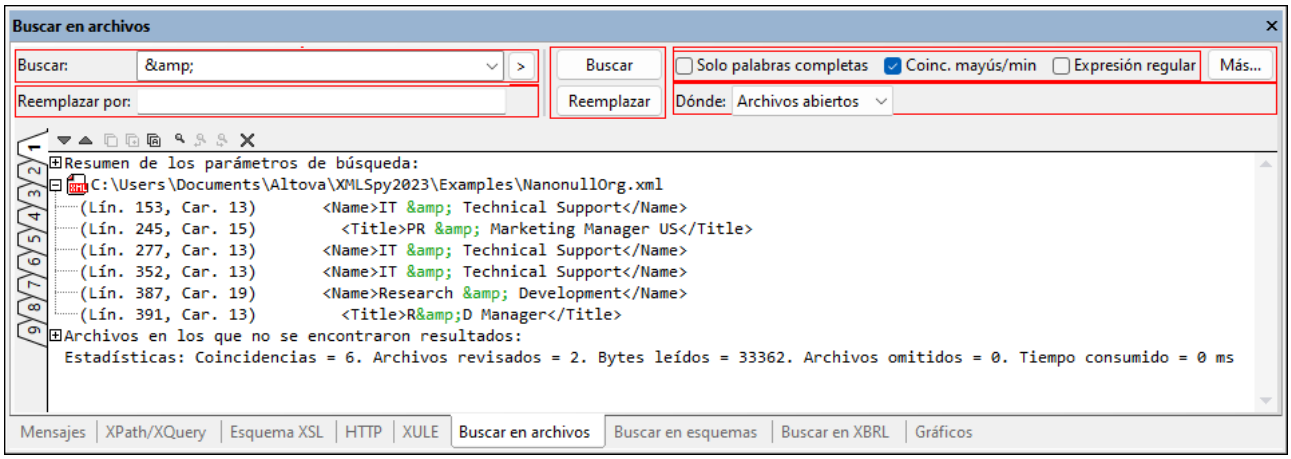
- En el panel superior hay: (i) un cuadro combinado donde puede seleccionar el método HTTP que desea utilizar, (ii) un campo de entada para la URL del servidor web y (iii) botones relacionados con la ejecución de solicitudes HTTP (**Enviar**, **Importar** y **Restaurar**).
- En el panel izquierdo puede crear la solicitud 806.
- En el panel derecho aparece la respuesta 814.

Para aprender a usar la ventana de resultados HTTP consulte la sección HTTP 805.

3.1.9 Ventana de resultados: Buscar en archivos

La ventana Buscar en archivos (*ver imagen siguiente*) permite realizar operaciones de búsqueda y reemplazo dentro de varios documentos de una sola vez. Además, ofrece varios mecanismos que le ayudarán a navegar rápido por los resultados de la búsqueda. Los resultados de cada acción de búsqueda y reemplazo se

presentan en una de las pestañas, numeradas del 1 al 9. Al hacer clic en uno de los resultados, el elemento pertinente del documento aparecerá resaltado en la vista Texto.



Criterios de búsqueda

Hay dos criterios de búsqueda principales: (i) qué se busca y (ii) dónde se busca.

¿Qué se busca? La cadena de texto de búsqueda se introduce en el cuadro de texto "Buscar". Si la cadena debe coincidir con una palabra entera, marque la casilla "Sólo palabras completas". Por ejemplo, si marcamos la casilla "Sólo palabras completas" para la cadena de búsqueda `catar` se encontrará sólo la palabra `catar` y no la palabra `percatar`. Si las mayúsculas y minúsculas son relevantes para su búsqueda, marque la casilla "Coinc. mayús/min". Si el texto escrito en el cuadro de texto "Buscar" es una expresión regular, debe marcar la casilla "Expresión regular". La ventana contiene un ayudante de entrada para caracteres de expresiones regulares (pulse el botón >). Para más información sobre cómo usar expresiones regulares en las búsquedas, consulte la sección Buscar[1289]. El botón **Más** abre el cuadro de diálogo Buscar en archivos[1294], donde podrá configurar condiciones y acciones de búsqueda avanzadas. Para más información visite la sección Edición | Buscar en archivos[1294].

¿Dónde se busca? La búsqueda puede llevarse a cabo: (i) en todos los archivos que estén abiertos en la IGU, (ii) en los archivos del proyecto actual y (iii) en los archivos de la carpeta seleccionada. También puede configurar más condiciones de búsqueda en el cuadro de diálogo Buscar en archivos[1294] (haga clic en el botón **Más**).

Reemplazar por

En el cuadro de texto "Reemplazar por" introduzca la cadena de texto que debe reemplazar el resultado de la búsqueda. Tenga en cuenta que si deja vacío el cuadro de texto "Reemplazar por" y hace clic en el botón **Reemplazar** el resultado de la búsqueda se reemplazará por una cadena vacía.

Resultados de la búsqueda

Tras hacer clic en el botón **Buscar** o **Reemplazar**, los resultados de la búsqueda o de la operación de reemplazo aparecerán en la ventana de resultados Buscar en archivos. Los resultados se dividen en cuatro partes:

- Un resumen de los parámetros de búsqueda, que incluye la cadena de búsqueda y en qué archivos se buscó.

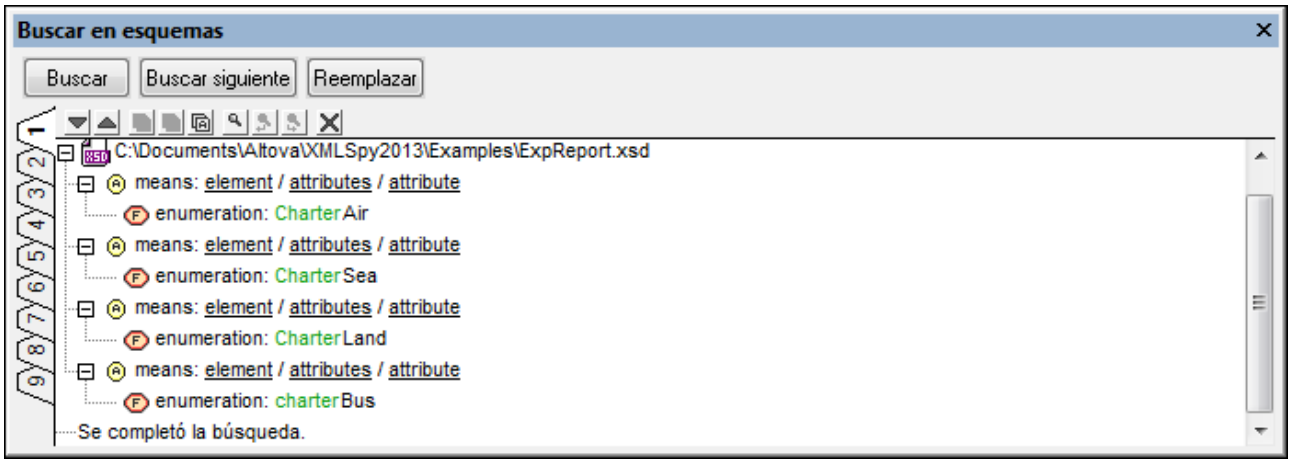
© 2016-2022 Altova GmbH

- Una lista de las cadenas de texto encontradas o reemplazadas (dependiendo de si pulsó el botón **Buscar** o **Reemplazar**). Los elementos de esta lista son vínculos al texto encontrado o reemplazado en la vista Texto del documento. Si el documento no está abierto, se abrirá en la vista Texto y el texto encontrado/reemplazado aparecerá resaltado.
- Una lista de los archivos que se buscaron y en los que no se encontraron resultados.
- Un resumen de las estadísticas de la búsqueda, incluido el número de coincidencias y el número de archivos revisados.

Nota: tenga en cuenta que la función de búsqueda en archivos ejecuta los comandos de búsqueda y reemplazo en varios archivos de forma simultánea y muestra los resultados en la ventana Buscar en archivos. Para realizar una búsqueda y pasar de un resultado al siguiente, utilice el comando Buscar[1289].

3.1.10 Ventana de resultados: Buscar en esquemas

Cuando un esquema XML está activo en la vista Esquema, podrá realizar búsquedas en el esquema gracias a la función inteligente de búsqueda y reemplazo de XMLSpy. Para acceder a la función de búsqueda y reemplazo de la vista Esquema: (i) utilice los comandos **Buscar** y **Reemplazar** del menú **Edición** o (ii) pulse los botones **Buscar** y **Reemplazar** de la ventana Buscar en esquemas (*ver imagen siguiente*).



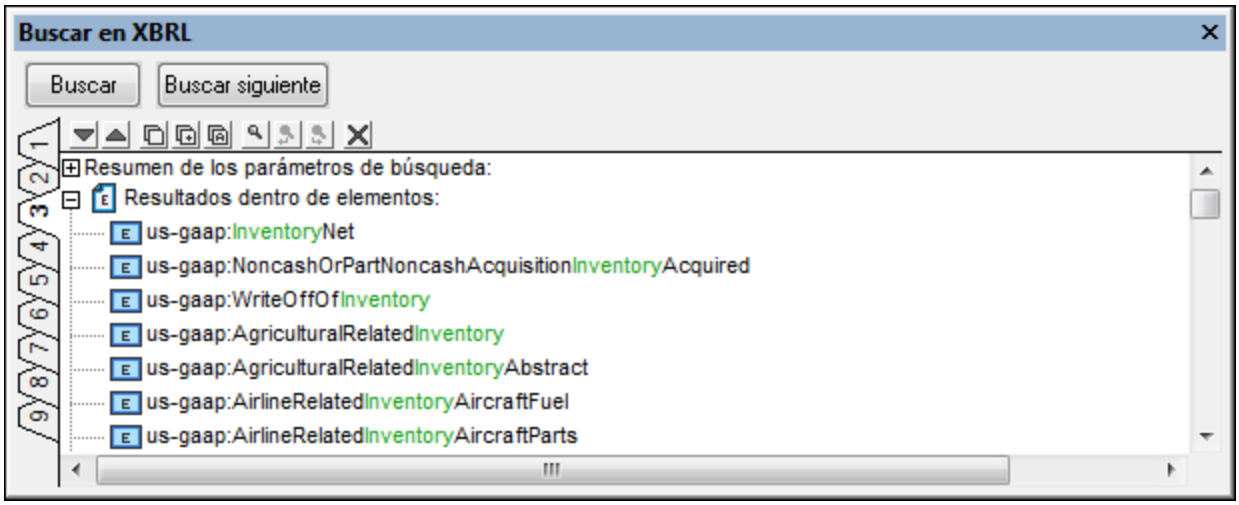
Los resultados de la función de búsqueda y reemplazo de la vista Esquema (es decir, cada vez que se ejecuta el comando **Buscar** o **Reemplazar**) se muestran en la ventana Buscar en esquemas. El término de la búsqueda aparecerá en verde en los resultados. Por ejemplo, en la imagen anterior, se buscó el término `charter`, sin tener en cuenta las mayúsculas y minúsculas. Observe que el resultado también indica la ubicación del archivo de esquema.

Los resultados se muestran en nueve pestañas (numeradas del 1 al 9). De este modo, puede guardar los resultados de una búsqueda en una pestaña, realizar una búsqueda nueva en otra pestaña y comparar los resultados. Para ver los resultados de una búsqueda nueva en otra pestaña, seleccione la pestaña antes de iniciar la búsqueda. Al hacer clic en un resultado de la ventana Buscar en esquemas, la ventana principal mostrará y resaltará el componente correspondiente en la vista Esquema. Así podrá buscar y navegar rápidamente hasta el componente pertinente. También puede copiar los mensajes en el portapapeles. Para más información, consulte la sección Resultados e información[486] de la descripción de la función de Búsqueda en esquemas[475].

3.1.11 Ventana de resultados: Buscar en XBRL

La ventana Buscar en XBRL (*ver imagen siguiente*) muestra los resultados de búsquedas realizados en un documento de taxonomía XBRL. Esta ventana incluye nueve pestañas. Así podrá comparar los resultados guardados en una pestaña con los resultados de otra pestaña.

La función de búsqueda en XBRL se puede ejecutar cuando un documento de taxonomía XBRL está abierto en la vista XBRL. Para más información sobre cómo realizar la búsqueda, consulte el apartado Búsqueda en XBRL 930 de la sección XBRL 817 del *Manual del usuario*.



Al hacer clic en el botón **Buscar** aparecerá el cuadro de diálogo "Buscar". Si hace clic en el botón **Buscar siguiente**, se buscará la siguiente repetición del término de búsqueda empezando desde la celda inmediatamente posterior a la celda de la vista XBRL donde está situado el cursor.

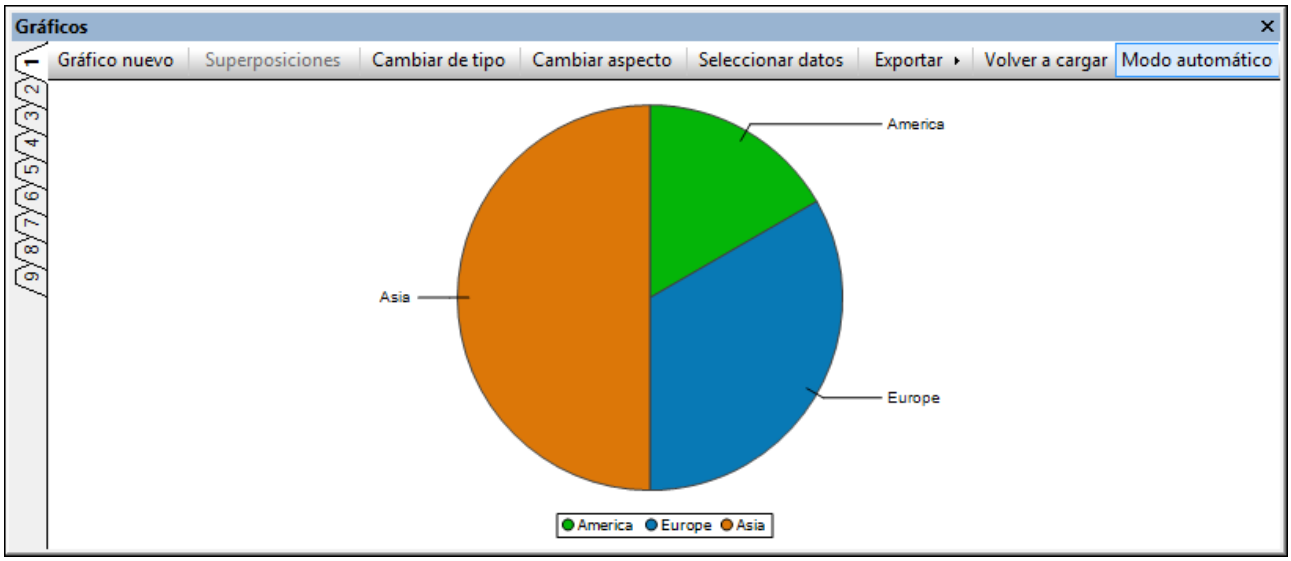
La barra de herramientas de la ventana Buscar en XBRL incluye estos comandos:

- Los iconos ***Siguiente*** y ***Anterior*** seleccionan el resultado siguiente y anterior respectivamente.
- Los comandos ***Copiar mensajes*** copian respectivamente el mensaje seleccionado, el mensaje seleccionado y sus secundarios o todos los mensajes en el portapapeles.
- Los comandos ***Buscar*** buscan cadenas de texto dentro de la ventana Buscar en XBRL.
- El comando ***Borrar*** elimina todos los mensajes de la pestaña activa.

3.1.12 Ventana de resultados: Gráficos

Cuando un documento XML está abierto en la vista Texto o Cuadrícula, se puede generar un gráfico (circular, de barras, etc.) que represente los datos del documento XML en la ventana Gráficos (*ver imagen siguiente*).

© 2016-2022 Altova GmbH



Crear el gráfico

Estos son los pasos básicos para crear un gráfico:

1. Coloque el cursor en el documento XML para seleccionar un nodo de contexto.
2. Haga clic en el botón **Gráfico nuevo** de la ventana Gráficos (*ver imagen anterior*) o haga clic con el botón derecho en la ventana principal y seleccione la opción **Gráfico nuevo** del menú contextual.
3. Aparece el cuadro de diálogo "Seleccionar columnas". Seleccione los datos para la tabla de datos del gráfico y haga clic en **Aceptar**. El gráfico se creará en la ventana Gráficos (*ver imagen anterior*).

Para más información consulte el apartado Gráficos[346] de la sección sobre XML[285] del *Manual del usuario*.

Modificar y gestionar gráficos

Puede crear un gráfico en cada una de las nueve pestañas de la ventana Gráficos (las pestañas están numeradas en el lado izquierdo de la ventana). Así podrá crear gráficos en varias pestañas y compararlos. Cuando cree un gráfico en una pestaña, éste se sobrescribe sólo si crea un gráfico nuevo en la misma pestaña. De lo contrario, los gráficos no se pueden borrar. Incluso si se cierra el documento XML usado para generar el gráfico, éste permanecerá en la pestaña en la que se creó.

La barra de herramientas de la ventana incluye varios botones:

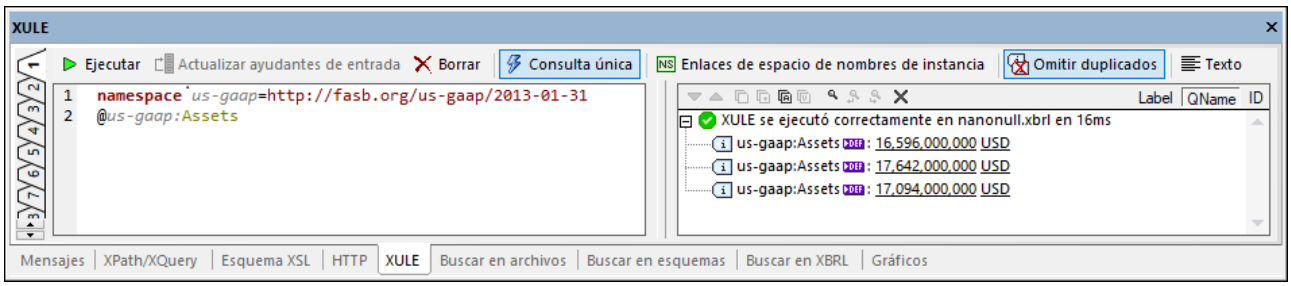
- *Gráfico nuevo:* abre el cuadro de diálogo "Seleccionar columnas" donde se configura la tabla de datos del gráfico.
- *Multicapas:* permite agregar y eliminar capas en el gráfico principal. Tras crear una capa nueva podrá agregar un gráfico nuevo a la capa haciendo clic en la pestaña *Capa nueva* y especificando qué datos se deben usar en este gráfico multicapa.
- *Cambiar de tipo:* permite cambiar el tipo de gráfico. Por ejemplo, de gráfico de barras a gráfico circular.
- *Cambiar aspecto:* permite cambiar las opciones de configuración, como el tamaño de la fuente o la combinación de colores.
- *Seleccionar datos:* abre el cuadro de diálogo "Seleccionar datos", que contiene la tabla de datos del gráfico y la selección de datos final que se presentará en el gráfico. Los datos para las series, el eje X y el eje Y se pueden cambiar en este diálogo. Los ejes X e Y se pueden seleccionar de forma gráfica

en la tabla de datos del gráfico. Al hacer clic en **Aceptar** se genera el nuevo gráfico en la ventana Gráficos.
- *Exportar:* el gráfico se puede exportar como archivo de imagen o como fragmento de código XSLT o XQuery. El fragmento de código XSLT o XQuery se puede usar en un documento XSLT o XQuery que, tras procesarse con el motor XSLT 2.0 o XQuery de Altova, representará el gráfico correctamente.
- *Volver a cargar* y *Modo automático:* si el botón **Modo automático** está pulsado, todos los cambios que se realicen en el documento XML subyacente se aplicarán automáticamente al gráfico de la ventana Gráficos. De lo contrario, para actualizar el gráfico deberá pulsar el botón **Volver a cargar**.

Consulte las secciones Gráficos 346 y XML 285 del manual para obtener más información.

3.1.13 Ventana de resultados: XULE

La ventana XULE permite consultar de forma interactiva el documento XBRL de instancia activo y ver el resultado de la consulta. La ventana XULE tiene nueve pestañas, cada una de las cuales está dividida en dos paneles: (i) el *panel de expresión XULE*, en el que introduce la expresión (o regla) XULE que quiere ejecutar y (ii) el panel *Resultados*, que muestra los resultados de la consulta.



Para ejecutar una expresión XULE en el documento XBRL de instancia activo siga estos pasos:

1. Asegúrese de que el documento XBRL de instancia en el que quiere realizar la consulta es el documento activo en la ventana principal.
2. Introduzca la expresión XULE en el panel de expresión XULE (*panel izquierdo*).
3. Haga clic en **Ejecutar** en la barra de herramientas de la ventana para ejecutar la expresión.
4. Los resultados de la consulta se muestran en el panel *Resultados* (*panel derecho*).

Para información más detallada consulte la descripción de la ventana XULE en la sección XBRL 924.

3.1.14 Barra de menú, barras de herramientas y barra de estado

Barra de menú

La barra de menú (*ver ilustración* 119) contiene los menús de la aplicación. Tenga en cuenta estas normas generales:

- Si los comandos de un menú **no son pertinentes** en una vista o en una posición determinada del documento, entonces no están disponibles.

© 2016-2022 Altova GmbH

- Algunos comandos abren un submenú con una lista de opciones adicionales. Los comandos de menú que tienen submenús se identifican con una punta de flecha apuntando hacia la derecha, situada después del nombre del comando.
- Algunos comandos abren un cuadro de diálogo que solicita más información al usuario. Esta información es necesaria para llevar a cabo el comando seleccionado. Estos comandos se identifican con unos puntos suspensivos (...) situados después del nombre del comando.
- Para acceder a un comando de menú, haga clic en el nombre del menú y después en el comando. Si un comando incluye un submenú, el submenú se abre al pasar el puntero del ratón sobre el comando. A continuación haga clic en el comando correspondiente del submenú.
- Los menús se pueden abrir si se pulsa la combinación de teclas correspondiente. La combinación de teclas de cada menú es **Alt+*TECLA***, siendo ***TECLA*** la letra que aparece subrayada en el nombre del menú. Por ejemplo, para abrir el menú **Archivo** la combinación de teclas es **Alt+A**.
- Los comandos de menú (es decir, los comandos de cada menú) se pueden ejecutar si selecciona primero (i) el menú con su combinación de teclas (ver el punto anterior) y después (ii) la combinación de teclas del comando en sí (**Alt+*TECLA***, siendo ***TECLA*** la letra que aparece subrayada en el nombre del comando). Por ejemplo, para crear un archivo nuevo (**Archivo | Nuevo**), pulse **Alt+A** y después **Alt+N**.
- Algunos comandos de menú se pueden seleccionar directamente si presiona una tecla de **acceso rápido** o combinación de teclas (**Ctrl+*TECLA***). Puede identificar los comandos con accesos rápidos porque la tecla de acceso rápido o la combinación de teclas aparece después del nombre del comando. Por ejemplo, puede usar la combinación de teclas **Ctrl+N** para crear un archivo nuevo o la tecla de acceso rápido **F8** para validar un archivo XML. También puede crear sus propias teclas de acceso rápido[1569] en la pestaña Teclado del cuadro de diálogo "Personalizar" (**Herramientas | Personalizar**).

Barras de herramientas

Las barras de herramientas (*ver ilustración*[119]) contienen iconos que funcionan como accesos rápidos para seleccionar comandos de menú. El nombre del comando aparece al pasar el puntero del ratón sobre el icono. Para ejecutar el comando, haga clic en el icono.

Los botones de las barras de herramientas se organizan en grupos. En el cuadro de diálogo **Herramientas | Personalizar | Barras de herramientas**[1566] puede indicar qué barras de herramientas desea activar. Estas opciones de configuración afectan a la vista actual. Para crear una configuración para otra vista, cambie a esa vista y vuelva a seleccionar las opciones correspondientes en el cuadro de diálogo **Herramientas | Personalizar | Barras de herramientas**[1566]. En la IGU también puede arrastrar barras de herramientas por sus controladores (o barras de título) y colocarlas en otra posición de la pantalla. Al hacer doble clic en el controlador la barra de herramientas se desacopla. Si hace doble clic otra vez en la barra de título, la barra de herramientas se acopla en la posición donde estuvo acoplada por última vez.

Barra de estado

La barra de estado está situada en la parte inferior de la ventana de la aplicación (*ver ilustración*[119]) y muestra (i) información de estado sobre la carga de archivos e (ii) información sobre los comandos de menú y los accesos rápidos a comandos de las barras de herramientas al pasar sobre estos el puntero del ratón. Si usa la versión de 64 bits de XMLSpy, la barra de estado incluye el sufijo (x64) después del nombre de la aplicación. Para la versión de 32 bits no se incluye ningún sufijo.

3.2 El entorno de aplicación

En esta sección describimos varios aspectos de la aplicación que debe conocer antes de empezar a trabajar con ella. Esta sección le ayudará a familiarizarse con la aplicación para que pueda empezar a usarla con mayor confianza. Aquí encontrará información importante sobre opciones de configuración y personalización y sobre cómo cambiarlas.

Esta sección se divide en tres partes:

- *Configuración y personalización*[136]*:* describe cómo definir importantes opciones de configuración y personalización.
- *Tutoriales, proyectos y ejemplos:*[139] repasa la ubicación de los archivos incluidos con la aplicación que no pertenecen al programa.
- *Características y archivos de ayuda*[139]*:* ofrece enlaces al sitio web de Altova, donde encontrará información sobre las funciones del producto, recursos de ayuda en varios formatos e información sobre otros productos de Altova.

3.2.1 Configuración y personalización

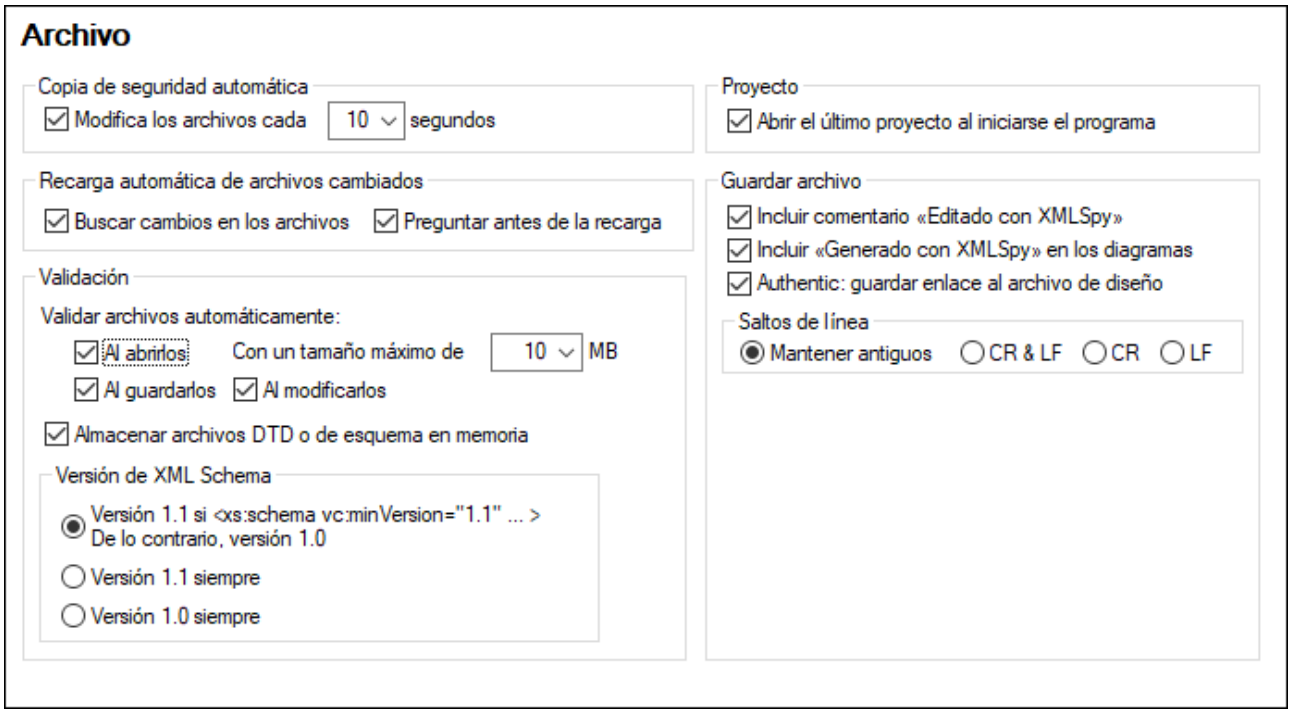
Esta sección explica las opciones de configuración y personalización que se pueden elegir en XMLSpy, así como algunos aspectos de los menús de XMLSpy. Esta sección se organiza en varias partes:

- Configuración[136]
- Personalización[137]
- Menús[138]

Configuración

El cuadro de diálogo "Opciones" incluye varias secciones donde se definen importantes opciones de configuración de XMLSpy (*ver imagen siguiente*, comando **Herramientas | Opciones**[1583]). Consulte las secciones de este cuadro de diálogo y las diferentes opciones para familiarizarse con ellas.

© 2016-2022 Altova GmbH

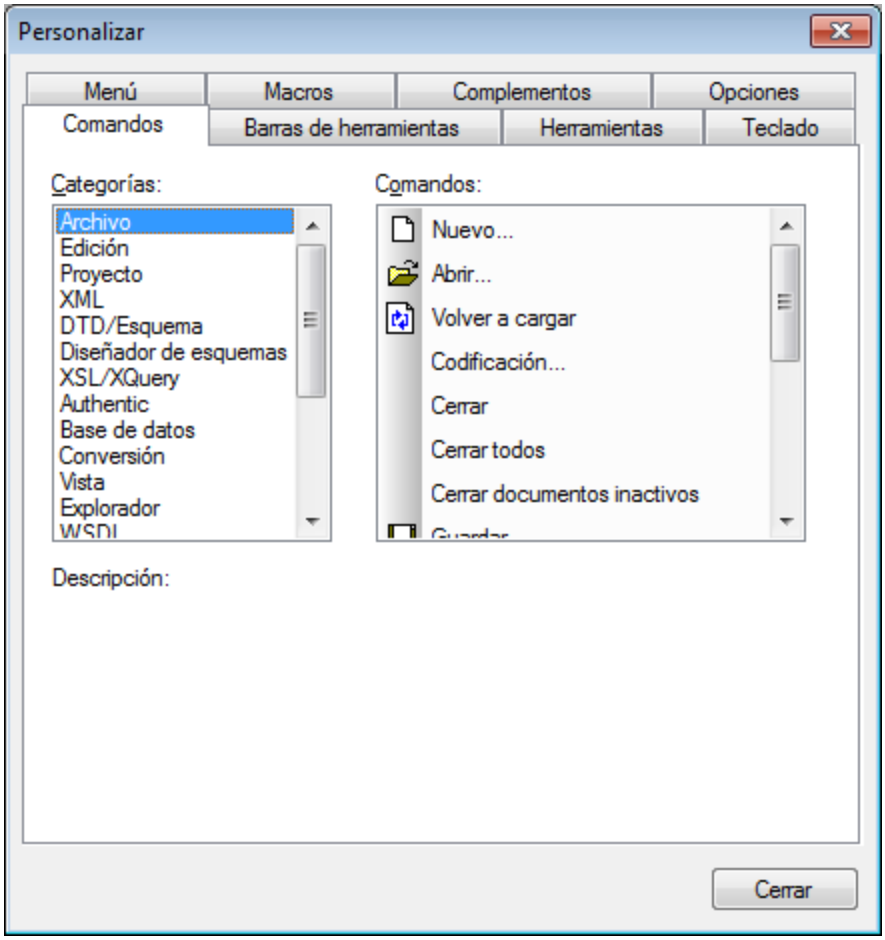


A continuación ofrecemos un resumen de las opciones de configuración más importantes. Para una descripción más detallada, consulte la sección Opciones (cuadro de diálogo)[1583] de la *Referencia del usuario*.

- *Tipos de archivos y vistas predeterminadas:* en la sección *Tipos de archivos* puede agregar tipos de archivos y XMLSpy los reconocerá. Los tipos de archivo se especifican con una extensión de archivo. Por cada tipo de archivo puede indicar a tipo de estándar debe ajustarse el archivo (p. ej. DTD, XQuery o JSON). Esto activará los ayudantes de entrada y edición correspondientes al estándar seleccionado para cada tipo de archivo.
- *Validación de archivos:* en la sección *Archivo* (*imagen anterior*) puede especificar si los archivos se deben validar automáticamente al abrirlos, al guardarlos o al abrirlos y guardarlos. En la sección *Tipos de archivos* (ver punto anterior) puede deshabilitar la validación de archivos para determinados tipos de archivo.
- *Funciones de edición:* en la sección *Edición* puede especificar cómo se deben organizar los ayudantes de entrada, cómo se deben generar los elementos nuevos y si la función de finalización automática debe estar habilitada. También encontrará opciones propias de cada vista en la sección *Vista*. En las sección *Fuentes y colores* puede configurar el aspecto de cada tipo de componente de texto en todas las vistas de la aplicación.
- *Motores XSLT y FO:* en la sección *XSL* puede indicar que las transformaciones realizadas dentro de la IGU se lleven a cabo con un motor XSL externo. También deberá indicar la ubicación del ejecutable del procesador FO que se debe utilizar para procesar ficheros FO dentro de XMLSpy. Para más información, consulte la sección Procesamiento XSLT[493].
- *Codificación:* la codificación predeterminada de archivos XML y no XML se puede configurar en la sección *Codificación*.

Personalización

También puede personalizar varios aspectos de XMLSpy, incluido el aspecto de la IGU. Estas opciones de personalización están disponibles en el cuadro de diálogo "Personalizar" (*ver imagen siguiente*, (comando de menú **Herramientas | Personalizar**[1564]).



Para una descripción más detallada de las opciones de personalización, consulte la sección **Referencia del usuario**[1564].

Menús

Los diferentes comandos de menú están habilitados o deshabilitados dependiendo de tres factores: (i) el tipo de archivo, (ii) la vista activa y (iii) la posición del cursor o el estado del documento. Por ejemplo:

- *Tipo de archivo:* el comando **DTD/Esquema | Incluir otra DTD** se habilita sólo si el archivo activo es una DTD. Asimismo, los comandos del menú **WSDL** se habilitan sólo cuando está activo un archivo WSDL.
- *Vista activa:* la mayoría de los comandos del menú **Diseño de esquemas** se activarán sólo si la vista activa es la vista Esquema.
- *Posición del cursor o estado del documento:* en la vista Cuadrícula, el comando utilizado para agregar un atributo como nodo secundario (**XML | Agregar secundario | Atributo**) estará habilitado si el punto seleccionado de la vista Cuadrícula es un elemento (*posición del cursor*). Si está activo un documento XSLT, el comando **Detener el depurador** sólo se activará una vez iniciada la sesión del depurador (*estado del documento*).

© 2016-2022 Altova GmbH

Recuerde que también puede personalizar los menús (**Herramientas | Personalizar**[1583]), arrastrarlos y reorganizarlos en la IGU (consulte la sección Barra de menú, barras de herramientas y barra de estado[134]).

3.2.2 Tutoriales, proyectos y ejemplos

El paquete de instalación de XMLSpy contiene tutoriales, proyectos y archivos de ejemplo.

Ubicación de los tutoriales, proyectos y archivos de ejemplo

Los tutoriales, proyectos y archivos de ejemplo de XMLSpy se instalan en esta carpeta:

```
C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\
```

La carpeta `Mis Documentos\Altova\XMLSpy2023` se instalará para cada usuario registrado en un equipo dentro de la carpeta `<usuario>` del usuario. Por tanto, según este sistema de instalación, cada usuario tendrá su propia carpeta `Examples` en un área de trabajo distinta.

Ubicación del tutorial, del proyecto y de los archivos de ejemplo

El tutorial, el proyecto y todos los archivos de ejemplo se encuentran en la carpeta `Examples`. La ubicación exacta de los diferentes archivos es:

- *Tutorial de XMLSpy:* carpeta `Tutorial`.
- *Tutorial de la vista Authentic:* carpeta `Examples`.
- *Tutorial de WSDL:* carpeta `Examples`.
- *Archivo de proyecto:* el proyecto Examples con el que se abre XMLSpy está definido en el archivo `Examples.spp`, guardado en la carpeta `Examples`.
- *Archivos de ejemplo:* carpeta `Examples` y subcarpetas de la carpeta `Examples`.

3.2.3 Características y archivos de ayuda

El sitio web de Altova www.altova.com contiene abundante información y gran cantidad de recursos relacionados con XMLSpy. A continuación mencionamos algunos ejemplos.

Lista de características de XMLSpy

El sitio web de Altova incluye una lista actualizada de las características de XMLSpy, en la que se comparan las características y funciones disponibles en cada edición de XMLSpy (Enterprise, Professional y Standard). En la web encontrará también una lista con las características más recientes publicadas en la última versión del producto.

Ayuda de XMLSpy

La presente documentación es la ayuda creada por Altova para XMLSpy. Está disponible en el sistema integrado de ayuda de XMLSpy, al que puede tener acceso desde el menú **Ayuda** o pulsando la tecla **F1**. Además todos los manuales del usuario de los productos de Altova están disponibles en estos formatos:

- Manuales en línea en formato HTML, a los que puede tener acceso desde la página Soporte del sitio web de Altova
- Manuales en PDF para imprimir, que se pueden descargar desde el sitio web de Altova
- Copias impresas, que se pueden comprar en el sitio web de Altova

Opciones de soporte técnico

Si necesita más detalles sobre la información disponible en el *Manual del usuario* (la presente documentación) o tiene alguna pregunta sobre los productos de Altova, visite el Centro de soporte técnico del sitio web de Altova. Allí encontrará:

- Enlaces a nuestras páginas de preguntas frecuentes
- Foros de debate sobre los productos de Altova y temas relacionados con XML
- Formularios de soporte técnico para enviarnos solicitudes de soporte técnico (si cuenta con un paquete de soporte). Nuestro equipo de soporte técnico se encargará de responder a su solicitud.

Productos de Altova

El sitio web de Altova incluye una lista con todos los productos de Altova.

© 2016-2022 Altova GmbH

4 Vistas de edición

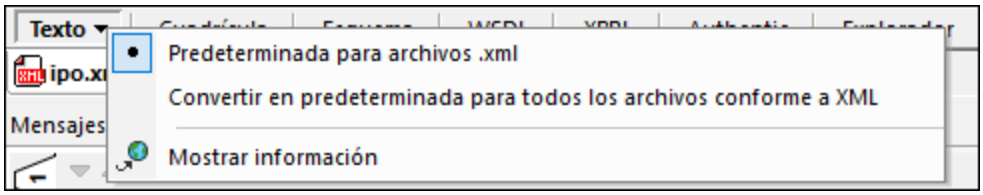
XMLSpy contiene varias vistas de edición muy potentes. Además de su vista Texto, cargada de funciones de edición inteligente, también ofrece vistas gráficas que facilitan considerablemente la edición de documentos. Dependiendo del tipo de documento que esté activo en XMLSpy, la ventana principal incluirá una o varias vistas de edición. Por ejemplo, cuando esté activo un archivo Office Open XML o ZIP, la ventana principal contendrá una sola vista de edición: la vista Archivo. Cuando esté activo un documento HTML, habrá dos vistas de edición: las vistas Texto y Explorador. Cuando esté activo un documento XML, habrá siete vistas de edición: las vistas Texto, Cuadrícula, Esquema, WSDL, XBRL, Authentic y Explorador. De estas siete vistas, la vista Esquema se habilitará sólo con documentos de esquema XML, la vista WSDL se habilitará sólo con documentos WSDL, etc.

Esta sección describimos las distintas vistas de edición que ofrece XMLSpy:

- Vista Texto 145
- Vista Cuadrícula 169
- Vista Esquema 171
- Vista WSDL 251
- Vista XBRL 264
- Vista Authentic 613
- Vista Explorador 279
- Vista Archivo 281

Selección de la vista predeterminada

Un documento de un tipo concreto (por ejemplo, un documento XML o JSON) se puede visualizar de distintas maneras. Puede seleccionar una vista predeterminada directamente en la interfaz; para ello haga clic o haga clic con el botón derecho en la pestaña actual de visualización y seleccione la opción apropiada en el menú que aparece (*imagen siguiente):*



Las opciones de este menú son:

- *Predeterminada para una extensión de archivo:* la vista de edición actual se convierte en la vista predeterminada para los archivos que tengan la misma extensión que el archivo activo (por ejemplo, archivos **.xml** o **.xq**).
- *Predeterminada para todos los archivos conforme a:* la vista de edición actual se convierte en la vista predeterminada para los archivos que sigan el mismo esquema (por ejemplo, todos los archivos conforme a XML o conforme a XQuery). En este caso puede tratarse de un conjunto de archivos mayor que el que abarca una extensión de archivo; por ejemplo, tanto la extensión **.xq** como la extensión **.xquery** pueden conformar con XQuery.
- *Mostrar información:* vincula con una página del sitio web de Altova que describe las características de la vista de edición actual.

Si se selecciona una de las opciones de vista predeterminada, aparecerá un botón de opción junto a ella (*imagen anterior*).

En estos casos no se puede cambiar la vista de edición predeterminada:

- *Archivos de esquema JSON:* estos archivos, como los archivos de instancia JSON, tienen una extensión **.json**, pero por el contenido se detectan como esquemas JSON, por lo que siempre se abren con la Vista Esquema JSON 714.
- *Archivos XML habilitados para Authentic:* son archivos XML (con una extensión **.xml**) que se han asignado a una hoja de estilos de Altova StyleVision, lo que permite visualizar el documento XML como un documento XML de Authentic. Estos archivos se abren siempre con la Vista Authentic 278.
- *Archivos XSD para XBRL (taxonomías XBRL):* estos archivos tienen una extensión **.xsd**, pero por el contenido se detectan como taxonomías XBRL (en vez de esquemas XML), por lo que se abren siempre con la Vista XBRL 264.

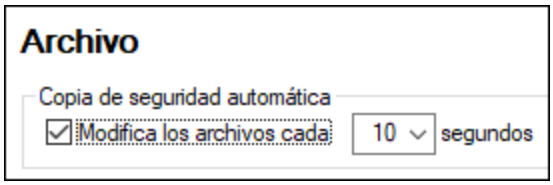
Nota: las normas que sigue cada extensión de archivo y la vista de edición predeterminada que corresponde a cada tipo de archivo individual se puede configurar en la sección Tipos de archivo del cuadro de diálogo "Opciones" 1587. La selección automática de vista predeterminada que acabamos de explicar permite configurar esta opción de forma rápida y fácil.

© 2016-2022 Altova GmbH

4.1 Copia de seguridad automática de archivos

Si modifica archivos con XMLSpy, la aplicación guarda una copia de seguridad automáticamente a intervalos regulares. En la pestaña Archivo del cuadro de diálogo "Opciones" (**Herramientas | Opciones | Archivo**[1584]) que se ve en la imagen siguiente, puede:

- Activar/desactivar las copias de seguridad automáticas
- Indicar la frecuencia de las copias de seguridad (entre 5 y 300 segundos)



Se pueden guardar como copias de todos los tipos de archivos que se pueden editar en XMLSpy. No se guardan copias de seguridad de archivos ZIP.

Indicadores

Las pestañas de los archivos que se encuentran en la parte inferior de la ventana principal muestran símbolos a la derecha del nombre del archivo; esos símbolos indican si existe una copia de seguridad de ese archivo y si esa copia de seguridad se ha guardado o no (*imagen siguiente*).



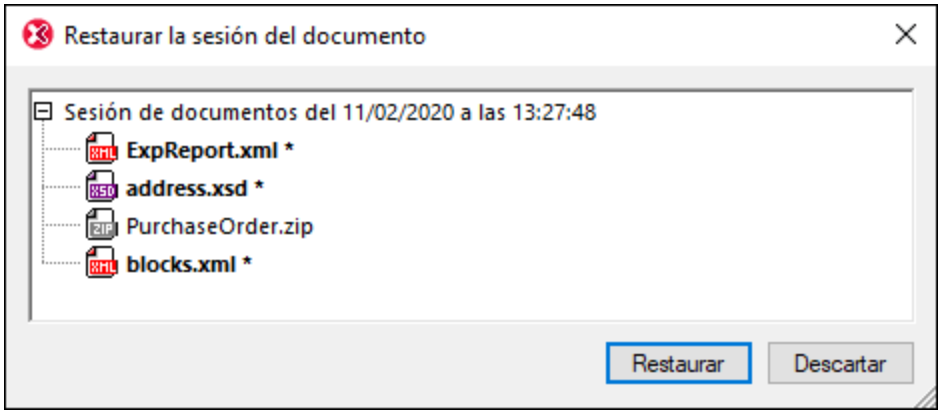
Guardada/No guardada
Si junto al nombre del archivo hay un círculo de color, significa que el archivo ha sido modificado. Si no hay ningún círculo, significa que el archivo no se ha modificado desde que se abrió o desde la última vez que se guardó. En la imagen anterior, por ejemplo, el archivo `address.xml` no se ha modificado.

Estado de la copia de seguridad
El color del círculo indica el estado de la copia de seguridad del archivo.

- *Amarillo:* el archivo se ha modificado, pero no se ha guardado o no hay una copia de seguridad.
- *Verde:* se ha hecho una copia de seguridad del archivo y este no se ha modificado desde entonces pero tampoco se ha guardado (si se guarda el círculo desaparece).
- *Rojo:* no se puede hacer una copia de seguridad de este archivo o ha ocurrido un error al intentar hacerla.
- *Gris:* la función de copia de seguridad automática se ha deshabilitado (*desde el cuadro de diálogo "Opciones"*[1584]*; véase más arriba*). La presencia del círculo indica que el archivo no se ha guardado desde la última modificación (si se hubiera guardado no habría ningún círculo).

Restaurar una copia de seguridad

Si XMLSpy se cierra de forma inesperada, la siguiente vez que se inicie la aplicación aparecerá u cuadro de diálogo de restauración con todos los documentos que estuvieran abiertos cuando XMLSpy se cerró (*imagen siguiente*). Para ver la ruta de acceso de cada uno de los archivos basta con pasar el cursor del ratón por encima del mismo. En el caso de archivos temporales sin guardar la ruta de acceso será la que aparezca por defecto actualmente al abrir el cuadro de diálogo "Guardar como" para ese archivo.



El estilo de fuente de los nombres de archivo que aparecen en este cuadro de diálogo, así como la presencia o ausencia de asteriscos, también proporcionan información:

- El estilo de fuente negrita y un asterisco indican que el archivo contiene cambios sin guardar. Al restaurar este archivo se recupera la última copia de seguridad que se haya guardado.
- El estilo normal indica que el archivo se guardó y no hay cambios sin guardar. Al restaurar este archivo se recupera la última versión que se guardó de él.
- Si el nombre del archivo aparece en gris, significa que el archivo ni se guardó ni hay una copia de seguridad de él (por ejemplo, porque es un archivo nuevo que no se llegó a editar). Este tipo de archivos no se pueden restaurar.

Desde este cuadro de diálogo puede:

- Hacer clic en **Restaurar** para restaurar los archivos en la IGU a partir de la última versión de su copia de seguridad.
- Haga clic en **Descartar** para no abrir ninguno de los archivos de la lista y descartar sus copias de seguridad.

© 2016-2022 Altova GmbH

4.2 Vista Texto

En la vista Texto (*ver imagen siguiente*) puede escribir directamente el texto de su documento (tanto código como contenido). En la vista Texto puede editar cualquier tipo de documento, incluso documentos no XML (como documentos HTML y XQuery). La vista incluye varias funciones que le ayudarán a editar el documento con rapidez y precisión.



En esta sección describimos características generales de la vista Texto que están disponibles para todo tipo de documentos. Algunos tipos de documentos, como XML, XQuery y CSS, cuentan con funciones propias, que se describen en las secciones dedicadas a dichos tipos de documentos. Por ejemplo, la sección XML | Editar datos XML en la vista Texto [290] describe las funciones propias de los documentos XML.

Las características generales de la vista Texto pueden agruparse en cuatro apartados:

- Formato en la vista Texto [146]. Este apartado describe cómo se configuran las propiedades de fuente, sangría y ajuste de líneas del documento.
- Visualización de documentos [149]. Este apartado contiene información sobre funciones de numeración de líneas, marcadores, expansión y contracción de nodos y otras funciones relacionadas con la visualización de documentos.
- Edición en la vista Texto [151]. Este apartado describe las funciones disponibles durante la edición, principalmente las funciones de edición inteligente.
- Navegar por los documentos [154]. Este apartado explica cómo se puede navegar por un documento en la vista Texto.
- Búsqueda y reemplazo [157]. Aquí se describen las funciones de búsqueda y reemplazo de la vista Texto.

- Ayudantes de entrada[164]. Los ayudantes de entrada son ventanas que ofrecen opciones contextuales de entrada de datos. Por ejemplo, un ayudante de entrada muestra los elementos o atributos que se pueden agregar de forma válida en una posición del documento. Las opciones de entrada del ayudante se pueden insertar con solo hacer doble clic en ellas.
- Vista dividida[165]. Esta función divide en dos la ventana de la vista Texto en la que se visualiza el documento activo. Esto permite ver dos partes distintas del mismo documento una junto a otra, lo cual resulta útil cuando se trabaja con documentos largos.
- Teclas de acceso rápido[167]. Contiene los atajos estándar de los comandos más usados en la vista Texto.

Cambiar a la vista Texto

Para abrir la vista Texto de un documento, haga clic en la pestaña **Texto** situada en la parte inferior de la ventana del documento o seleccione la opción de menú **Vista | Vista Texto**.

4.2.1 Formato en la vista Texto

La vista Texto ofrece varias opciones de formato, descritas a continuación.

Fuentes

Puede personalizar la familia de fuentes, el tamaño de la fuente y el color de fondo del texto para cada uno de estos grupos de documentos: (i) documentos XML genéricos (incluidos documentos HTML), (ii) documentos XQuery y (iii) documentos CSS.

Puede aplicar colores diferentes a los elementos de texto de un documento que tengan diferencias semánticas. Por ejemplo, puede aplicar un color a los nombres de elementos, otro a los nombres de atributo y otro al contenido de los elementos. Cuando se definen colores diferentes para elementos de texto diferentes, se habilita la función de color de sintaxis. Las fuentes del texto se pueden personalizar en la sección *Fuentes y colores*[1603] de la vista Texto del cuadro de diálogo "Opciones"[1603]. Para más información sobre cómo configurar estas opciones de configuración, consulte la sección Herramientas | Opciones | Fuentes y colores[1603] de la presente documentación.

© 2016-2022 Altova GmbH

Sangría

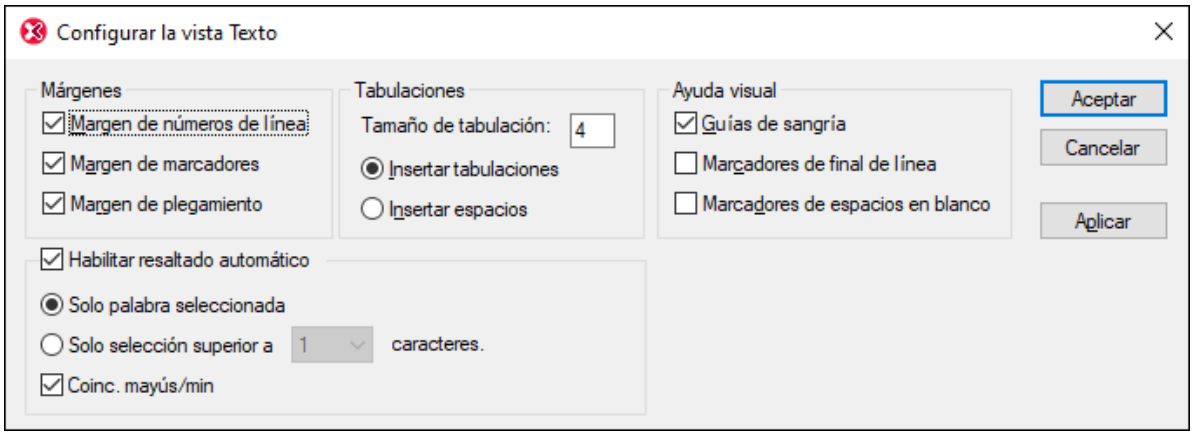
Puede indentar el documento para que sea visible la estructura, como se ve a continuación. Cuando el documento aparece con este tipo de sangría jerárquica hablamos de formato pretty-print. En un documento con este formato, la sangría aumenta conforme se va descendiendo de nivel en la estructura. Para visualizar un documento en formato pretty-print debe: (i) configurar su sangría (según usted prefiera) y (ii) aplicarle el formato pretty-print.



Para configurar y aplicar el formato pretty-print, siga los pasos que describimos a continuación. La configuración del formato pretty-print se aplica a nivel de toda la aplicación. Esto quiere decir que una vez haya guardado la configuración del formato pretty-print, esta se aplica a todos los documentos XML y JSON. Es decir, solo tiene que cambiar la configuración si quiere cambiar el aspecto de sus documentos cuando se visualizan con formato pretty-print. Una vez haya hecho estos cambios a nivel de la aplicación debe aplicar el formato pretty-print de forma individual a cada uno de los documentos. Un documento siempre aparece con el formato pretty-print que se le haya aplicado y ese formato no cambia hasta que no se le vuelva a aplicar un formato pretty-print.

1. En la sección Pretty-print[1591] del cuadro de diálogo Opciones (**Herramientas | Opciones**[1583]), habilite el formato pretty-print marcando la casilla *Usar la sangría indicada en la pestaña de configuración de la vista Texto*. Esta es la opción que activa la sangría que se ve en los documentos con formato pretty-print. Si no se marca la casilla *Aplicar sangría,* todas las líneas del documento tendrán una sangría cero.
2. En la sección Pretty-print[1591] (*véase punto anterior*), haga clic en el botón **Configurar la vista Texto** (en la parte superior derecha) para ir al cuadro de diálogo Configurar la vista Texto (*imagen siguiente*) y configurar la sangría. (Otras formas de llegar a este cuadro de diálogo es a través (i) del comando de

menú **Vista | Configurar la vista Texto**[1489] y (ii) del icono **Configurar la vista Texto** de la barra de herramientas.) En el cuadro de diálogo Configurar la vista, el campo *Tamaño de tabulación* indica el número de espacios que ocupan una tabulación y la sangría. Justo debajo puede elegir si quiere que la sangría se componga de tabulaciones o de espacios. En ambos casos la sangría equivale al número **x** de espacios que indique en la casilla *Tamaño de tabulación* (ya que 1 tabulación = X espacios).



3. Una vez haya configurado el formato pretty-print a nivel de la aplicación como explicamos en los pasos anteriores, puede aplicarlo a documentos individuales. Asegúrese de que el documento al que quiere aplicar el formato pretty-print es el documento activo. Haga clic en el comando **Edición | Pretty-Print**[1288] o en el icono **Pretty-print** de la barra de herramientas Texto. Esto hará que el texto del documento se presente con el formato que haya indicado en la sección Pretty Print[1591] del cuadro de diálogo "Opciones". Al hacer clic en el comando **Pretty-print** se eliminan espacios en blanco al principio o al final de línea.

Nota: la longitud de la sangría de una línea se indica mediante guías de sangría, unas líneas de puntos verticales (*ver la primera imagen de esta sección*) que se pueden activar o desactivar con solo marcar la casilla *Guías de sangría* del panel Ayuda visual del cuadro de diálogo "Configurar la vista texto" (*ver imagen anterior*).

Nota: para más información sobre espacios en blanco consulte Espacios en blanco[337].

Usar tabulaciones y espacios para aplicar formato

También puede usar tabulaciones y espacios para formatear texto, sobre todo en documentos no XML, para los que la opción pretty-print no está disponible. Al pulsar la tecla **Entrar** o **Mayús+Entrar**, el cursor salta a la siguiente línea, en la posición correspondiente a la posición inicial de la línea anterior.

Ajuste automático de línea

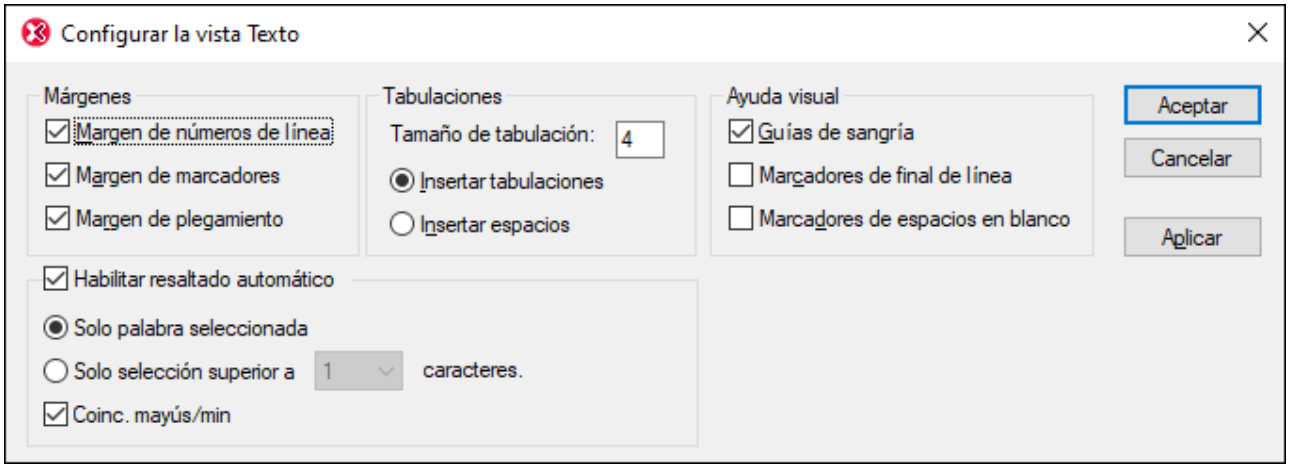
Cuando las líneas de texto sobrepasen la amplitud de la ventana principal, se puede utilizar el comando **Vista | Ajuste automático de línea**[1488]. El icono de este comando está disponible en la barra de herramientas Texto[149].

© 2016-2022 Altova GmbH

4.2.2 Visualización de documentos

La vista Texto ofrece funciones visuales que facilitan la visualización y edición de secciones de texto de gran tamaño. Algunas funciones prácticas son: (i) los números de línea[149], (ii) los marcadores[149], (iii) el plegamiento de código[150] (expandir y contraer la visualización de nodos), (iv) las guías de sangría[151] y (v) los marcadores de final de línea y de espacios en blanco[151]. Estos comandos están disponibles en el cuadro de diálogo "Configurar la vista Texto" (*ver la primera imagen que aparece a continuación*) y en la barra de herramientas Texto (*segunda imagen*).

Para abrir el cuadro de diálogo "Configurar la vista Texto"[1489] seleccione el comando **Vista | Configurar la vista Texto**, el botón **Configurar la vista Texto** de la barra de herramientas o el menú contextual de la vista Texto. Las opciones de configuración del cuadro de diálogo "Configurar la vista Texto" afectan a toda la aplicación, no sólo al documento activo.





También son muy prácticas la función de Zoom[151] y de navegación y búsqueda[154].

Números de línea

Los números de línea aparecen en el margen de números de línea (*ver imagen siguiente*). Este margen se puede activar o desactivar en el cuadro de diálogo "Configurar la vista Texto" (*ver imagen anterior*). Cuando se contrae una sección de texto, los números de línea del texto contraído también se ocultan. El comando Ir a la línea o carácter[157] está relacionado con los números de línea.

Marcadores

Las líneas del documento se pueden marcar para realizar consultas rápidas más adelante. Si el margen de marcadores está activado, los marcadores aparecen en el margen de marcadores. De lo contrario, las líneas marcadas se resaltan en color cian.

El margen de marcadores se puede activar o desactivar en el cuadro de diálogo "Configurar la vista Texto" (*ver imagen anterior*).

El menú **Edición** y la barra de herramientas Texto incluyen comandos para editar los marcadores y navegar por ellos. Para insertar un marcador, utilice el comando **Edición | Insertar o quitar marcador**. Para quitar un marcador, seleccione el marcador y después seleccione el comando **Edición | Insertar o quitar marcador**. Para navegar por los marcadores de un documento, utilice los comandos **Edición | Ir al siguiente marcador** y **Edición | Ir al marcador anterior**. La barra de herramientas Texto también incluye iconos para estos comandos (*ver imagen anterior*).

Plegamiento de código

La característica plegamiento de código permite expandir y contraer nodos en documentos XML, XQuery, JSON y CSS. Los nodos que se pueden expandir/contraer se marcan en el margen de plegamiento de código con el signo +/- (*imagen siguiente*). El margen puede activarse o desactivarse en el cuadro de diálogo "Configurar la vista Texto". En el ejemplo de la imagen siguiente puede ver que hay tres nodos contraídos: el elemento `shipTo` y dos elementos `item`. Cuando está contraído, el nodo se marca con puntos suspensivos (ver marca verde en la imagen). Si pasamos el puntero por encima de los puntos suspensivos, aparece en pantalla un cuadro emergente con el contenido del nodo contraído (ver marca azul en la imagen). Si el contenido del nodo es demasiado largo para el cuadro emergente, esto se indica con puntos suspensivos también.



El icono **Expandir o contraer todos los plegamientos** de la barra de herramientas Texto expande **todos** los nodos o los contrae hasta el elemento de nivel superior.

Estas son las diferentes opciones que ofrece el icono +/- de los nodos:

Clic en [-]	Contrae el nodo.
Clic en [+]	Expande el nodo y sus nodos descendientes aparecen expandidos o contraídos, dependiendo de cuál fuera su estado antes de contraer el nodo matriz.

© 2016-2022 Altova GmbH

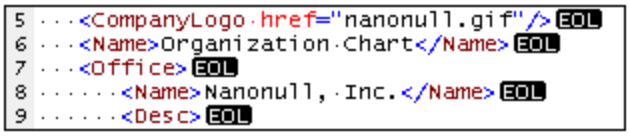
Mayús+clic en [–]	Contrae todos los nodos descendientes, pero el nodo en el que se hizo clic permanece expandido.
Ctrl+clic en [+]	Expande el nodo y todos sus descendientes.

Guías de sangría

Las guías de sangría son líneas de puntos verticales que indican la longitud de la sangría de una línea (*ver imagen anterior*). Las guías de sangría se pueden activar o desactivar en el cuadro de diálogo "Configurar la vista Texto".

Marcadores de final de línea y marcadores de espacios en blanco

Los marcadores de final de línea (EOL) y los marcadores de espacios en blanco se pueden activar o desactivar en el cuadro de diálogo "Configurar la vista Texto". La captura de pantalla que aparece a continuación incluye estos marcadores. Cada punto representa un espacio en blanco.



Alejarse y acercarse con el zoom

Puede acercarse y alejarse en la vista Texto si mueve la rueda del ratón mientras pulsa la tecla **Ctrl**. De este modo podrá aumentar y reducir el tamaño del texto de la vista Texto. Si desea aumentar el tamaño de la fuente, cambie las opciones de configuración en el cuadro de diálogo "Opciones" 146.

4.2.3 Edición en la vista Texto

Las funciones de edición de texto descritas en este apartado están disponibles en la vista Texto para todos los tipos de documento. Las funciones de edición de texto descritas en este apartado están disponibles en la vista Texto para todos los tipos de documento. Estas funciones se suman a las funciones típicas de edición, como **Cortar**, **Copiar**, **Pegar**, **Eliminar** y **Seleccionar todo** (estos comandos también están disponibles en el menú **Edición**).

- Color de sintaxis 152
- Búsqueda de la etiqueta inicial y la etiqueta final 153
- Edición inteligente 152
- Finalización automática 153
- Subir o bajar los elementos del mismo nivel 154
- Seleccionar un elemento entero e ir a su primario 154
- Operaciones de arrastrar y colocar y menús contextuales 154
- Número ilimitado de operaciones de deshacer 154
- Corrección ortográfica 154

La vista Texto incluye algunas funciones especiales para determinados tipos de documentos (como XML 285 y XQuery 505). Estas funciones se describen en las secciones dedicadas a dichos tipos de documento.

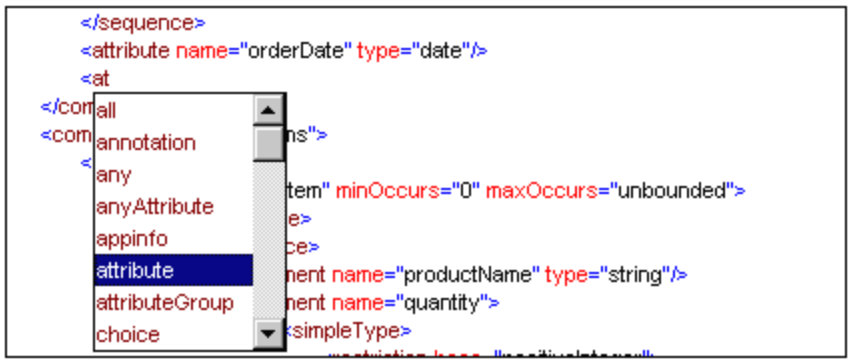
Nota: cuando esté trabajando con archivos de gran tamaño, puede deshabilitar la función de finalización automática y los ayudantes de entrada para que la carga y edición de estos archivos sea más rápida. En estos casos el usuario puede indicar el tamaño de archivo máximo. Para más información, consulte la sección Opciones | Edición[1590] de la Comandos de menú.

Color de sintaxis

El color de sintaxis se aplica de acuerdo al valor semántico del texto. Por ejemplo, en los documentos XML, dependiendo de si el nodo XML es un elemento, un atributo, contenido, una sección , un comentario o una instrucción de procesamiento, el nombre del nodo (y en ocasiones el contenido del nodo) tendrá un color diferente. Podemos distinguir cuatro grupos de tipos de documento: (i) XML genérico (que incluye documentos HTML), (ii) XQuery, (iii) CSS y (iv) JSON. Las propiedades de texto (incluido el color) de cada grupo se pueden configurar en la sección Fuentes y colores: Vista Texto del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).

Elementos repetidos

Si está trabajando con un documento XML basado en un esquema, XMLSpy le ofrece varias opciones de edición inteligente en la vista Texto. Gracias a esta función podrá insertar rápidamente el elemento, atributo o valor de atributo correcto de acuerdo con el modelo de contenido que se definió en el esquema para el elemento que esté editando. Por ejemplo, si escribe la etiqueta inicial de un elemento, aparecerá una lista con los nombres de todos los elementos permitidos en esa posición de acuerdo con el esquema (*ver imagen siguiente*). Seleccione el nombre del elemento que necesita y pulse la tecla **Entrar**. El elemento se insertará en la etiqueta inicial. Además, después de que el usuario cree la etiqueta inicial, XMLSpy añade automáticamente la etiqueta final (*véase* ***[153] *Finalización automática más abajo*).



Estas listas emergentes también aparecen en todos estos casos:

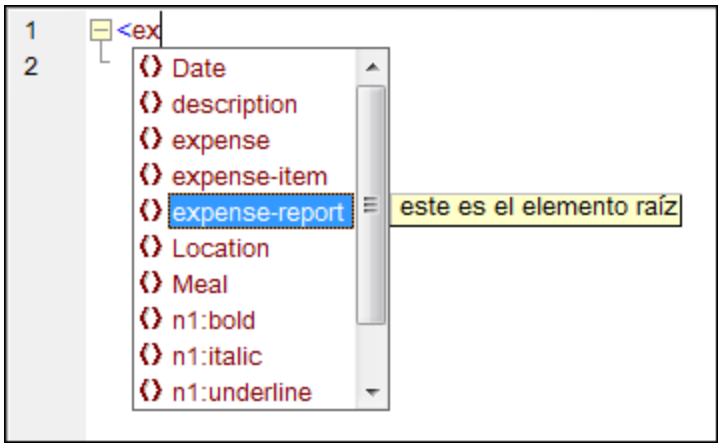
- Cuando el cursor está dentro de la etiqueta inicial de un elemento para el que se definió un atributo y se pulsa la barra espaciadora. La lista emergente incluirá todos los atributos disponibles.
- Cuando el cursor está entre las comillas que delimitan el valor de un atributo que tiene valores enumerados. La lista emergente incluirá todos los valores enumerados.
- Cuando escriba `</` (que indica el inicio de una etiqueta final), la lista emergente incluye el nombre del elemento que se dispone a cerrar.
- Cuando desee escribir un elemento vacío con una sola etiqueta o convertir un elemento vacío formado por dos etiquetas en un elemento vacío con una sola etiqueta, escriba la barra de cierre después del nombre del elemento: `<elemento/`. Si la etiqueta final ya existe, ésta se eliminará: `<elemento/>`.

© 2016-2022 Altova GmbH

Finalización automática

Cuando se editan documentos XML u otros documentos marcados (como HTML) es fácil cometer errores y que el formato del documento no sea correcto. Por ejemplo, pueden faltar las etiquetas finales, estas pueden estar mal escritas o pueden estar en una posición no válida. XMLSpy puede finalizar automáticamente la etiqueta inicial y final de los elementos e insertar todos los atributos necesarios en cuanto el usuario termina de teclear el nombre del elemento. El cursor se coloca automáticamente entre la etiqueta inicial y final del elemento, para que pueda seguir añadiendo contenido o elementos secundarios inmediatamente: `<img src="" alt=""> </img>`

XMLSpy se sirve de las normas XML sobre formato correcto y validez para ofrecer esta función de finalización automática. La información sobre la estructura del documento se obtiene del esquema en el que se basa el documento. (En el caso de esquemas de uso generalizado, como HTML y XSLT, la información del esquema está integrada en XMLSpy.) La función de finalización automática no sólo utiliza la información sobre la estructura del documento. Por ejemplo, la función de finalización automática utiliza de forma activa las enumeraciones y las anotaciones de esquema de un esquema XML. Si en el esquema se enumeran valores para un nodo determinado, esas enumeraciones aparecerán en las opciones de finalización automática cuando se disponga a editar dicho nodo. Asimismo, si el esquema incluye anotaciones para un nodo determinado, esas anotaciones aparecerán cuando empiece a escribir el nombre de dicho nodo en el documento (*ver imagen siguiente*). (*Este es el elemento raíz* es la anotación de esquema del elemento `expense-report`.)



La función de finalización automática se puede activar o desactivar en la sección Edición del cuadro de diálogo "Opciones" 1590 (**Herramientas | Opciones | Edición**).

Búsqueda de la etiqueta inicial y la etiqueta final

Si coloca el cursor dentro de una etiqueta inicial o final de un elemento de marcado y pulsa **Ctrl+E**, la selección se desplaza a la otra etiqueta del elemento. Si pulsa **Ctrl+E** repetidas veces, puede cambiar de la etiqueta inicial a la etiqueta final y viceversa. Esta característica es muy útil a la hora de localizar la etiqueta inicial y final de un elemento XML. Además, los nombres de los elementos se resaltan con dos colores distintos 1603 para indicar si los nombres de las etiquetas coinciden o no. Se trata de una práctica ayuda visual. Los colores que se utilizan para resaltar los nombres de los elementos se pueden configurar en el cuadro de diálogo Opciones 1603. Cuando el usuario edita el nombre de un elemento en la etiqueta inicial, la aplicación edita automáticamente la etiqueta final para que coincidan.

Subir o bajar los elementos del mismo nivel

Si coloca el cursor dentro de un elemento y pulsa **Alt+FlechaArriba** o **Alt+FlechaAbajo**, el elemento seleccionado se coloca encima o debajo del elemento del mismo nivel que esté antes o después de él.

Seleccionar un elemento entero e ir a su primario

Si coloca el cursor dentro de un elemento y pulsa **Ctrl+Mayús+E**, se selecciona el elemento entero. Haga clic en **Ctrl+Alt+E** para que la aplicación resalte el elemento primario. Con estas dos teclas de acceso rápido podrá encontrar rápidamente la posición actual del cursor dentro de la estructura del documento.

Operaciones de arrastrar y colocar y menús contextuales

Además, si hace clic con el botón derecho en el texto, el menú contextual que aparece contiene comandos de edición frecuentes (como Cortar[1280], Copiar[1280], Pegar[1280], Eliminar[1280], Enviar por correo[1274] e Ir a la línea o carácter[1488]).

Número ilimitado de operaciones de deshacer

XMLSpy ofrece un número ilimitado de operaciones de Deshacer[1280] y Rehacer[1280] para todas las operaciones de edición.

Corrección ortográfica

En la vista Texto puede revisar la ortografía de los documentos con ayuda de los diccionarios integrados. También puede crear diccionarios del usuario y editarlos con palabras que no formen parte del diccionario. Para más información, consulte la descripción de los comandos Ortografía[1539] y Opciones de ortografía[1542].

4.2.4 Navegar por los documentos

Estas son las funciones que puede utilizar para navegar por el documento en la vista Texto:

- La función de resaltado de texto[154] le ayudará a encontrar todas las coincidencias de la cadena de texto o de la palabra que seleccione. Cada coincidencia aparece en la barra de desplazamiento para que pueda navegar con facilidad por todas las coincidencias.
- Vista general del documento en la barra de desplazamiento[156]. Esta característica muestra la posición relativa del cursor y de la selección de texto dentro del documento.
- Ir a la línea o carácter[157]. Esta característica lleva al usuario a la línea y carácter especificados directamente.

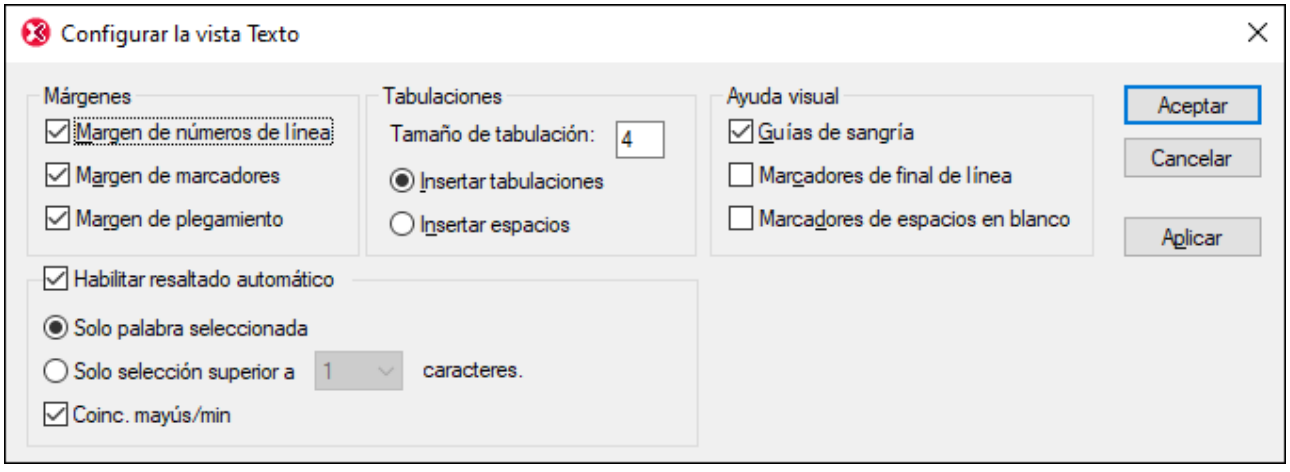
Resaltado de texto

Si se habilita la función de resaltado de texto en el cuadro de diálogo "Configurar la vista Texto" (**Vista | Configurar la vista Texto**[1489]), la aplicación resalta todas las coincidencias del texto seleccionado por el usuario. La selección se resalta en azul pálido y las coincidencias se resaltan en naranja pálido (*imagen siguiente*). La selección y sus coincidencias aparecen marcadas en la barra de desplazamiento por medio de marcadores grises cuadrados. Observe que la barra de desplazamiento también indica la posición actual del cursor por medio de un marcador azul.

© 2016-2022 Altova GmbH

```
<expense-report xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ExpReport.xsd" currency="USD" detailed="false"
total-sum="556.9">
    <Person>
        <First>Fred</First>
        <Last>Landis</Last>
        <Title>Project Manager</Title>
        <Phone>123-456-7890</Phone>
        <Email>f.landis@nanonull.com</Email>
    </Person>
    <expense-item type="Lodging" expto="Sales">
        <Date>2003-01-01</Date>
        <expense>122.11</expense>
    </expense-item>
    <expense-item type="Lodging" expto="Development">
        <Date>2003-01-02</Date>
        <expense>122.12</expense>
        <description>Played penny arcade</description>
    </expense-item>
    <expense-item type="Lodging" expto="Marketing">
        <Date>2003-01-02</Date>
        <expense>299.45</expense>
        <description>Treated Clients</description>
    </expense-item>
    <expense-item type="Entertainment" expto="Development">
        <Date>2003-01-02</Date>
        <expense>13.22</expense>
        <Misc misctype="TeamBuilding"/>
        <description>Bought signed XMLSpy Handbook</description>
    </expense-item>
</expense-report>
```

Para habilitar la función de resaltado basta con marcar la casilla *Habilitar resaltado automático* en el cuadro de diálogo "Configurar la vista Texto" (**Vista | Configurar la vista Texto**[1489]). Puede definir como selección una palabra entera o un número fijo de caracteres. También puede especificar si se deben tener en cuenta o no las mayúsculas y minúsculas.



Debe tener en cuenta estos aspectos de la función de resaltado:

- En el caso de la selección de caracteres, podrá especificar el número mínimo de caracteres que deben coincidir con la selección, empezando por el primer carácter de la selección. Por ejemplo, puede elegir que coincidan dos caracteres o más. En este caso, si selecciona un solo carácter del

documento, no se encontrarán coincidencias. Pero si selecciona dos o más caracteres, sí se encontrarán coincidencias. Por ejemplo, si selecciona `t`, la aplicación no encontrará coincidencias. Pero si selecciona `ty`, la aplicación mostrará todas las coincidencias que contengan `ty`; si selecciona `typ`, la aplicación mostrará todas las coincidencias que contengan `typ` y así sucesivamente.

- Cuando se trate de búsquedas de palabras, la aplicación considera que estos elementos son palabras independientes: nombres de elementos (sin paréntesis angulares), paréntesis angulares de etiquetas de elementos, nombres de atributos y valores de atributos sin comillas.

Nota: la función de búsqueda de la etiqueta inicial y la etiqueta final[153] es una característica distinta que no se ve afectada por el estado de la casilla *Habilitar resaltado automático*. Los colores utilizados en esta función se pueden configurar en la sección *Fuentes y colores*[1603] del cuadro de diálogo "Opciones"[1603] (**Herramientas | Opciones**).

Vista general del documento en la barra de desplazamiento

La barra de desplazamiento ofrece estas características:

- Muestra la relación de tamaño entre: (i) el documento entero (la barra de desplazamiento), (ii) el segmento del documento que aparece en pantalla (control de posición), (iii) la selección de texto actual (barra azul) y (iv) la posición actual del cursor (marcador de cursor).
- Permite navegar por el documento (i) arrastrando el control de posición de la barra de desplazamiento hacia arriba y hacia abajo o (ii) haciendo clic en las flechas **Subir** y **Bajar** (marcadas con un círculo verde en la imagen siguiente).



Debe tener en cuenta estos aspectos de la barra de desplazamiento:

- La longitud de la barra de desplazamiento corresponde a la longitud total del documento.
- Si en la ventana solo cabe una parte del documento, esta porción del documento corresponde al control de posición de la barra de desplazamiento (*imagen anterior*). El control de posición puede moverse hacia arriba y hacia abajo para ver en pantalla el resto del documento. Es decir, el control de posición representa la ventana.
- La selección de texto actual se marca con una barra azul en la barra de desplazamiento. El tamaño de la barra azul en relación al tamaño de la barra de desplazamiento es proporcional al tamaño de la

 © 2016-2022 Altova GmbH

selección de texto en relación al tamaño total del documento. Si la selección de texto no es superior a una línea, la barra azul no aparecerá.
- La posición del cursor se indica por medio de un marcador de color azul oscuro. La posición relativa del marcador en la barra de desplazamiento corresponde a la posición relativa del cursor en el documento.

Ir a la línea o carácter

Este comando del menú **Vista** y de la barra de herramientas Texto permite ir directamente a una línea o a un carácter determinados del texto del documento.

4.2.5 Buscar y reemplazar

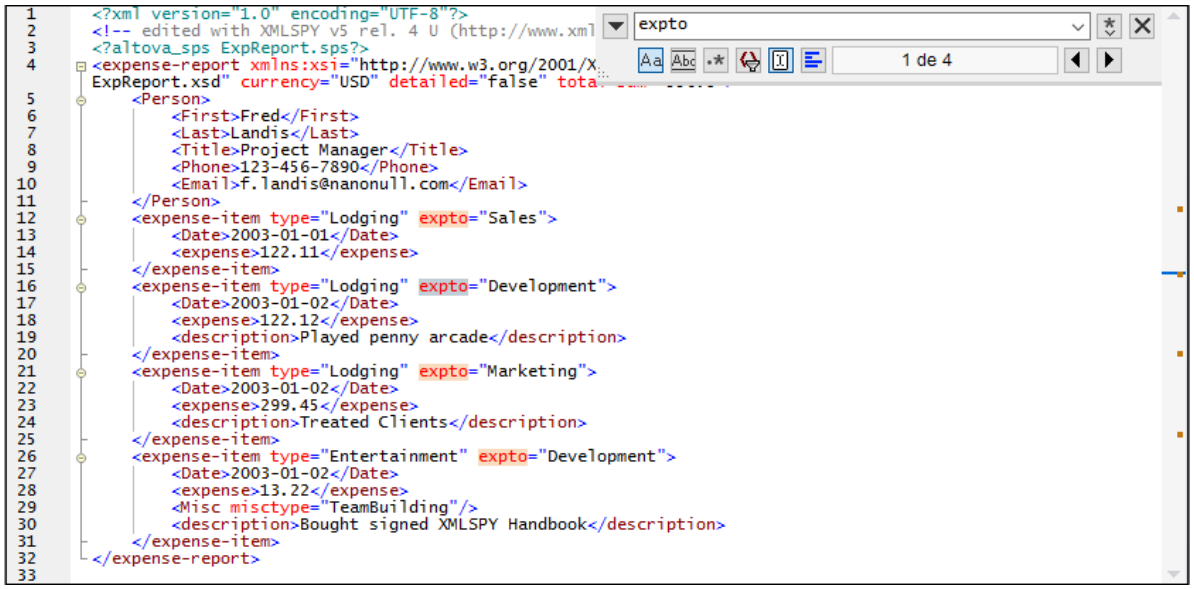
Temas de este apartado:

- Buscar texto en un documento 157
- Opciones de búsqueda 158
- Reemplazar texto 159
- Uso de expresiones regulares 160
- Ejemplos de expresiones regulares 161

Buscar texto en un documento

Pulse **Ctrl+F** (o seleccione el comando de menú **Edición | Buscar**) para abrir el cuadro de diálogo "Buscar" (*imagen siguiente, esquina superior derecha*). El término de búsqueda introducido en este cuadro de diálogo puede buscarse en todo el documento o dentro de una selección de texto.

- Introduzca la cadena de texto que desea buscar o utilice el cuadro combinado para seleccionar una cadena de las últimas 10 búsquedas realizadas.
- Una vez introducida o seleccionada la cadena de búsqueda, todos resultados se resaltan y la posición de los resultados se señala por medio de marcadores de color beige en la barra de desplazamiento (*imagen siguiente*).
- El resultado seleccionado por el usuario se marca con un color distinto al de los demás resultados y su posición en la barra de desplazamiento se señala con un marcador de color azul oscuro.
- El número total de resultados aparece justo debajo del término de búsqueda, junto con la posición de índice del resultado seleccionado por el usuario (*imagen siguiente*). Por ejemplo, `2 de 4` indica que el usuario tiene seleccionado el segundo de los cuatro resultados.
- Puede recorrer los resultados en ambos sentidos seleccionado los botones **Anterior (Mayús+F3)** y **Siguiente (F3)** situados en la esquina inferior derecha del cuadro de diálogo.



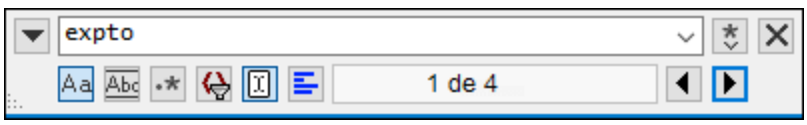
- Para cambiar del modo **Buscar** al modo **Reemplazar** y viceversa haga clic en el botón de flecha situado en la esquina superior izquierda del cuadro de diálogo.
- Para cerrar el cuadro de diálogo haga clic en el botón **Cerrar** situado en la esquina superior derecha.

Debe tener en cuenta estos aspectos del cuadro de diálogo:

- El cuadro de diálogo "Buscar" no es modal. Esto significa que puede dejarlo abierto mientras utiliza la vista Texto.
- Si tiene texto seleccionado antes de abrir el cuadro de diálogo, el texto seleccionado se inserta automáticamente en el campo del término de búsqueda.
- Para realizar búsquedas dentro de una selección: (i) marque la selección, (ii) active la opción **Buscar en la selección** 158 para bloquear la selección e (iii) introduzca el término de búsqueda. Para buscar dentro de otra selección, desbloquee la selección actual desactivando la opción **Buscar en la selección** 158, marque una nueva selección y active otra vez la opción **Buscar en la selección** 158.
- Después de cerrar el cuadro de diálogo "Buscar" puede repetir la búsqueda actual con solo pulsar **F3** (búsqueda hacia adelante) o **Mayús+F3** (búsqueda hacia atrás). El cuadro de diálogo aparece al pulsar estas teclas.

Opciones de búsqueda

Las opciones para determinar los criterios de búsqueda pueden configurarse con los botones situados debajo del campo del término de búsqueda (*imagen siguiente*). Si una opción está activada, su botón aparece en color azul (p. ej. en la imagen siguiente está activado el primer botón).



Estas son las opciones de búsqueda que se pueden seleccionar:

- **Coinc. mayús/min:** la búsqueda tiene en cuenta las mayúsculas y minúsculas a la hora de realizar la búsqueda (`Address` no es lo mismo que `address`).

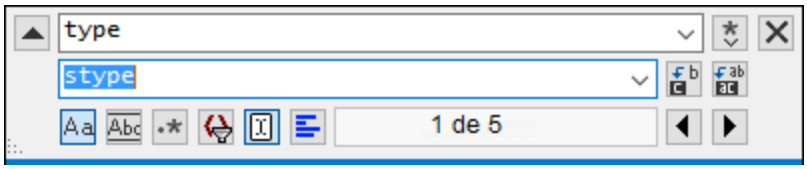
 © 2016-2022 Altova GmbH

- **Sólo palabras completas**: solo se consideran coincidencias las palabras completas. Por ejemplo, si activa esta opción y busca el término `fit`, la aplicación devuelve como resultado la palabra `fit`, pero no `fitness`.
- **Expresión regular:** si activa esta opción, el término de búsqueda se lee como expresión regular (*véase más abajo* 160).
- **Filtrar resultados:** seleccione en qué componentes del documento se debe ejecutar la búsqueda.
- **Buscar delimitador:** cuando se introduce un término de búsqueda, los resultados del documento se resaltan y uno de ellos se marcará como selección actual. Con la opción **Buscar delimitador** puede definir si esta primera selección se hace en relación a la posición actual del cursor o no. Es decir, si la opción **Buscar delimitador** está activada, entonces el primer resultado seleccionado será el siguiente resultado a partir de la posición actual del cursor. Por el contrario, si la opción **Buscar delimitador** está desactivada, el primer resultado seleccionado será el primer resultado del documento, empezando desde el principio.
- **Buscar en selección:** si activa esta opción, la selección actual se bloquea y la búsqueda se ejecuta en la selección solamente. De lo contrario, la búsqueda se ejecuta en todo el documento. Antes de realizar una selección nueva deberá desbloquear la selección actual desactivando el botón de la opción **Buscar en selección**.

Reemplazar texto

Para reemplazar una cadena de texto por otra:

1. Pulse **Ctrl+H** (o seleccione el comando de menú **Edición | Reemplazar**) para abrir el cuadro de diálogo "Reemplazar" (*imagen siguiente*). También puede cambiar al modo **Reemplazo** del cuadro de diálogo "Buscar" haciendo clic en el botón de flecha situado en la esquina superior izquierda del cuadro de diálogo.



2. Introduzca la cadena de texto que desea reemplazar en el campo de búsqueda y en el campo de reemplazo introduzca la cadena de texto nueva. El número de coincidencias que se reemplazarán y el índice de la coincidencia seleccionada aparecen justo debajo del campo de reemplazo. Además, la posición de las coincidencias aparecen marcadas en beige en la barra de desplazamiento (véase *Buscar texto en un documento* 157). En la imagen anterior, por ejemplo, la búsqueda de la cadena `type` devolvió 5 coincidencias y la coincidencia seleccionada es la tercera.
3. Los botones **Reemplazar siguiente** y **Reemplazar todos** están situados a la derecha del campo de la cadena de reemplazo. Si hace clic en el botón **Reemplazar siguiente** pueden ocurrir dos cosas: (i) si el cursor está situado junto a una coincidencia o dentro de una, se reemplaza esa coincidencia o (ii) si el cursor está situado fuera de una coincidencia, la función de reemplazo salta a la coincidencia siguiente. Haga clic en **Reemplazar siguiente** para reemplazar esta coincidencia. Haga clic en el botón **Reemplazar todos** para reemplazar todas las coincidencias.

Debe tener en cuenta estos aspectos de la función de reemplazo:

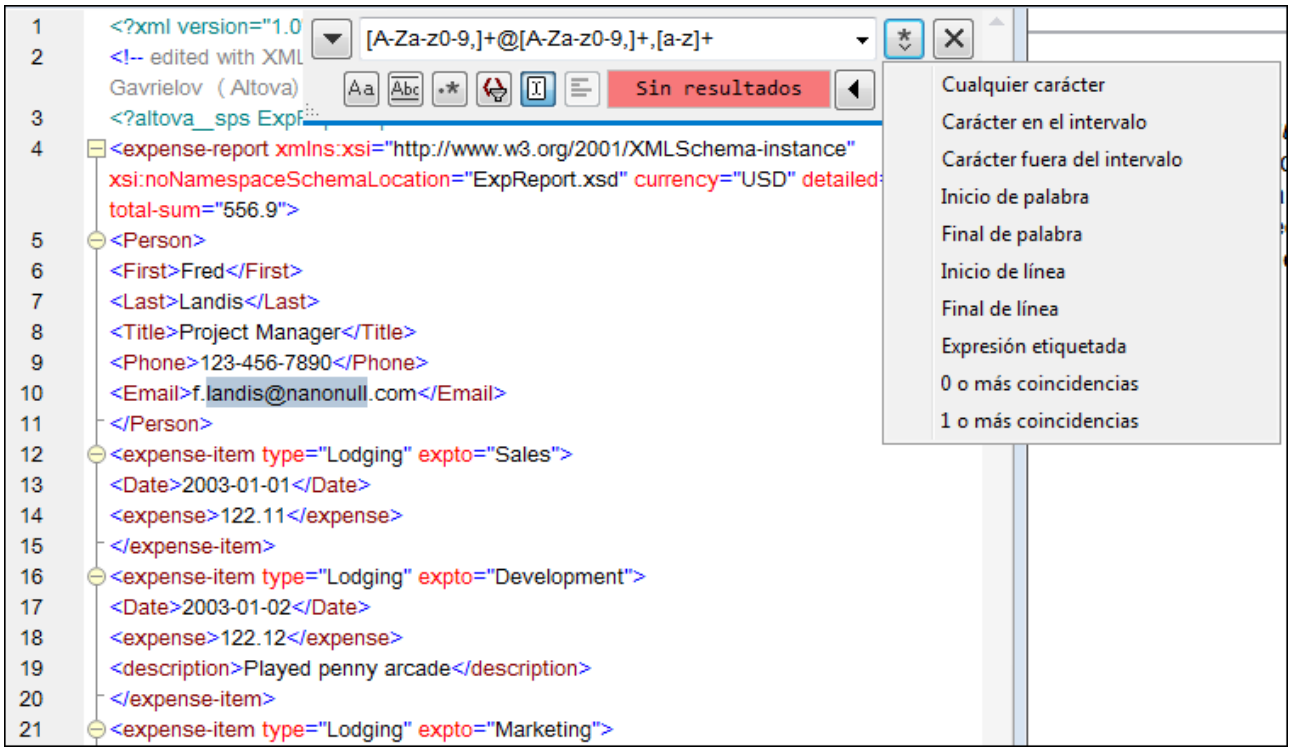
- Si desea reemplazar texto situado dentro de una selección (y no en todo el documento): (i) marque la selección, (ii) active la opción **Buscar en la selección** 158 para bloquear la selección, (iii) introduzca el término de búsqueda y reemplazo y (iv) haga clic en **Reemplazar siguiente** o **Reemplazar todos** según corresponda. Para reemplazar texto dentro de otra selección, desbloquee la selección actual

desactivando la opción **Buscar en la selección** [158], marque una nueva selección y active otra vez la opción **Buscar en la selección** [158].

- Para deshacer una acción de reemplazo pulse **Ctrl+Z** o seleccione el comando **Edición | Deshacer**.

Uso de expresiones regulares

Puede usar expresiones regulares para buscar cadenas de texto en el documento. Para ello lo primero es activar la opción *Expresión regular* (*véase más arriba* [158]). Al activar esta opción estamos especificando que el texto del campo del término de búsqueda debe evaluarse como expresión regular. El segundo paso consiste en introducir la expresión regular en el campo del término de búsqueda. Si necesita ayuda para construir su expresión regular, haga clic en el botón **Generador de expresiones regulares** (situado a la derecha del campo del término de búsqueda, tal y como muestra la imagen siguiente). Seleccione un elemento de la lista desplegable para introducir los caracteres correspondientes en el campo de búsqueda. Por ejemplo, en la imagen siguiente puede ver una expresión regular sencilla que se utiliza para buscar direcciones de correo electrónico. Para más información sobre los caracteres de las expresiones regulares consulte la tabla que aparece más abajo.



Metacaracteres de las expresiones regulares

A continuación puede consultar una lista de los metacaracteres de las expresiones regulares.

.	Cualquier carácter. Es un comodín para un solo carácter.
(	El inicio de una región en la que se realizará la búsqueda.
)	El final de la región en la que se realizará la búsqueda.
(abc)	Los metacaracteres (y) marcan el inicio y el final de una expresión etiquetada. Las expresiones etiquetadas son ideales si necesita etiquetar (es decir, recordar) una región

© 2016-2022 Altova GmbH

	encontrada en la búsqueda para poder hacer referencia a ella más tarde (referencia inversa). Puede etiquetar un máximo de nueve expresiones (y hacerles referencia más tarde, desde el campo *Buscar* o *Reemplazar*). Por ejemplo, `(the) \1` encuentra la cadena `the the`. Esta expresión significa: encontrar la cadena "the" (y recordarla como región etiquetada), seguido de un carácter de espacio y de una referencia inversa a la región etiquetada encontrada previamente.
`\n`	En una expresión de búsqueda o reemplazo, indica el texto coincidente con la n-ésima expresión etiquetada, siendo n un número del 1 al 9. Por ejemplo, si el término de búsqueda es `Fred([1-9])XXX` y el término de reemplazo es `Sam\1YYY`, el texto `Fred2XXX` se reemplazará con `Sam2YYY`.
`\<`	Inicio de palabra.
`\>`	Final de palabra.
`\x`	Encuentra el carácter `x` que sigue a la barra inversa, cuando el carácter `x` es un carácter que se utiliza en la sintaxis de expresiones regulares. Por ejemplo, `\[` encuentra el carácter `[` y no el inicio de un conjunto de caracteres.
`[...]`	Conjunto de caracteres. Por ejemplo, `[abc]` encuentra los caracteres `a`, `b` o `c`. También puede usar intervalos como `[a-z]` para buscar cualquier carácter en minúsculas.
`[^...]`	Un carácter cualquiera no perteneciente al conjunto. Por ejemplo, `[^A-Za-z]` encuentra cualquier carácter excepto caracteres alfabéticos en mayúsculas o minúsculas.
`^`	Inicio de línea (a no ser que se use dentro de un conjunto de caracteres, ver fila anterior).
`$`	Final de línea. Por ejemplo, `A+$` encuentra una A o más de una A que estén al final de una línea.
`*`	0 veces o más. Por ejemplo, `Sa*m` encuentra `Sm`, `Sam`, `Saam`, `Saaam`, etc.
`+`	1 o más veces. Por ejemplo `Sa+m` encuentra `Sam`, `Saam`, `Saaam`, etc

Representación de caracteres especiales
Estas expresiones regulares son especiales:

\r	Retorno de carro (CR). Puede usar CR (**\r**) o LF (**\n**) para buscar o crear una línea nueva.
\n	Avance de línea (LF). Puede usar CR (**\r**) o LF (**\n**) para buscar o crear una línea nueva.
\t	Tabulación
****	Utilice esto para aplicar escape a los caracteres que aparecen en las expresiones regulares. Por ejemplo: **\\\n**

Ejemplos de expresiones regulares

En este apartado explicamos cómo usar expresiones regulares para encontrar y reemplazar texto. En muchos casos la búsqueda y reemplazo de texto es una tarea sencilla que no necesita el uso de expresiones regulares. Sin embargo, a veces una operación de búsqueda y reemplazo no es suficiente y será necesario manipular el texto. Por ejemplo, imagine que tiene un archivo XML que incluye varios miles de líneas y necesita cambiar el nombre de determinados elementos de una sola vez, pero sin cambiar el contenido de los elementos. O, por ejemplo, imagine que necesita cambiar el orden de varios atributos de un elemento. En estos dos casos el uso de expresiones regulares le permitiría evitar mucho trabajo manual.

Ejemplo nº1: cambiar el nombre de elementos

El fragmento de código XML que aparece a continuación incluye una lista de libros. Imaginemos que nuestro objetivo es reemplazar el elemento `<Category>` de cada libro con el elemento `<Genre>`. Podemos hacerlo usando expresiones regulares.

```
<?xml version="1.0" encoding="UTF-8"?>
<books xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="books.xsd">
   <book id="1">
      <author>Mark Twain</author>
      <title>The Adventures of Tom Sawyer</title>
      <category>Fiction</category>
      <year>1876</year>
   </book>
   <book id="2">
      <author>Franz Kafka</author>
      <title>The Metamorphosis</title>
      <category>Fiction</category>
      <year>1912</year>
   </book>
   <book id="3">
      <author>Herman Melville</author>
      <title>Moby Dick</title>
      <category>Fiction</category>
      <year>1851</year>
   </book>
</books>
```

Estos son los pasos que debe seguir:

1. Pulse **Ctrl+H** para abrir el cuadro de diálogo "Buscar y reemplazar".
2. Haga clic en el botón **Usar expresiones regulares** .
3. En el campo *Buscar* introduzca este texto: `<category>(.+)</category>`. Esta expresión regular encuentra todos los elementos `category`, que aparecerán resaltados.

© 2016-2022 Altova GmbH



Para encontrar el contenido de cada elemento (que desconocemos), usamos la expresión etiquetada `(.+)` . La expresión `(.+)` indica que se deben "buscar una instancia o varias de cualquier carácter (`.+`) y recordar este resultado". Como podrá ver en el paso siguiente, más adelante necesitaremos la referencia a la expresión etiquetada.

4. En el campo *Reemplazar* introduzca este texto: `<genre>\1</genre>` . Esta expresión regular define el texto de reemplazo. Observe que usa `\1` para hacer referencia inversa a la expresión etiquetada del campo *Buscar*. En otras palabras, en este contexto `\1` significa "el contenido del elemento `<category>` actual que coincida con la búsqueda".
5. Haga clic en **Reemplazar todos** y observe el resultado. Todos los elementos `category` se reemplazaron con `genre`, tal y como era nuestra intención.

Ejemplo nº2: cambiar el orden de los atributos

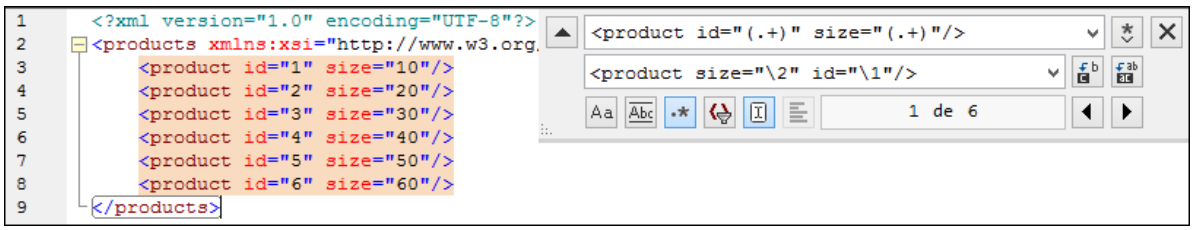
El fragmento de código XML que aparece a continuación incluye una lista de producto. Cada producto tiene dos atributos llamados `id` y `size`. Imaginemos que nuestro objetivo es cambiar el orden de los atributos `id` y `size` de cada elemento `product` (es decir, en todos los elementos `product` el atributo `size` debe aparecer antes de `id`). Podemos hacerlo usando expresiones regulares.

```
<?xml version="1.0" encoding="UTF-8"?>
<products xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="products.xsd">
   <product id="1" size="10"/>
   <product id="2" size="20"/>
   <product id="3" size="30"/>
   <product id="4" size="40"/>
   <product id="5" size="50"/>
   <product id="6" size="60"/>
</products>
```

Estos son los pasos que debe seguir:

1. Pulse **Ctrl+H** para abrir el cuadro de diálogo "Buscar y reemplazar".
2. Haga clic en el botón **Usar expresiones regulares** .

3. En el campo *Buscar* introduzca este texto: `<product id="(.+)" size="(.+)"/>` . Esta expresión regular busca un elemento `product` en el documento XML. Observe que, para encontrar el valor de cada atributo (que desconocemos) se usa la expresión etiquetada `(.+)` dos veces. La expresión etiquetada `(.+)` encuentra el valor de cada atributo (que se supone será una instancia o varias de cualquier carácter, es decir `.+`).
4. En el campo *Reemplazar* introduzca este texto: `<product size="\2" id="\1"/>` . Esta expresión regular contiene el texto de reemplazo para cada uno de los elementos `product` que se encontraron. Observe que usa las referencias `\1` y `\2`, que corresponden a las expresiones etiquetadas del campo *Buscar*. En otras palabras, `\1` significa "el valor del atributo `id`" y `\2` significa "el valor del atributo `size`".

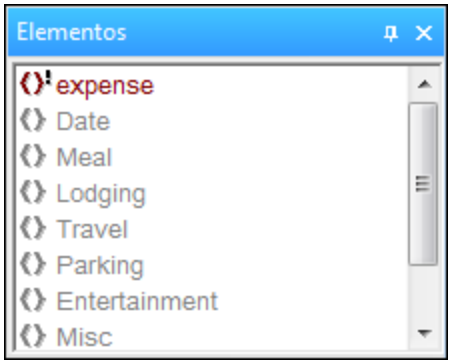


5. Haga clic en **Reemplazar todos** y observe el resultado. Todos los elementos `product` del documento XML se reemplazaron de modo que el atributo `size` aparezca delante del atributo `id`.

4.2.6 Ayudantes de entrada de la vista Texto

Los ayudantes de entrada disponibles en la vista Texto dependerán del tipo de documento que se esté editando. A continuación encontrará una lista con los tipos de documentos más comunes y sus ayudantes de entrada. El uso general de los ayudantes de entrada se describe en este apartado 165. Las funciones especiales propias de cada tipo de documento, si las hubiese, se describen en las secciones dedicadas a cada tipo de documento.

- *XML:* Elementos (*ver imagen siguiente*), Atributos, Entidades



- *HTML:* Elementos, Atributos, Entidades
- *CSS:* Esquema CSS Outline, Propiedades CSS, Elementos HTML
- *DTD:* no tiene ayudantes de entrada propios
- *XQuery:* Palabras clave XQuery, Variables XQuery, Funciones XQuery
- *WSDL:* Vista general, Detalles
- *Texto:* Entidades

 © 2016-2022 Altova GmbH

Observe que varios tipos de documentos, como XSD, XSLT, XHTML y RDF, son básicamente documentos XML y, por tanto, todos ellos contarán también con los ayudantes de entrada Elementos, Atributos y Entidades.

Visualización y uso de las opciones de los elementos de entrada

Las opciones de los distintos ayudantes de entrada tienen colores distintos. Estos códigos de colores se explican en detalle en la documentación de los ayudantes de entrada de los respectivos tipos de documentos. En general, podemos destacar estos aspectos de los ayudantes de entrada:

- Los ayudantes de entrada son contextuales y muestran sólo las opciones que se pueden insertar en cada momento.
- Si la opción de entrada ya se insertó en la posición seleccionada (o en otra posición equivalente y válida) y no se puede insertar por segunda vez en esa posición (por ejemplo, un atributo XML), entonces la opción de entrada aparece en color gris.
- Si la opción de entrada es obligatoria, junto a ella aparece un signo de exclamación.
- Para insertar una opción del ayudante de entrada en la posición del cursor, haga doble clic en la opción.
- Cuando se inserta un elemento desde el ayudante de entrada Elementos, su etiqueta inicial y su etiqueta final se insertan en el texto. Los elementos obligatorios también se insertan, siempre y cuando activara esa opción de configuración en el diálogo "Opciones" (**Herramientas | Opciones | Edición**).
- Cuando se inserta un atributo desde el ayudante de entrada Atributos, se inserta en la posición del cursor e incluye el signo igual (=) y comillas (' ') para delimitar el valor del atributo. El cursor se sitúa entre las comillas para que pueda empezar a escribir el valor del atributo inmediatamente.

Nota: cuando esté trabajando con archivos de gran tamaño, puede deshabilitar la función de finalización automática y los ayudantes de entrada para que la carga y edición de estos archivos sea más rápida. En estos casos el usuario puede indicar el tamaño de archivo máximo. Para más información, consulte la sección Opciones | Edición[1590] de la *Comandos de menú*.

4.2.7 Vista dividida

Con esta opción puede dividir la ventana principal de la vista Texto en dos y mostrar en ambas el documento activo. Puede navegar de forma independiente en cada una de las ventanas, lo que permite consultar dos partes distintas del mismo documento una junto a la otra. Los cambios tienen efecto en el documento base y se reflejan en ambas vistas.

Alternar entre la Vista dividida y la Vista única

Para crear varias vistas del **documento activo**:

- *División horizontal:* arrastre hacia abajo el icono de división horizontal que se encuentra en la esquina superior derecha (resaltado con un círculo verde en la imagen siguiente).
- *División vertical:* arrastre hacia la derecha el icono de división vertical que se encuentra en la esquina inferior izquierda (resaltado con un círculo azul en la imagen siguiente).



Para volver a tener una única vista puede:

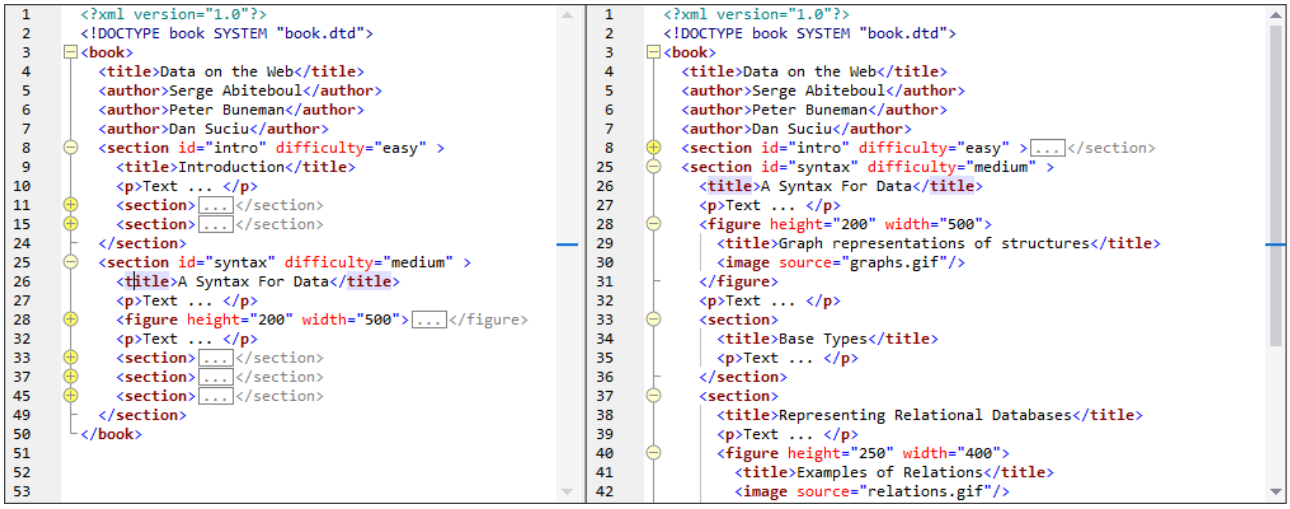
- hacer doble clic en la barra separadora o
- mover la barra separadora hasta uno de los bordes de la ventana principal que sean paralelos a ella.

Nota: las vistas divididas se crean de forma individual para cada documento.

Editar y navegar por documentos en la Vista dividida

La mayor ventaja de trabajar con la Vista dividida es que se pueden ver partes distintas de un mismo documento una junto a la otra, lo cual es muy útil cuando se trabaja con documentos largos. Esta función también permite aplicar cambios al documento desde las dos vistas abiertas del mismo documento. En la imagen siguiente se muestra la Vista dividida en vertical de un documento.

© 2016-2022 Altova GmbH



Tenga en cuenta que:

- todas las opciones de visualización[149] (como la numeración de líneas o el plegamiento de código fuente), opciones de edición[151], opciones de navegación[154], etc. que están disponibles en la Vista única de un documento también están disponibles en las dos vistas de la Vista dividida (*imagen anterior*).
- puede navegar por separado en cada una de las vistas de la Vista dividida.
- puede usar plegamiento de código de forma separada en cada una de las vistas.
- todas las acciones de edición, incluidas las de los ayudantes de entrada, se reflejan en ambas vistas.

4.2.8 Teclas de acceso rápido

A continuación puede ver las teclas de acceso rápido predeterminadas para los comandos más frecuentes de la vista Texto. Puede cambiar estas teclas de acceso rápido predeterminadas en la pestaña *Teclado*[1569] del cuadro de diálogo "Personalizar".

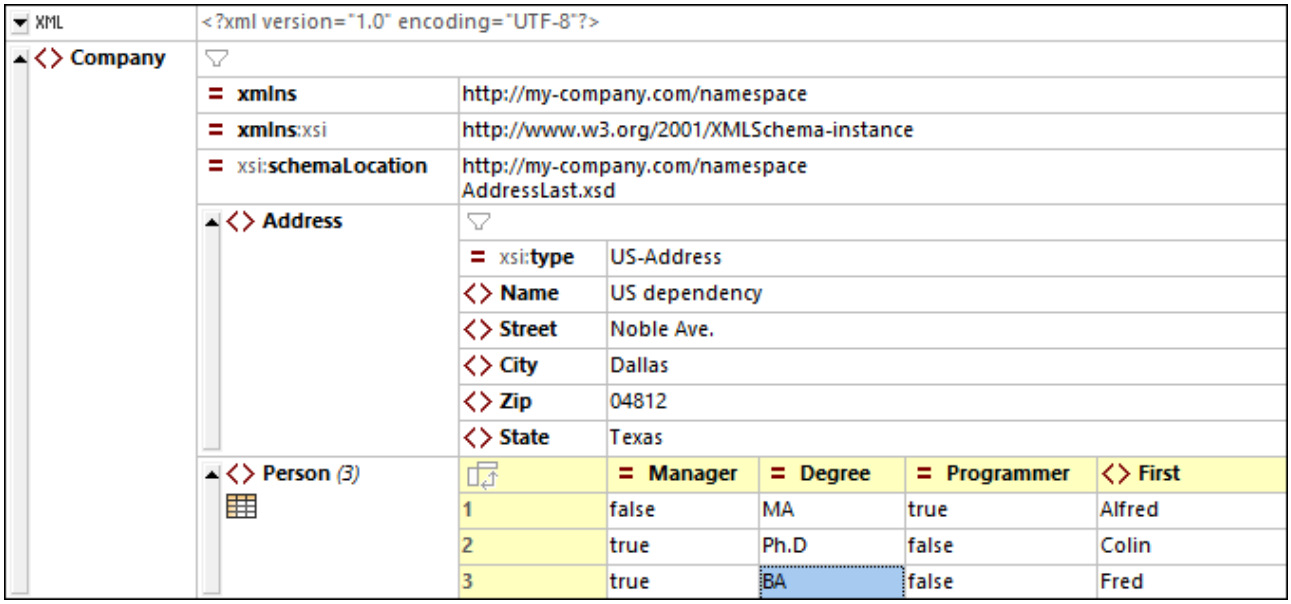
Comandos de la vista Texto

CTRL + E	Pasar de la etiqueta de inicio a la etiqueta final y viceversa
CTRL + Mayús + E	Seleccionar elemento que contiene el cursor
CTRL + Alt + E	Ir al elemento primario
CTRL + "+"	Acercarse
CTRL + "-"	Alejarse
CTRL + 0	Restaurar nivel de zoom
CTRL + rueda de desplazamiento hacia adelante	Acercarse
CTRL + rueda de desplazamiento hacia atrás	Alejarse

© 2016-2022 Altova GmbH

4.3 Vista Cuadrícula

La Vista Cuadrícula se puede usar para documentos XML, JSON y DTD. (*La imagen siguiente es de la vista Cuadrícula de un documento XML.*) En ella se puede ver la estructura jerárquica de documentos XML y DTD mediante contenedores anidados, que se pueden expandir y contraer. Esta vista permite entender la estructura del documento. En la vista Cuadrícula tanto el contenido como la estructura del documento se puede manipular con total facilidad.

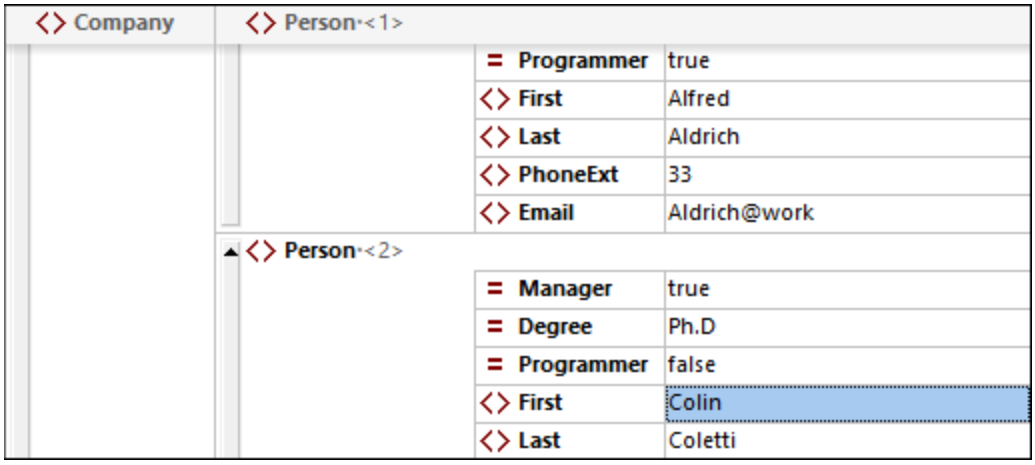


Para más detalles sobre cómo trabajar con la vista Cuadrícula consulte:

- Datos XML en la vista Cuadrícula 293
- Vista Cuadrícula JSON 679
- Documentos DTD 442

Encabezados anclados

Al desplazarse hacia abajo en un documento XML, JSON o DTD en la vista Cuadrícula, el nodo de nivel superior se puede seguir viendo en la barra de encabezados que aparece en la parte superior. Conforme siga bajando en la barra de encabezados van apareciendo más elementos antecesores del nodo que esté en ese momento en la parte superior de la ventana. Por ejemplo, en la imagen siguiente en la barra aparecen los antecesores del nodo `/Company/Person[1]/@Programmer`.



Si hace clic en cualquiera de los nodos de la barra de encabezados, ese nodo pasa a estar seleccionado y aparece en la ventana de visualización.

Personalizar la vista Cuadrícula

- Para cambiar el tamaño de una columna, coloque el cursor sobre el borde de la columna y arrástrelo hasta que alcance el ancho deseado.
- Para ajustar el tamaño de una columna al contenido de su entrada de mayor longitud, haga doble clic en el borde derecho de la columna.
- Para ajustar el ancho de las columnas y poder visualizar todo el contenido, seleccione el comando Vista | Ancho de tabla óptimo 1488 o haga clic en el icono "Ancho de tabla óptimo".
- El alto de las celdas viene dado por su contenido, pero también se puede ajustar. Seleccione Herramientas | Opciones | Vista | Vista Cuadrícula mejorada, "Limitar alto de celda a xx líneas".

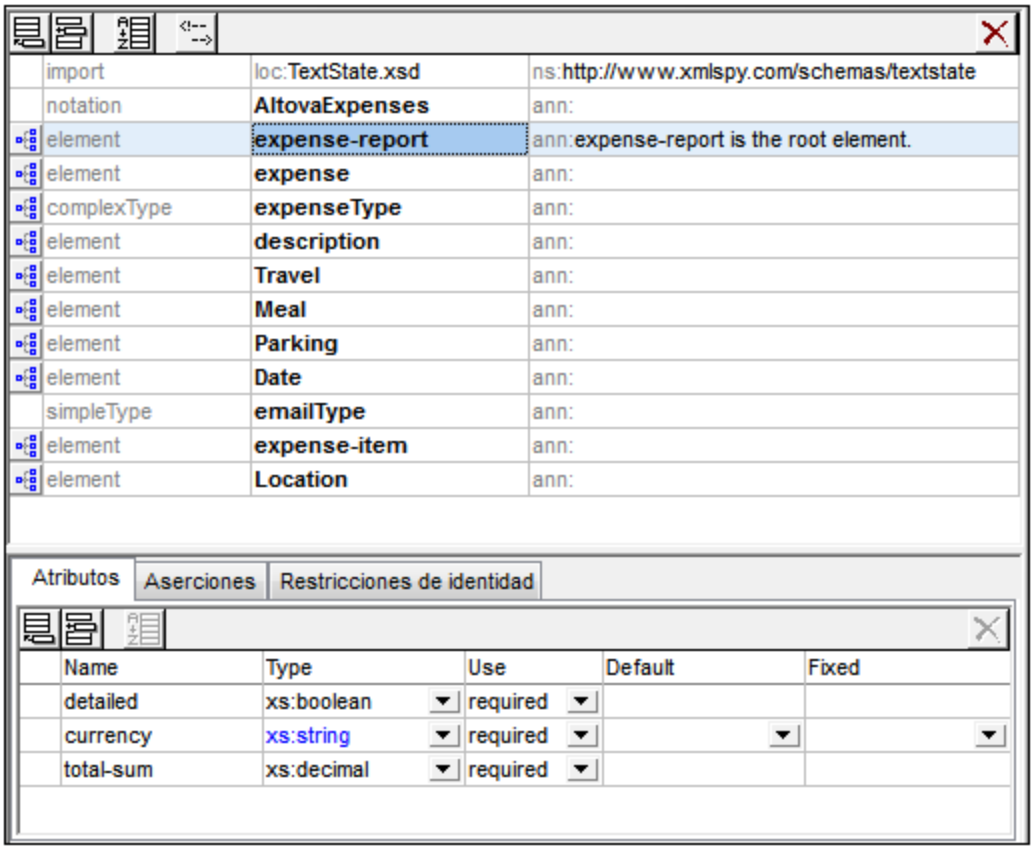
Nota: si marca datos en la vista Cuadrícula y después cambia a la vista Texto, los datos también estarán marcados en la vista Texto.

© 2016-2022 Altova GmbH

4.4 Vista Esquema

Sitio web de Altova: Editor de esquema XML

En la vista Esquema (*imagen siguiente*) puede ver y editar esquemas XML de forma gráfica. La interfaz gráfica de esta vista permite crear esquemas con rapidez y precisión. La vista Esquema se divide en dos paneles: (i) el panel superior, donde puede diseñar las relaciones estructurales entre los componentes del esquema y (ii) el panel inferior, donde puede definir el componente seleccionado en el panel superior. La vista Esquema también ofrece tres ayudantes de entrada que le ayudarán a crear esquemas válidos: Componentes, Detalles y Facetas.



Panel superior: diseño de esquemas

En el panel superior de la vista Esquema puede alternar entre dos vistas:

- Vista general del esquema 177: muestra todos los componentes globales del esquema (como elementos y tipos complejos globales) en una sencilla lista en forma de tabla (*ver imagen* 177). Si hace clic en el icono de un componente global de la lista, se abre la vista del modelo de contenido del componente seleccionado. Recuerde que no todos los componentes globales pueden tener un modelo de contenido (por ejemplo, los tipos simples).

- Vista del modelo de contenido[190]: muestra el modelo de contenido del componente global seleccionado (*ver imagen*[190]). Para volver a la vista general del esquema, haga clic en el icono **Mostrar nodos globales** situado en la esquina superior izquierda del panel.



Cambiar a la vista del modelo de contenido: disponible solamente para los componentes globales que tienen un modelo de contenido. Abre el modelo de contenido del componente global en la vista del modelo de contenido[190].



Mostrar nodos globales: disponible en la vista del modelo de contenido. Abre la vista general del esquema[177].

Panel inferior: atributos, aserciones y restricciones de identidad

El panel inferior de la vista Esquema (*ver imagen*[212]) incluye varias pestañas para definir atributos, aserciones y restricciones de identidad[212] para el componente seleccionado en el diseño (es decir, en el panel superior). En adelante, nos referiremos a este panel como "panel AARID".

- En el modo XSD 1.0, el panel inferior tiene dos pestañas: (i) Atributos[213] y (ii) Restricciones de identidad[220].
- En el modo XSD 1.1, el panel inferior tiene tres pestañas: (i) Atributos[213], (ii) Aserciones[216] y (iii) Restricciones de identidad[220].

El panel AARID aparece en la vista general del esquema siempre y en la vista del modelo de contenido a veces. En la vista del modelo de contenido puede ver los tres tipos de definiciones (atributos, aserciones y restricciones de identidad), bien en el diagrama del modelo, bien en el panel AARID. Para ello, active o desactive los botones pertinentes de la barra de herramientas Diseño de esquemas: (i) **Mostrar atributos en diagrama**, (ii) **Mostrar aserciones en diagrama** o (iii) **Mostrar restricciones de identidad en diagrama**. También puede configurar la presentación de las definiciones en el cuadro de diálogo "Configurar visualización del esquema" (**Diseño de esquemas | Configurar opciones de vista**[1378]). Si todos los tipos de definición del panel AARID se muestran en el diagrama del modelo de contenido, entonces desaparace el panel inferior de la vista.

Configuración del esquema

El cuadro de diálogo "Configuración del esquema" (**Diseño de esquemas | Configuración del esquema**[1370]) sirve para definir la configuración global del esquema activo. Las opciones de configuración equivalen a los atributos del elemento `xs:schema`.

Organización de esta sección del manual

Esta sección se divide en los siguiente apartados:

- Modo XSD: XSD 1.0 o 1.1[173]: seleccione el modo de edición.
- Vista general del esquema[177]: edite las propiedades de los componentes globales.
- Vista del modelo de contenido[190]: edite el modelo de contenido de cada componente global.
- Atributos, aserciones y restricciones de identidad[212]: defina estas propiedades de cada componente.
- Ayudantes de entrada[228]: use los ayudantes para definir las propiedades de los componentes rápidamente.
- Restricciones inteligentes[244]: cree y edite tipos derivados a partir de tipos base de forma gráfica.
- Uso de atributos con prefijo xml:[248] añada los atributos `base`, `id`, `lang` y `space` a los componentes del esquema de forma gráfica.
- Adelante y atrás: navegar de una posición a otra[250]: use esta función de la vista Esquema para visitar posiciones del esquema visitadas con anterioridad.

© 2016-2022 Altova GmbH

Conectarse a SchemaAgent

Desde XMLSpy también se puede conectar a SchemaAgent, visualizar los componentes de otros esquemas en la IGU y utilizar esos componentes en el esquema que está editando. Para más información sobre cómo trabajar con SchemaAgent en XMLSpy consulte la sección Trabajar con SchemaAgent 463.

Búsqueda en esquemas

La función de búsqueda en esquemas permite realizar búsquedas inteligentes en esquemas, es decir, búsquedas que están limitadas por ciertos criterios relacionados con el esquema. Por ejemplo, puede restringir la búsqueda a un tipo de componente concreto. La búsqueda en esquemas se describe en detalle en la sección DTD y esquemas XML 475.

4.4.1 Modo XSD: XSD 1.0 o XSD 1.1

En la vista Esquema puede seleccionar si el esquema XML (XSD) se edita y valida de acuerdo con la especificación XML Schema 1.0 (XSD 1.0) o de acuerdo con la especificación XML Schema 1.1 (XSD 1.1). El **modo XSD** que se utiliza para editar un archivo depende de dos configuraciones: la de la aplicación y la del documento XSD.

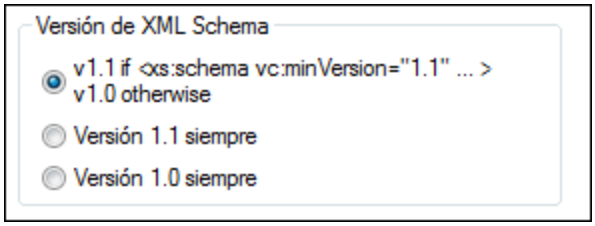
Selección del modo XSD

El modo XSD elegido define qué funciones de edición y validación (XSD 1.0 o 1.1) están disponibles en la vista Esquema del documento activo. Tiene dos opciones:

- seleccionar una configuración para toda la aplicación, en cuyo caso todos los documentos XSD que se abran en la vista Esquema se editan en el modo seleccionado, o
- guardar el número de versión XSD del documento XSD y dejar que la aplicación seleccione automáticamente el modo XSD en base a esta información.

Modo XSD para toda la aplicación

Puede configurar el modo XSD para toda la aplicación en la sección *Archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**). Si selecciona la opción *Versión 1.1 siempre* o la opción *Versión 1.0 siempre*, el modo seleccionado es el modo general de la aplicación. Todos los esquemas XML que se abran en la vista Esquema se editarán en el modo seleccionado. Por el contrario, si selecciona la opción *Versión 1.1 si <xs:schema vc:minVersion="1.1"...>...*, la elección del modo dependerá de la información disponible en el esquema XML que esté abierto y no en el modo general de la aplicación. Siga leyendo para obtener más información.



Puede alternar el modo general de la aplicación (*Versión 1.1 siempre* o *Versión 1.0 siempre*) siempre que quiera en la sección *Archivo* del cuadro de diálogo "Opciones" (*imagen anterior*).

Nota: si en esta sección selecciona un modo XSD para toda la aplicación y en la vista Esquema cambia de modo con el botón **XSD 1.0** o **XSD 1.1** de la barra de herramientas (*ver más abajo*), el modo cambia provisionalmente. Cuando vuelva a cargar el documento, la aplicación volverá al modo XSD seleccionado para toda la aplicación. El documento se vuelve a cargar cada vez que se cambia de vista o cuando se actualiza la vista Esquema (**Archivo | Volver a cargar**).

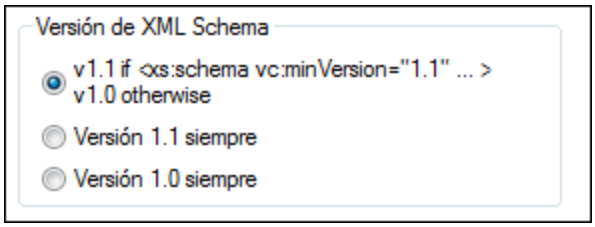
Modo XSD para el documento



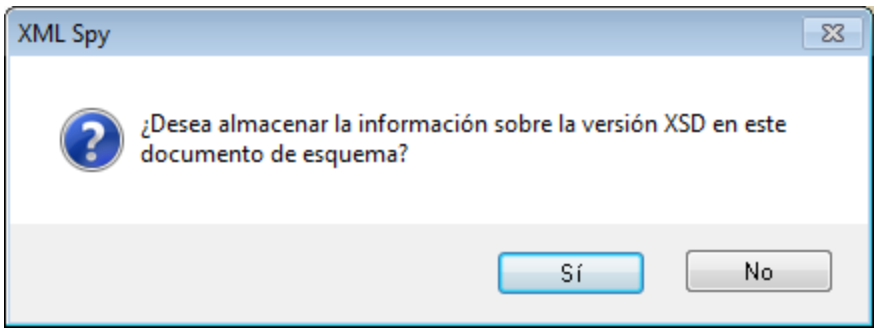
Nota: si está presente, el atributo `vc:minVersion` debe estar en el espacio de nombres `http://www.w3.org/2007/XMLSchema-versioning`. En este caso, el documento de esquema XML debe tener una declaración de espacio de nombres que enlace el prefijo de espacio de nombres `vc:` con este espacio de nombres. Si usa el botón **XSD 1.1** de la barra de herramientas (imagen anterior), el espacio de nombres se añade automáticamente. Si hace clic en el botón **XSD 1.0**, esta declaración de espacio de nombres se elimina si no hay ningún nombre de nodo del documento en el espacio de nombres.

Para activar el modo XSD para todo el documento y especificar el modo XSD de un documento:

1. *Active el modo XSD para todo el documento:* en la sección *Archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**), elija la opción *Versión 1.1 si <xs:schema vc:minVersion="1.1"...>. De lo contrario, versión 1.0 (imagen siguiente)*. Esto indica a XMLSpy que el modo XSD de la vista Esquema se establece en función del atributo `vc:minVersion` del elemento `xs:schema`.



2. *Especifique la versión XSD del documento:* en la barra de herramientas Diseño de esquemas (*imagen anterior*) de la vista Esquema haga clic en el botón **XSD 1.0** o **XSD 1.1**. Aparece un cuadro de diálogo de confirmación (*imagen siguiente*).



 © 2016-2022 Altova GmbH

3. Esto es lo que ocurre cuando se hace clic en **Sí**: (i) el valor correspondiente se inserta en el atributo `vc:minVersion` del elemento `xs:schema` y (ii) si seleccionó **XSD 1.1**, el espacio de nombres `XMLSchema-versioning` se declara con un enlace al prefijo de espacio de nombres `vc:`. Si seleccionó **XSD 1.0**, la declaración de espacio de nombres se elimina si no queda ningún nodo en el espacio de nombres `XMLSchema-versioning`. Ahora el esquema XML contiene el número de versión XSD. Al guardar el archivo, el modo XSD se guarda en él. Cuando vuelva a abrir o cargar el archivo, la vista Esquema cambia automáticamente al modo XSD correspondiente al número de versión del atributo `vc:minVersion` del elemento `xs:schema`.

Nota: si eligió un modo XSD para todo el documento y el documento XSD no tiene el atributo `vc:minVersion` o el valor del atributo `vc:minVersion` no es `1.0` ni `1.1`, entonces la vista Esquema usa por defecto el modo XSD 1.0.

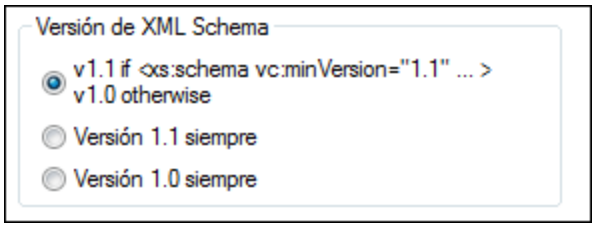
Nota: no se debe confundir el atributo `vc:minVersion` con el atributo `xsd:version`. El primero almacena el número de versión XSD, mientras que el segundo almacena el número de versión del documento.

Modo XSD de documentos nuevos

Cuando cree un documento XSD nuevo se le preguntará si desea crearlo en la versión XSD 1.0 o XSD 1.1. Si selecciona XSD 1.1, el documento nuevo se crea con el atributo `/xs:schema/@vc:minVersion="1.1"` y se declara el espacio de nombres `XMLSchema-versioning` con un enlace al prefijo de espacio de nombres `vc:`. Si selecciona If XSD 1.0, no se añade ni el atributo `vc:minVersion` ni la declaración de espacio de nombres `XMLSchema-versioning`. No obstante, el modo XSD habilitado en la vista Esquema depende una vez más de la versión XSD seleccionada en la sección *Archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**). Para más información siga leyendo con atención.

El modo XSD habilitado

El modo XSD habilitado en la vista **Esquema** depende tanto de (i) la presencia/ausencia del atributo `/xs:schema/vc:minVersion` del documento XSD (y, si está presente, de su valor), como de (ii) la opción seleccionada en la sección *Archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).



Versión de XML Schema	Atributo `xs:schema/@vc:minVersion`	Modo XSD
Versión 1.0 siempre	Ausente (o presente con cualquier valor)	1.0
Versión 1.1 siempre	Ausente (o presente con cualquier valor)	1.1
Valor de @vc:minVersion	`1.1`	1.1

Valor de @vc:minVersion	Falta el atributo o está presente y tiene cualquier valor menos 1.1	1.0

Nota: en las dos primeras situaciones de la tabla anterior, es posible abrir un esquema XSD 1.1 en modo XSD 1.0 y viceversa. Si existen incoherencias, XMLSpy se ocupará de ellas tal y como se describe más abajo.

Características del modo XSD

Las características de la interfaz y de edición de la vista Esquema cambian según el modo XSD activo (XSD 1.0 o XSD 1.1).

Si el modo activo es **XSD 1.0**:

- no hay funciones de edición compatibles con los nuevos componentes y propiedades de XML Schema 1.1. Sin embargo, si en el documento XSD ya existen componentes o propiedades XSD 1.1, se representan en la vista Esquema y se pueden eliminar, pero no editar.
- el documento se valida con la especificación XSD 1.0. Por tanto, si en el documento XSD ya existen componentes o propiedades XSD 1.1, se emite un error de validación.

Si el modo activo es **XSD 1.1**, las funciones de edición son totalmente compatibles con las características de XML Schema 1.1. El documento se valida con la especificación XML Schema 1.1.

Uso de las características XSD 1.1 en el modo XSD 1.0

A continuación explicamos qué ocurre cuando el documento incluye una característica XSD 1.1 que no es compatible con XSD 1.0 (una aserción, por ejemplo).

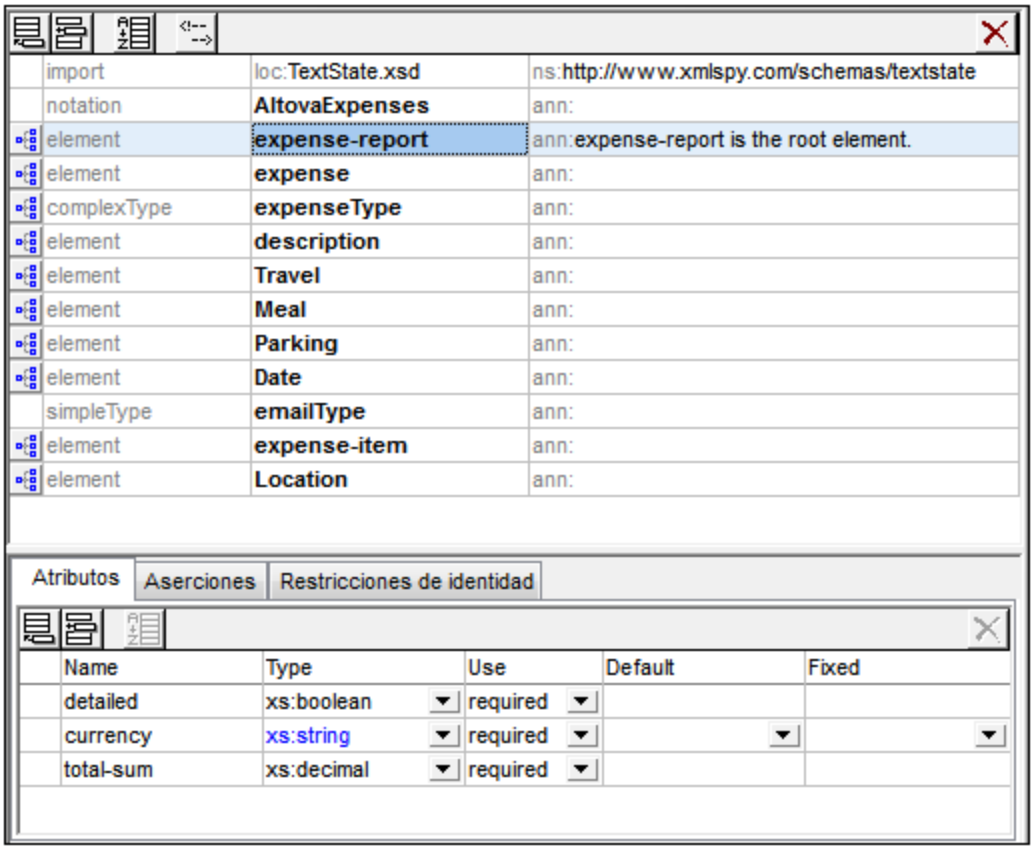
- ***Facetas assertion:*** si el tipo simple seleccionado tiene facetas `assertion`, la pestaña *Aserciones* aparece en el ayudante de entrada Facetas. No se puede editar, sólo eliminar.
- ***Elementos assert:*** el elemento `assert` aparece en el diagrama del tipo complejo. El panel AARID no tiene la pestaña *Aserciones*. No se puede añadir desde el menú contextual. No se puede editar, sólo eliminar.
- ***Atributos:*** si el esquema tiene atributos, entonces aparece la nueva propiedad `inheritable`. No se puede editar, sólo se puede seleccionar el valor vacío (lo cual equivale a eliminar las propiedades).
- ***Tipos complejos:*** si está presente la nueva propiedad `defaultAttributesApply`, esta aparece en el ayudante de entrada Detalles. No se puede editar, sólo se puede seleccionar el valor vacío (lo cual equivale a eliminar las propiedades).
- ***Documentación:*** los nuevos componentes y propiedades de XSD 1.1 no se incluyen en la documentación de la vista Esquema.
- ***Facetas:*** las facetas desconocidas dan lugar a errores de validación y aparecen en color rojo.
- ***Buscar en esquemas:*** durante la búsqueda los componentes y propiedades propios de XSD 1.1 se pasan por alto.
- ***Restricciones de identidad (IDC):*** la propiedad `isRef` aparece en caso de referencia y se puede desactivar. Se desactiva al modificar el nombre de la restricción de identidad.

© 2016-2022 Altova GmbH

- ***Grupos de sustitución múltiple:*** cuadro combinado para seleccionar un grupo de sustitución (en XSD 1.0 solamente se admite un grupo de sustitución).
- ***openContent:*** si existe en el esquema, aparece en el diagrama. No se puede añadir desde el menú contextual. No se puede editar, sólo eliminar. `defaultOpenContent` no aparece dentro de los tipos complejos.
- ***Elementos* override*:*** si están presentes, aparecen en la tabla de componentes globales. No se pueden añadir desde el menú contextual. No se pueden editar, sólo eliminar. Los elementos secundarios de `xs:override` se omiten y no se incluyen en el ayudante de entrada Componentes.
- ***Configuración del esquema:*** si están presentes, el cuadro de diálogo "Configuración del esquema" muestra las nuevas propiedades `defaultAttributes` y `xpathDefaultNamespace`. No se pueden editar, sólo se puede seleccionar el valor vacío (lo cual equivale a eliminar las propiedades).
- ***Tipos simples:*** los tipos desconocidos dan lugar a errores de validación y aparecen en color rojo.
- ***Alternativas de tipo:*** si están presentes, aparecen en el diagrama. No se pueden añadir desde el menú contextual. No se pueden editar, sólo eliminar.
- ***Comodines:*** si están presentes, aparecen las nuevas propiedades. No se puede editar, sólo se puede seleccionar el valor vacío (lo cual equivale a eliminar las propiedades).

4.4.2 Vista general del esquema

La vista general del esquema (*imagen siguiente*) enumera todos los componentes globales del esquema (elementos globales `import`, `element`, `complexType`, etc.).



En esta vista puede insertar, anexar o eliminar componentes globales, así como modificar sus propiedades. Para modificar las propiedades de un componente, seleccione un componente de la lista. Dependiendo del tipo de componente global elegido, sus propiedades se podrán editar en el ayudante de entrada Detalles[232], en el ayudante de entrada Facetas[234] o en el panel Atributos/Aserciones/Restricciones de identidad (AARID)[212].

Los componentes globales que pueden tener un modelo de contenido tienen a su izquierda un icono llamado **Cambiar a la vista del modelo de contenido**. Al hacer clic en este icono se abre la vista del modelo de contenido[190] del componente, donde puede editar su modelo de contenido.

Cambiar a la vista del modelo de contenido: disponible solamente para los componentes globales que tienen un modelo de contenido. Abre el modelo de contenido del componente global en la vista del modelo de contenido[190].

Mostrar nodos globales: disponible en la vista del modelo de contenido. Abre la vista general del esquema[177].

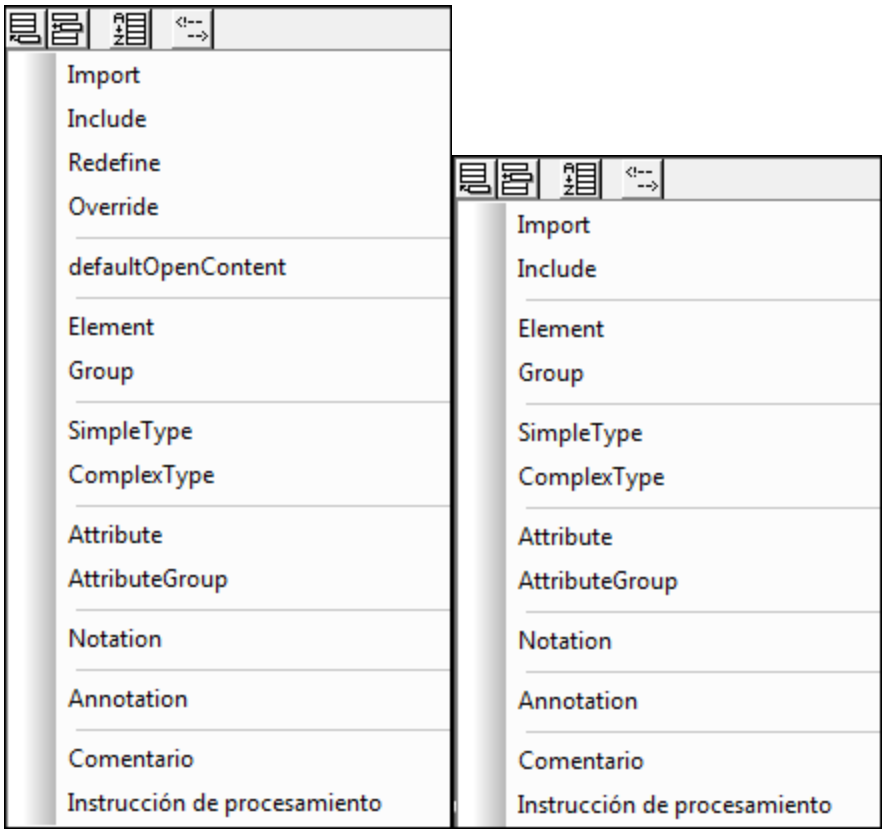
En este apartado describimos primero los mecanismos de la interfaz gráfica[179] de la vista general del esquema para después describir en detalle los diferentes componentes globales[182] del esquema.

© 2016-2022 Altova GmbH

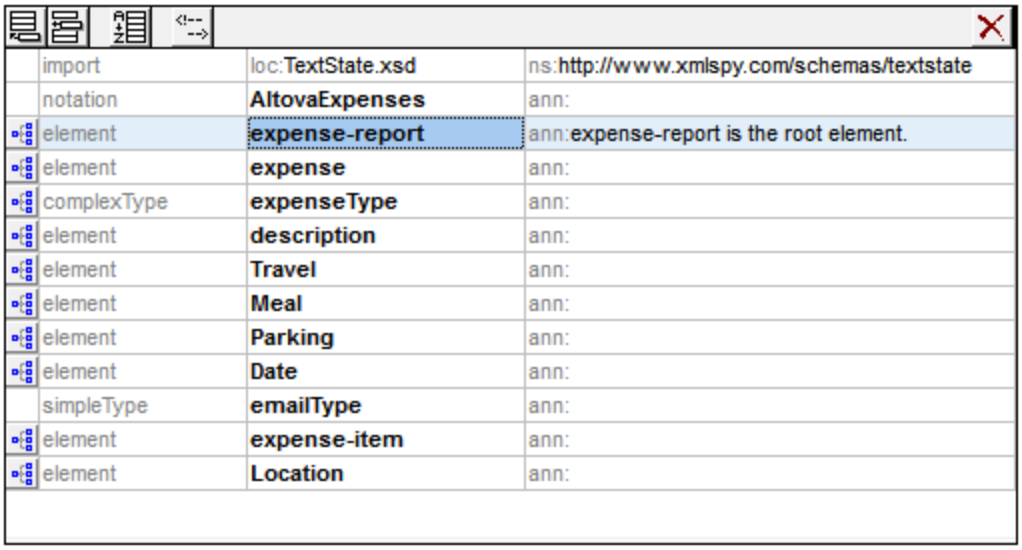
4.4.2.1 Mecanismos de la interfaz gráfica

Los componentes globales del esquema se añaden como elementos secundarios del elemento de nivel superior `xs:schema`. Para añadir un componente global, haga clic en los iconos **Anexar** o **Insertar** situados en la esquina superior izquierda del panel superior (*véase también la lista de iconos que aparece más abajo*) y después seleccione qué tipo de componente desea añadir (*imágenes siguientes*).

Las imágenes siguientes muestran los tipos de componentes globales que se pueden añadir en el modo XSD 1.1 (*a la izquierda*) y en el modo XSD 1.0 (*a la derecha*). Los elementos *override* y *defaultOpenContent* son características de la especificación XSD 1.1.



Puede añadir al esquema todos los componentes globales que desee. Los componentes globales del esquema se presentan en una lista en forma de tabla en la vista general del esquema (*imagen siguiente*).



	import	loc:TextState.xsd	ns:http://www.xmlspy.com/schemas/textstate
	notation	**AltovaExpenses**	ann:
	element	**expense-report**	ann:expense-report is the root element.
	element	**expense**	ann:
	complexType	**expenseType**	ann:
	element	**description**	ann:
	element	**Travel**	ann:
	element	**Meal**	ann:
	element	**Parking**	ann:
	element	**Date**	ann:
	simpleType	**emailType**	ann:
	element	**expense-item**	ann:
	element	**Location**	ann:

Edición en la vista general del esquema

En la vista general del esquema puede:

- Arrastrar los componentes y colocarlos en una posición nueva.
- Navegar por la lista usando las teclas de dirección y el tabulador.
- Usar operaciones de cortar/copiar y pegar para mover componentes globales, atributos, aserciones y restricciones de identidad a otra posición y de un esquema a otro.
- Hacer clic con el botón derecho en un componente para acceder a un menú contextual que incluye opciones para cortar, copiar, pegar, eliminar y editar los datos de anotación del componente.
- Introducir una línea nueva en los comentarios y en las anotaciones globales. Para ello pulse **Ctrl+Entrar**. Para introducir una tabulación, pulse **Ctrl+Tabulador**.

Iconos de la vista general del esquema

Anexar componente global: añade un componente global nuevo al final de la lista de componentes globales. Si por definición el componente debiera aparecer al principio del documento, entonces se añade al principio de la lista.

Insertar un componente global: añade un componente global nuevo encima del componente seleccionado. Si por definición el componente debiera aparecer al principio del documento, entonces se añade al principio de la lista.

Criterio de ordenación: abre el cuadro de diálogo "Ordenar componentes", donde puede configurar cómo se ordenan los componentes (según su nombre y después según la clase o viceversa), antes de realizar la ordenación (*consulte la descripción que aparece más abajo*).

Comentarios: abre un menú donde puede seleccionar si los comentarios globales se presentan en una sola línea o en varias (*consulte la descripción que aparece más abajo*).

Cambiar a la vista del modelo de contenido: disponible solamente para los componentes globales que tienen un modelo de contenido. Abre el modelo de contenido del componente global en la vista del modelo de contenido 190.

 © 2016-2022 Altova GmbH

Mostrar nodos globales: disponible en la vista del modelo de contenido. Abre la vista general del esquema 177.

Cambiar de la vista general del esquema a la vista del modelo de contenido y viceversa

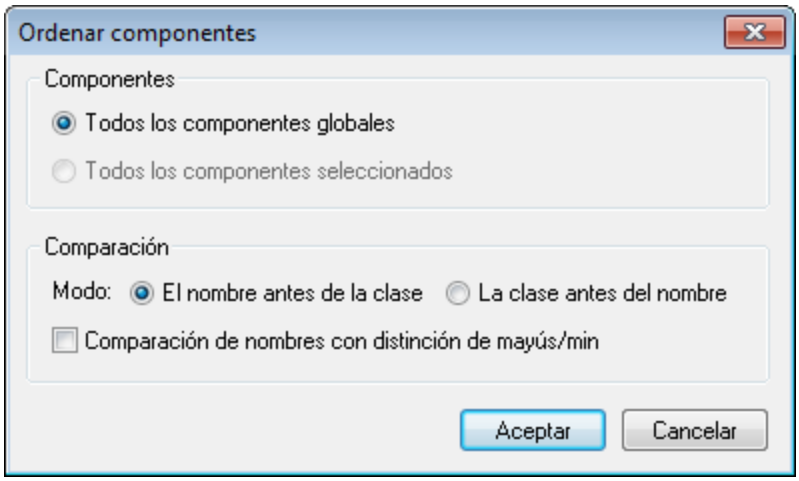
Algunos componentes globales (como los tipos complejos, las declaraciones de elemento y los modelos de grupo) tienen a la izquierda el icono **Cambiar a la vista del modelo de contenido** (*véase la lista de iconos anterior*). Este icono indica que el componente global puede tener un modelo de contenido que describe su estructura y su contenido.

Al hacer clic en este icono se pasa de la vista general del esquema a la vista del modelo de contenido 190 del componente global elegido. Otros componentes globales (como las anotaciones, los tipos simples o los grupos de atributos) no pueden tener un modelo de contenido y por tanto no tienen el icono **Cambiar a la vista del modelo de contenido**. Para volver a la vista general del esquema desde la vista del modelo de contenido, haga clic en el icono **Mostrar nodos globales** (*véase la lista de iconos que aparece más arriba*).

Ordenar los componentes globales

Para ordenar los componentes globales de la lista haga clic en el icono **Criterio de ordenación** de la barra de herramientas de la vista general del esquema (*véase la lista de iconos que aparece más arriba*). Aparece el cuadro de diálogo "Ordenar componentes" (*imagen siguiente*), donde puede elegir si se ordenan todos los componentes globales o solamente los componentes seleccionados. Puede mantener pulsada la tecla **Mayús** mientras hace clic en los componentes para seleccionar un grupo de componentes y pulsar la tecla **Ctrl** mientras hace clic para añadir más componentes al grupo seleccionado.

Nota: los componentes globales que por definición deben aparecer al principio del documento (como los elementos `include` e `import`) no cambian de posición cuando se usa la función de ordenación.

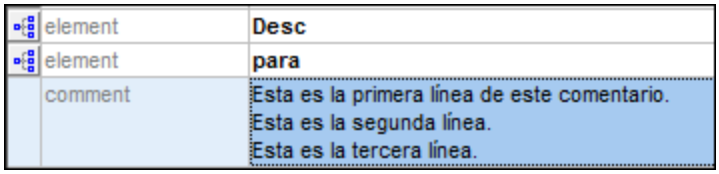


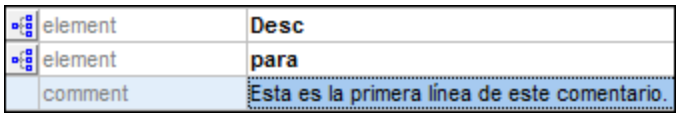
Después de seleccionar los componentes que se deben ordenar, puede elegir si se ordenan por orden alfabético (*El nombre antes de la clase*) o si se ordenan según el tipo de componente y después por orden alfabético (*La clase antes del nombre*).

El criterio de ordenación se implementa en el texto del esquema.

Comentarios globales: modo de presentación

Los comentarios globales se pueden presentar en un campo de texto multilínea (opción predeterminada) o en un campo de texto de una sola línea (*imágenes siguientes*).





Para cambiar de modo de presentación, haga clic en el icono **Comentarios** situado en la parte superior de la vista general del esquema y seleccione la opción deseada. Si selecciona el modo multilínea y desea crear una línea nueva, pulse **Ctrl+Entrar**. Si cambia el comentario multilínea al modo de una sola línea, al pasar el puntero por encima del comentario aparece un cuadro de texto emergente que incluye todas las líneas.

4.4.2.2 Componentes globales

Son componentes globales aquellos que se añaden como secundarios del elemento de nivel superior `xs:schema` (a diferencia de los componentes locales, que se crean dentro de otros componentes). A algunos componentes globales (como los tipos complejos, los elementos y los atributos) se les puede hacer referencia desde otros componentes del esquema.

Crear componentes globales en la vista general del esquema

Los componentes globales se crean y editan en la vista general del esquema[177], con ayuda de los iconos **Anexar**[179] e **Insertar**[179]. El modelo de contenido de un componente global (algunos componentes globales no pueden tener un modelo de contenido. *Consulte la tabla que aparece más abajo*) se crea y edita en la vista del modelo de contenido[190] del componente global. (Haga clic en el icono **Cambiar a la vista del modelo de contenido** situado a la izquierda del nombre de componente para abrir su vista del modelo de contenido[190].)

Mientras se crean en la vista general del esquema algunos componentes globales se añaden también al ayudante de entrada Componentes[229]. Si un componente tiene un modelo de contenido y hace doble clic en su nombre en el ayudante de entrada Componentes, entonces se abre su vista del modelo de contenido[190].

Si el componente global tiene una definición de tipo (simple o complejo), puede llegar a ella haciendo clic en el comando del menú contextual **Ir a la definición del tipo**. En el caso de los tipos simples integrados aparece una caja de texto con información sobre el tipo simple.

Nota: también puede crear algunos componentes globales (element, attribute, simpleType, complexType y group) mientras edita el modelo de contenido en la vista del modelo de contenido. Para ello, haga clic con el botón derecho en la vista y seleccione el comando **Global nuevo | < tipo de componente global >** del menú contextual.

© 2016-2022 Altova GmbH

Nota: mientras edita el modelo de contenido en la vista del modelo de contenido, puede convertir elementos locales en globales (o en tipos complejos globales, si el elemento tiene un elemento secundario o un atributo). Seleccione el elemento local, haga clic con el botón derecho y seleccione el comando **Convertir en... | Elemento global** o **Convertir en... | complexType global**.

Componente global	Posición en el esquema	Modelo de contenido
`include`	Al principio	No
`import`	Al principio	No
`redefine`	Al principio	No
`override` [1.1]	Al principio	No
`defaultOpenContent` [1.1]	Después de los elementos `include`, `import`, `redefine` y `override` y antes de los demás componentes.	Sí
`element`	En cualquier posición después de `defOpenCont`	Sí
`group`	En cualquier posición después de `defOpenCont`	Sí
`simpleType`	En cualquier posición después de `defOpenCont`	No
`complexType`	En cualquier posición después de `defOpenCont`	Sí
`attribute`	En cualquier posición después de `defOpenCont`	No
`attributeGroup`	En cualquier posición después de `defOpenCont`	No
`notation`	En cualquier posición después de `defOpenCont`	No
`annotation`	En cualquier posición después de `defOpenCont`	No
Comentario	En cualquier posición	No
Instrucción de procesamiento	En cualquier posición	No

A continuación señalamos algunos aspectos que debe tener en cuenta mientras edita estos componentes en la vista Esquema.

Componentes include, import, redefine y override

Estos cuatro componentes globales permiten reutilizar otros documentos de esquema dentro del documento de esquema actual.

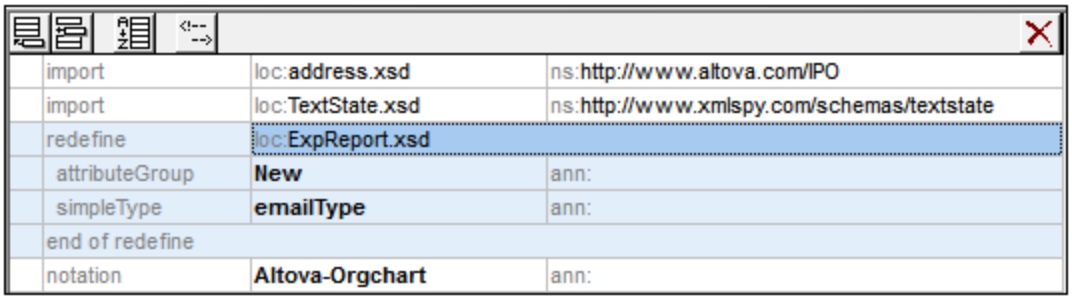
- El componente `include` reutiliza los documentos que tienen el mismo espacio de nombres de destino que el esquema actual.
- El componente `import` reutiliza los documentos que tienen un espacio de nombres de destino diferente al del esquema actual.
- Los componentes `redefine` y `override` son un tipo de `include` (tienen el mismo espacio de nombres de destino que el esquema actual). Sin embargo, estos componentes modifican parte de los esquemas incluidos. `Redefine` es una característica de XSD 1.0 que en XSD 1.1 se convierte en el elemento `override`, que es más flexible.

Estos cuatro componentes tienen un atributo `schemaLocation` que apunta al esquema que se debe volver a utilizar. Si hace doble clic en el campo `loc` de estos componentes en la vista Esquema, puede buscar el archivo que se debe reutilizar y configurar su ruta de acceso como relativa al documento actual. El componente `import` también tiene un atributo `namespace` que lleva el espacio de nombres de destino del esquema importado.

Cuando se reutiliza un esquema en el esquema actual (mediante el uso de `include`, `import`, `redefine` o `override`), sus componentes globales, espacios de nombres y restricciones de identidad aparecen en el ayudante de entrada Componentes 229 del documento actual.

Componentes redefine

En un componente `redefine` puede modificar los `complexType`, `simpleType`, `group` y `attributeGroup`. El componente que se debe redefinir estará en el esquema especificado en el campo `loc` del componente `redefine` (por ejemplo, en la imagen siguiente, los componentes que se deben redefinir están en el esquema `ExpReport.xsd`). Después de añadir el componente `redefine`, debe añadir el componente que se debe redefinir entre las filas `redefine` y `end of redefine` de la lista de componentes globales (por ejemplo, en la imagen siguiente se redefinen los componentes `New` y `emailType`). Estos dos componentes existen en el esquema `ExpReport.xsd` y se redefinen para el esquema actual.



Para redefinir un componente, siga estos pasos:

1. Seleccione la fila `end of redefine`.
2. Haga clic en el icono **Insertar** situado en la parte superior izquierda de la vista.
3. Seleccione el tipo de componente que desea definir (`complexType`, `simpleType`, `group` o `attributeGroup`). El componente se añade dentro del componente `redefine`.

 © 2016-2022 Altova GmbH

4. Ponga a este nuevo componente el mismo nombre que el componente que desea redefinir. El componente tendrá las mismas propiedades que el componente del esquema que desea reutilizar.
5. Para redefinir el componente selecciónelo y modifique sus propiedades en los ayudantes de entrada Detalles y Facetas o modifique su modelo de contenido, si lo tiene, en la vista del modelo de contenido.

Nota: en el componente `redefine` también puede arrastrar/colocar o copiar/pegar componentes de la lista de componentes globales o del ayudante de entrada Componentes.

A los componentes redefinidos se les puede hacer referencia desde otros componentes del esquema.

Componentes `override`

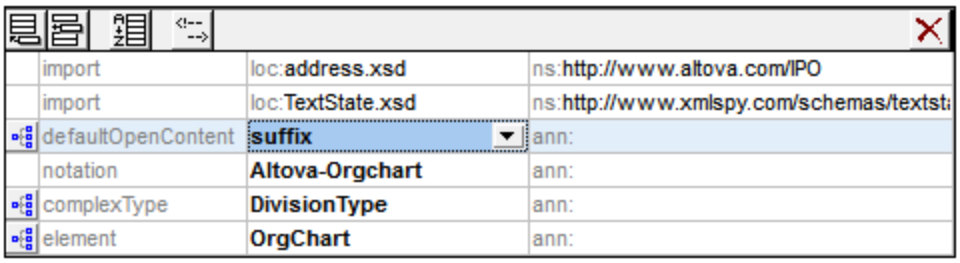
En un elemento `override` puede definir estos componentes: complexType, simpleType, elementos globales, atributos globales, group, attributeGroup y anotaciones. Si dentro de un elemento `override` se define uno de estos componentes, el componente reemplazará en el esquema reemplazado todos los componentes del mismo tipo que tengan el mismo nombre que el componente de reemplazo. El esquema reemplazado se especifica en el campo `loc` del componente `override`.

Los componentes `override` son diferentes de los componentes `redefine` (*más arriba*) porque los `override` se definen desde cero y no se basan en un componente ya existente. En la vista Esquema los componentes que se deben reemplazar se añaden igual que los componentes que se deben redefinir. Inserte el componente de reemplazo encima de la fila `end of override` y después defina sus propiedades (véase el párrafo anterior sobre componentes `redefine`). La principal diferencia entre un `override` y un `redefine` es que cuando se añade un componente a un `override`, el componente no se basa en ningún componente del esquema reutilizado.

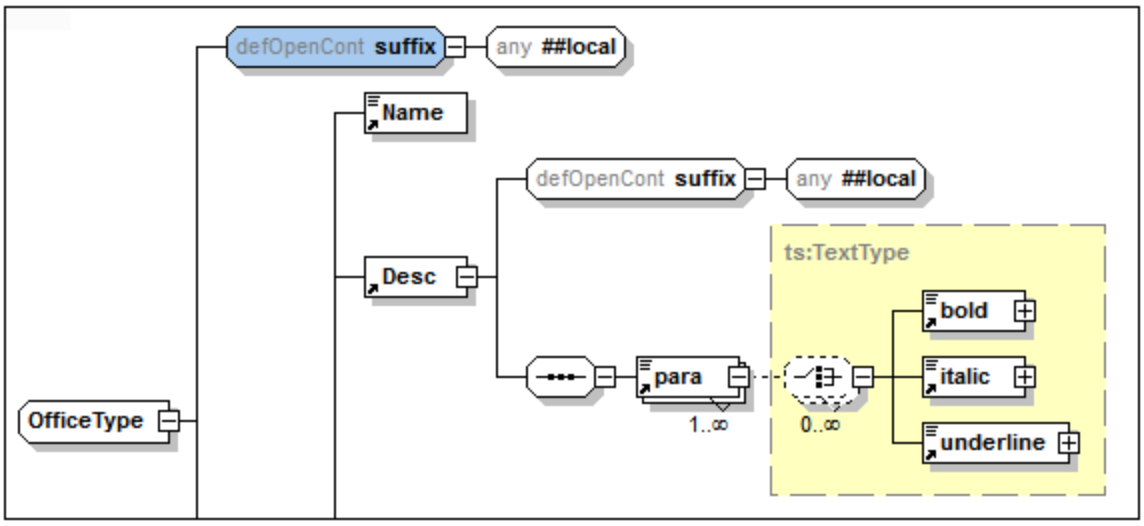
Elemento `defaultOpenContent`

El elemento `defaultOpenContent` es nuevo en la especificación XSD 1.1 y especifica que hay elementos sin definir que se pueden añadir a cualquier `complexType` de contenido mixto o de sólo elementos. Es similar al elemento `openContent` 209 (también nuevo en XSD 1.1) y la principal diferencia entre ellos es que el elemento `openContent` afecta a un solo `complexType`, mientras que el elemento `defaultOpenContent` afecta a todos los `complexType` del esquema.

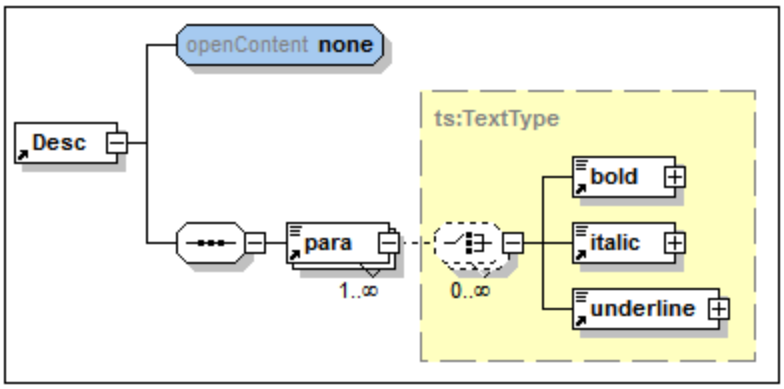
El elemento `defaultOpenContent` aparece una sola vez en el documento (*imagen siguiente*), entre los componentes `include`, `import`, `redefine` y `override` y el resto de los componentes. Tiene un atributo `mode`, cuyo valor puede ser `interleave` o `suffix`. El valor predeterminado es `interleave`.



El elemento `defaultOpenContent` tiene un modelo de contenido que se puede editar en la vista del modelo de contenido. Después de declararse, el elemento `defaultOpenContent` afectará a todos los complexType del esquema. En la imagen siguiente, por ejemplo, puede ver que el `defaultOpenContent` se aplicó automáticamente complexType `OfficeType` y `Desc`.



Para invalidar un elemento `defaultOpenContent` que afecta a un complexType determinado, añada un elemento secundario `openContent` al complexType. En la imagen siguiente, por ejemplo, al elemento `Desc` se le añadió un elemento `openContent`, que reemplaza al elemento `defaultOpenContent`.



Elementos globales `(element)`

In Schema Overview, you can create a global element. If the global element is to have a content model, then this is defined in the Content Model View of the global element. With the element selected in either view, you can define attributes 213, assertions 216, and identity constraints 220 in the respective tabs of the AAIDC pane 212. Facets and other properties can be defined in the element's Facets 234 and Details 232 entry helpers. Global elements can then be referenced by complex types.

En la vista general del esquema puede crear un elemento global. Si el elemento global debe tener un modelo de contenido, este se define en la vista del modelo de contenido del elemento global. Además, si selecciona el elemento en cualquiera de estas dos vistas, también puede definir sus atributos 213, aserciones 216 y restricciones de identidad 220 en las pestañas del panel AARID 212. También puede introducir facetas y otras propiedades en los ayudantes de entrada Facetas 234 y Detalles 232. A los elementos globales se les puede hacer referencia desde los complexType.

© 2016-2022 Altova GmbH

Modelos de grupo (`group`)

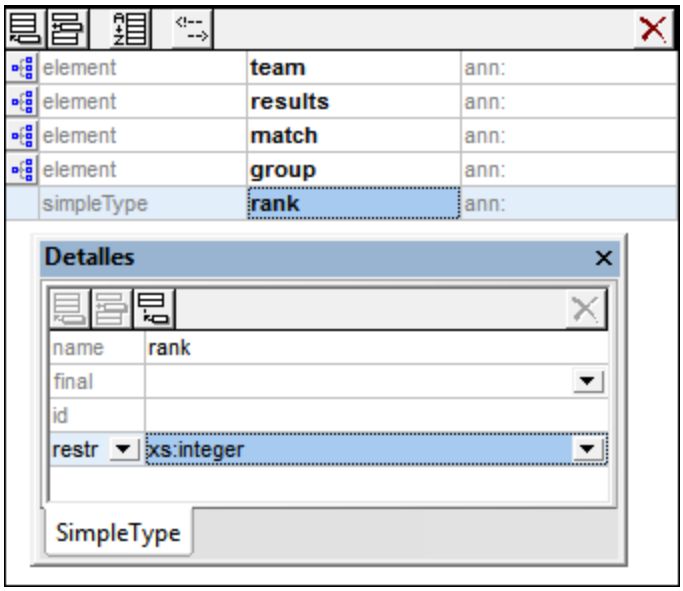
En la vista general del esquema puede crear modelos de grupo con nombre a los que se les puede hacer referencia desde los complexType. Un modelo de grupo con nombre (el elemento `xs:group`) le permite predefinir un modelo de contenido para volver a utilizarlo. Puede incluir uno de estos tres tipos de grupo: un grupo `sequence`, un grupo `choice` o un grupo `all`.

Para crear un modelo de grupo con nombre en la vista general del esquema, añada un componente `Group`, póngale un nombre y por último defina su modelo de contenido en la vista del modelo de contenido. El modelo de grupo se puede añadir más tarde al modelo de contenido de un complexType.

Tipos simples con nombre (`simpleType`)

En la vista general del esquema puede crear tipos simples con nombre (*imagen siguiente*), a los que se puede hacer referencia en las declaraciones de elemento y de atributo.

En el ayudante de entrada Detalles puede especificar el contenido del tipo simple (`restriction`, `list`, `union`) y sus tipos: tipo base, tipo de elemento, tipo de miembro, respectivamente. En la imagen siguiente, por ejemplo, el tipo base de la restricción del tipo simple es `xs:integer`. Para más información, consulte el subapartado dedicado al ayudante de entrada Detalles 232. Para restringir un tipo simple con facetas, use las opciones del ayudante de entrada Facetas 234.



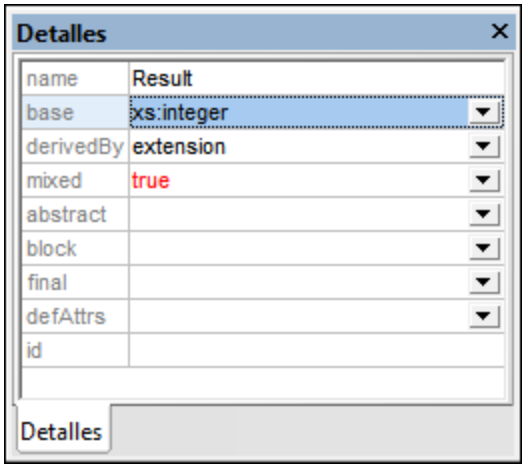
Nota: se pueden declarar tipos anónimos en un elemento o atributo de contenido simple tanto en la vista general del esquema 177 como en la vista del modelo de contenido 190. Cuando a la propiedad `derivedBy` se le da el valor `restriction`, `list` o `union` (en el ayudante de entrada Detalles), está creando un tipo simple anónimo dentro de dicha declaración de elemento o atributo. Puede definir facetas de restricción (en el ayudante de entrada Facetas) y más propiedades en el ayudante de entrada Detalles.

Tipos complejos con nombre `(complexType)`

En la vista general del esquema puede crear tipos complejos con nombre, a los que se puede hacer referencia en las declaraciones de elemento. Seleccione el tipo complejo con nombre y defina sus atributos[213] y aserciones[216] en las pestañas del panel AARID[212].

Un tipo complejo puede tener cuatro tipos de contenido (*véase la lista siguiente*). El tipo de contenido se especifica en el ayudante de entrada Detalles y, si quiere y está permitido, en el modelo de contenido en la vista del modelo de contenido[190].

- *Contenido simple:* establezca el tipo base del contenido simple (*imagen siguiente*). El atributo `mixed` (para contenido mixto) debe tener el valor `false` (valor predeterminado). Por eso, en la imagen siguiente, el valor `true` aparece en rojo. No puede tener un modelo de contenido.



- *Contenido de sólo elementos:* cree elementos secundarios en el diagrama del modelo de contenido. No habrá un tipo base.
- *Contenido mixto:* el atributo `mixed` debe tener el valor `true`. Los datos de caracteres pueden estar en cualquier posición del elemento, entre los nodos secundarios. Los datos de caracteres no tienen un tipo de datos y, por tanto, no puede haber un tipo base (*imagen anterior*). Los elementos secundarios se pueden crear en el diagrama del modelo de contenido.
- *Contenido vacío:* el elemento no tendrá datos de caracteres ni elementos secundarios. No puede haber un tipo base y el atributo `mixed` debe tener el valor `false`. Los datos de los elementos de contenido vacío se suelen almacenar en atributos.

Nota: puede definir atributos y aserciones para los cuatro tipos de contenido (en el panel AARID[212]).

Note: los tipos complejos anónimos se crean dentro de un elemento creando un modelo de contenido para dicho elemento en la vista del modelo de contenido[190].

Atributos y grupos de atributos globales `(attribute, attributeGroup)`

Los atributos y grupos de atributos globales se añaden al esquema en la vista general del esquema.

- Las propiedades de un `attribute` global se definen en el ayudante de entrada Detalles del atributo.

© 2016-2022 Altova GmbH

- Después de crear un `attributeGroup` global puede añadir atributos al grupo de esta manera: (i) seleccione el `attributeGroup` global en la lista de componentes globales, (ii) añada atributos al grupo en la pestaña *Atributos* del panel AARID 212 y por último (iii) defina las propiedades de cada atributo en el ayudante de entrada Detalles del atributo.

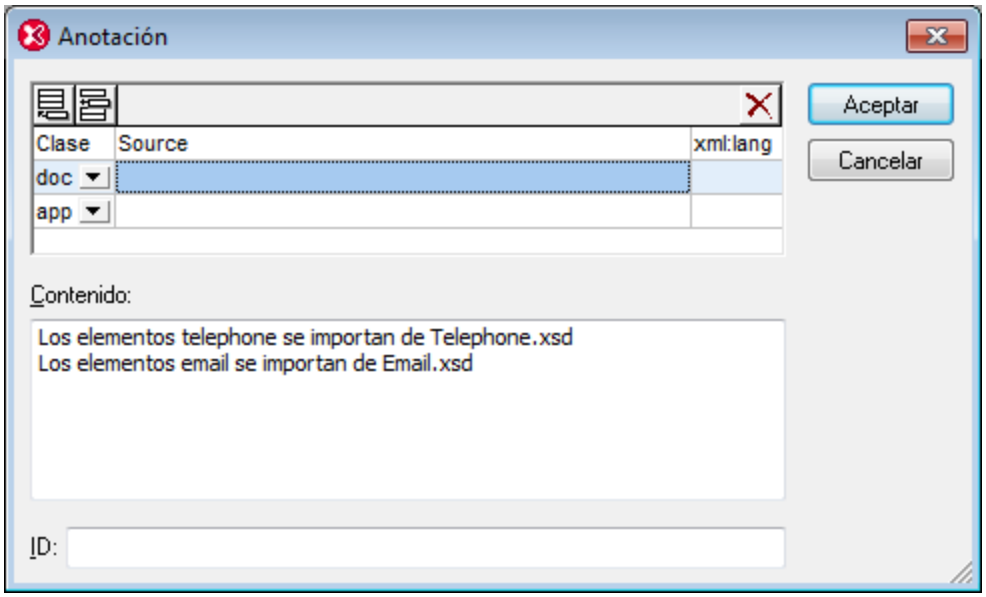
Una vez creados los `attribute` y `attributeGroup` globales, se les puede hacer referencia en las declaraciones de elementos y tipos complejos.

Notaciones `(notation)`

Los elementos `notation` son siempre globales, es decir, no hay notaciones locales. Las propiedades de un elemento `notation` se especifican en el ayudante de entrada Detalles. Su nombre se puede especificar directamente en la lista de componentes globales. Todas las notaciones del esquema aparecen en el ayudante de entrada Componentes.

Anotaciones globales

Las anotaciones globales son componentes globales diferentes de las anotaciones opcionales disponibles en algunos componentes globales. Las anotaciones globales se pueden editar en el cuadro de diálogo "Anotación" (imagen siguiente). Para abrir este cuadro de diálogo, haga clic con el botón derecho en la anotación y seleccione la opción Datos de anotación completos.



Una anotación puede tener un atributo `id` y varios elementos secundarios `documentation` y `appinfo`. Para añadir elementos `documentation` y `appinfo` haga clic en los botones **Anexar** o **Insertar** situados en la parte superior izquierda del diálogo y seleccione `doc` o `app` en el cuadro combinado. Después seleccione un `doc` o `app` del panel superior del diálogo e introduzca los datos en el cuadro de texto *Contenido*. Si desea crear una línea nueva en el cuadro de texto (para que el contenido sea multilínea) pulse la tecla **Entrar**. En el ejemplo de la imagen anterior está seleccionado el elemento `documentation`, que tiene contenido de dos líneas. Por cada elemento `documentation` y `appinfo` también puede introducir los atributos opcionales `source` y `xml:lang`.

En la vista general del esquema solamente aparece el primer elemento `documentation` o `appinfo` de la anotación global y se puede editar directamente en la lista de componentes globales. Si su contenido es multilínea, al pasar el puntero por encima de la anotación aparecen todas las líneas en un cuadro emergente.

Para ver o editar el contenido de los demás elementos `documentation` o `appinfo`, abra el cuadro de diálogo "Anotación" de la anotación global.

element	**Date**	ann:las fechas están en formato US
element	**Location**	ann:
annotation	Las direcciones se tomarán de la base de datos EuroCust	

Nota: las anotaciones opcionales disponibles para algunos componentes globales también se pueden editar en el cuadro de diálogo "Anotación" del mismo modo que las anotaciones globales.

Comentarios e instrucciones de procesamiento

En la vista general del esquema puede insertar comentarios e instrucciones de procesamiento en cualquier posición de la lista de componentes globales. Los comentarios y las instrucciones de procesamiento no se pueden añadir en la vista del modelo de contenido. Si dentro de un simpleType o complexType hay comentarios o instrucciones de procesamiento, estos se recopilan y se colocan al final del objeto en el que están incluidos. Por tanto, se recomienda usar anotaciones en lugar de comentarios en esos casos.

4.4.3 Vista del modelo de contenido

Un modelo de contenido describe la estructura y el contenido de un elemento. Los componentes globales que pueden tener un modelo de contenido son:

- los `complexType`
- los `element`
- los `group`
- los `defaultOpenContent`

En la vista general del esquema, los componentes globales que pueden tener un modelo de contenido se identifican con el icono Cambiar a la vista del modelo de contenido, situado a la izquierda del nombre del componente.

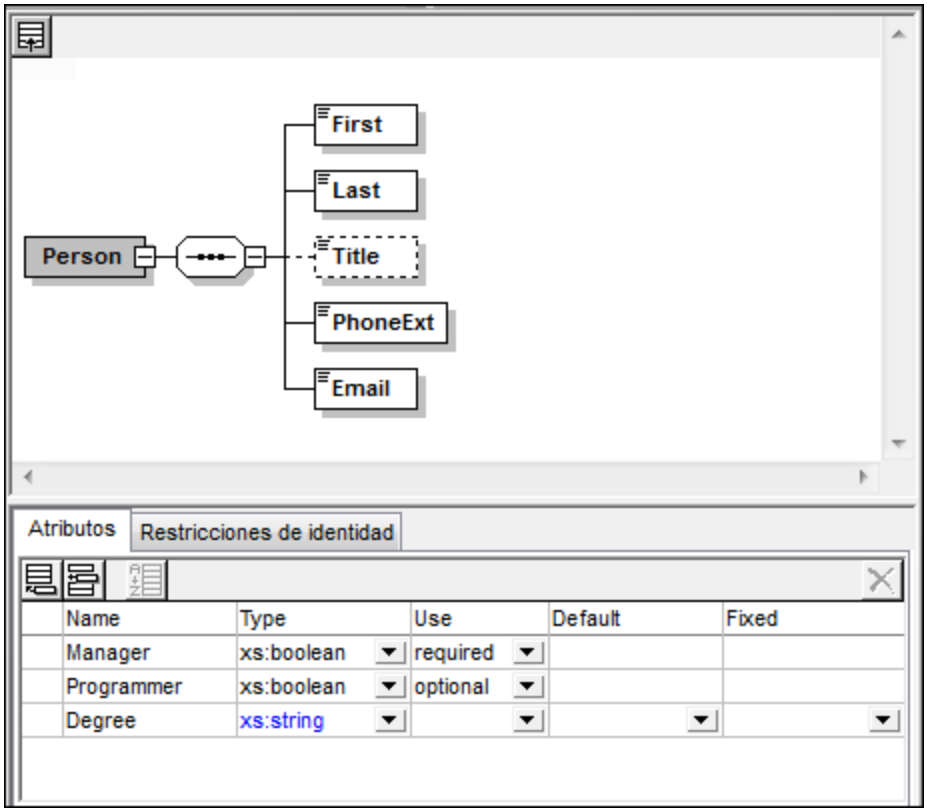
	Cambiar a la vista del modelo de contenido: disponible solamente para los componentes globales que tienen un modelo de contenido. Abre el modelo de contenido del componente global en la vista del modelo de contenido 190.
	Mostrar nodos globales: disponible en la vista del modelo de contenido. Abre la vista general del esquema 177.

Al hacer clic en el icono **Cambiar a la vista del modelo de contenido** se abre la vista del modelo de contenido del componente (*imagen siguiente*). Otra opción es (i) seleccionar un componente en la vista general del esquema 177 y después hacer clic en la opción de menú **Diseño de esquemas | Mostrar diagrama** o (ii) hacer doble clic en el nombre de un componente en el ayudante de entrada Componentes 228. Tenga en cuenta que sólo se puede abrir un modelo de contenido a la vez. Si hay un modelo de contenido abierto, puede saltar al modelo de contenido de un componente del modelo de contenido actual. Para ello, mantenga pulsada la tecla **Ctrl** mientras hace doble clic en el componente pertinente.

© 2016-2022 Altova GmbH

Descripción general de la vista del modelo de contenido

El modelo de contenido se representa en forma de árbol en la vista del modelo de contenido (*imagen siguiente*). El aspecto del árbol se puede configurar en el cuadro de diálogo "Configurar visualización del esquema" (**Diseño de esquemas | Configurar opciones de vista**[1378]).



Es necesario tener en cuenta que:

- hay dos tipos de objetos en el árbol del modelo de contenido: los compositores y los componentes. En el árbol también puede visualizar los atributos, las aserciones, las restricciones de identidad y los openContent en unos recuadros unidos al componente.
- cada nivel del árbol está unido a los niveles adyacentes por medio de un compositor. El modelo de contenido se puede extender tantos niveles como se quiera, sin limitaciones.
- si hace clic con el botón derecho en un objeto, puede añadir un objeto relativo a este primer objeto.
- los componentes del modelo de contenido pueden ser componentes locales o pueden hacer referencia a componentes globales.
- los objetos se pueden mover por el modelo de contenido mediante operaciones de arrastrar y colocar.
- puede usar teclas de acceso rápido para copiar (Ctrl+C) y pegar (Ctrl+V) objetos.
- las propiedades de un objeto se pueden editar en el ayudante de entrada Detalles y en el panel AARID[212].
- los atributos, las aserciones y las restricciones de identidad de un componente se presentan en un panel situado bajo la vista del modelo de contenido, el denominado panel AARID[212]. Los atributos y las restricciones de identidad también se pueden presentar dentro del diagrama del modelo de contenido, en lugar de en el panel AARID[212]. Para activar esta opción de presentación, use el cuadro

de diálogo Configurar visualización del esquema[1378]. También puede activarla con los botones Mostrar en el diagrama de la barra de herramientas Diseño de esquemas (imagen siguiente).



- los componentes del mismo nivel se pueden ordenar: haga clic en ellos con el botón derecho y seleccione el comando Ordenar componentes del menú contextual. Puede elegir entre ordenarlos según su nombre local o según la clase de componente.

Todas estas funciones se explican más detalladamente en los diferentes apartados de esta sección y en el tutorial.

Para volver a la vista general del esquema[177], haga clic en el icono **Mostrar nodos globales** o seleccione la opción de menú **Diseño de esquemas | Mostrar nodos globales**.

4.4.3.1 Objetos del modelo de contenido

En la vista del modelo de contenido los objetos que aparecen en el diagrama se pueden dividir en tres grandes grupos:

- Compositores[192]: (i) sequence, (ii) choice, (iii) all
- Componentes[193]: (i) element, (ii) complex type, (iii) group, (iv) comodín
- Objetos varios[196]: (i) attribute, (ii) attributeGroup, (iii) assert, (iv) restricciones de identidad, (v) openContent

A continuación describimos la representación gráfica de estos objetos.

Compositores

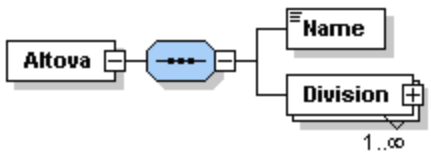
El **compositor** define el orden en que aparecen los elementos secundarios. Hay tres tipos de compositores: `sequence`, `choice` y `all`.

Para insertar un compositor:

1. Haga clic con el botón derecho en el elemento al que desea añadir elementos secundarios
2. Seleccione la opción **Agregar secundario | Sequence** (o **Choice** o **All**).

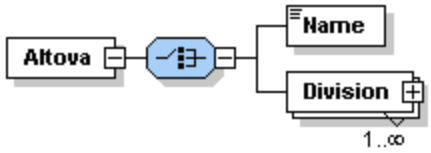
Se añade el compositor, que dependiendo del tipo tiene este aspecto:

- **Sequence**

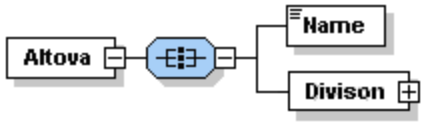


- Choice

 © 2016-2022 Altova GmbH



- All



Para cambiar el tipo de compositor, haga clic con el botón derecho en el compositor y seleccione **Cambiar modelo de grupo | Sequence** (o **Choice** o **All**). Después de añadir el compositor deberá añadir los elementos secundarios o un grupo.

Componentes

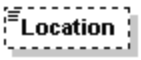
A continuación describimos los componentes que se usan en los modelos de contenido. La representación gráfica de cada uno de ellos ofrece información sobre el tipo del componente y sus propiedades estructurales.

- Elemento simple obligatorio



Detalles: el rectángulo significa que es un elemento y el borde sólido indica que el elemento es obligatorio. La ausencia de un intervalo numérico indica que es un elemento simple (es decir, con `minOcc=1` y `maxOcc=1`). El nombre del elemento es `Country`. El color azul indica que el elemento está seleccionado (se hizo clic sobre el elemento). Los elementos que no están seleccionados son de color blanco.

- Elemento simple opcional



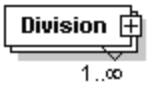
Detalles: el rectángulo significa que es un elemento y el borde discontinuo indica que el elemento es opcional. La usencia de un intervalo numérico indica que es un elemento simple (es decir, con `minOcc=1` y `maxOcc=1`). El nombre del elemento es Location.
Nota: la opción del menú contextual **Opcional** convierte un elemento obligatorio en opcional.

- Elemento múltiple obligatorio

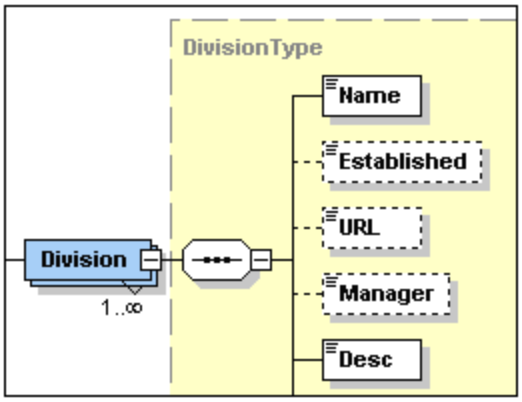


Detalles: el rectángulo significa que es un elemento y el borde sólido indica que es un elemento obligatorio. El intervalo numérico 1..5 significa que `minOcc=1` y `maxOcc=5`. El nombre del elemento es `Alias`.

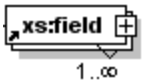
- Elemento múltiple obligatorio con elementos secundarios



Nota: la opción del menú contextual **Ilimitado** cambia el valor de `maxOcc` a `unbounded` (ilimitado). Si hace clic en el signo +, se expande la vista en forma de árbol y aparecen los elementos secundarios.



- Elemento que hace referencia a un elemento global



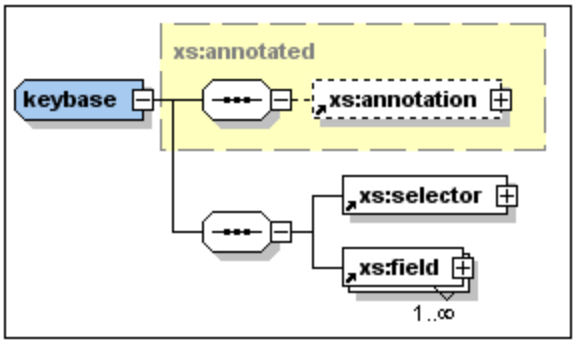
Detalles: la flecha en la zona inferior izquierda del elemento denota que el elemento hace referencia a un elemento global. El rectángulo significa que es un elemento y el borde sólido indica que el elemento es obligatorio. El intervalo numérico 1..infinito significa que `minOcc=1` y `maxOcc=unbounded`. El signo + denota un contenido complejo (es decir, contiene como mínimo un elemento o atributo secundario). El nombre del elemento es `xs:field`.
Nota: a los elementos globales se les puede hacer referencia desde definiciones de tipo complejo o simple, lo cual permite volver a utilizar una declaración global en varias posiciones del esquema. Puede crear una referencia a un elemento global de dos formas diferentes: (i) ponga al elemento local el mismo nombre que al elemento global o (ii) haga clic con el botón derecho en el elemento local y seleccione la opción **Referencia** del menú contextual. Para ver la definición de un elemento global, mantenga pulsada la tecla Ctrl mientras hace doble clic en el elemento. Otra opción es hacer clic con el botón derecho en el elemento y seleccionar la opción **Ir a la definición**. Si crea una referencia a un elemento que no existe, el nombre del elemento aparece en colo rojo, advirtiéndole de que no hay ninguna definición a la que se pueda hacer referencia.

- Tipo complejo (complexType)



 © 2016-2022 Altova GmbH

Detalles: el hexágono irregular con un signo + significa que es un tipo complejo. El nombre del tipo complejo es `keybase`. Se trata de un tipo complejo global. Un tipo complejo global se declara en la vista general del esquema y su modelo de contenido se suele definir en la vista del modelo de contenido. Un tipo complejo global se puede usar (i) bien como tipo de datos de un elemento, (ii) bien como tipo base de otro tipo complejo. Para ello es necesario asignar el tipo complejo al elemento o al otro tipo complejo en el ayudante de entrada Detalles (bien en la vista del modelo de contenido, bien en la vista general del esquema).



El tipo complejo `keybase` se declaró en la vista general del esquema con un tipo base `xs:annotated`. El tipo base se representa dentro de un rectángulo con un borde discontinuo de color gris y el fondo amarillo. Después, en la vista del modelo de contenido, se crearon los elementos secundarios `xs:selector` y `xs:field`. (Observe las pequeñas flechas de la esquina inferior izquierda de los rectángulos `xs:selector` y `xs:field`. Estas flechas indican que ambos elementos hacen referencia a los elementos globales del mismo nombre.)

Un tipo complejo local se define directamente en la vista del modelo de contenido. Esto se hace creando un elemento o atributo secundario para un elemento. Los tipos complejos locales no tienen un símbolo propio.

Nota: el tipo base de un modelo de contenido se representa en forma de rectángulo con un borde discontinuo de color gris y el fondo amarillo. Haga doble clic en el nombre del tipo base para ir a su modelo de contenido.

- Modelo de grupo (group)



Detalles: el octágono irregular con un signo + indica que es un modelo de grupo. Un modelo de grupo le permite definir y volver a utilizar declaraciones de elementos.
Nota: cuando se declara un modelo de grupo (en la vista general del esquema) se le da un nombre. Después define su modelo de contenido (en la vista del modelo de contenido), asignándole un compositor secundario que contiene las declaraciones de elemento. Cuando se usa un modelo de grupo, se inserta como secundario o se inserta o anexa dentro del modelo de contenido de otro componente (en la vista modelo de contenido).

- Comodines

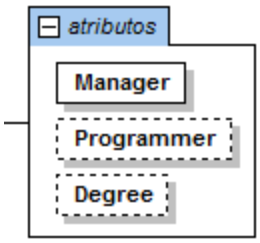


Detalles: el octágono irregular con la marca any a la izquierda indica que se trata de un comodín.
Nota: los comodines se usan como marcadores de posición para elementos no indicados en el esquema o de otros espacios de nombres. `##other` = los elementos pueden pertenecer a cualquier espacio de nombres que no sea el espacio de nombres de destino definido en el esquema. `##any` = los elementos pueden pertenecer a cualquier espacio de nombres. `##targetNamespace` = los elementos deben pertenecer al espacio de nombres de destino definido en el esquema. `##local` = los elementos no pueden pertenecer a un espacio de nombres. `anyURI` = los elementos pertenecen al espacio de nombres que indique el usuario.

Objetos varios

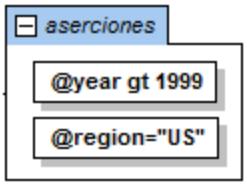
En esta categoría se encuentran los atributos (`attribute`), los grupos de atributos (`attributeGroup`), las aserciones (`assert`), las restricciones de identidad y los `openContent`.

- Atributos (attribute) y grupos de atributos (attributeGroup)



Nota: los atributos se pueden editar en el diagrama del modelo de contenido y en el ayudante de entrada Detalles. Los atributos se pueden ver y editar en el diagrama de la vista del modelo de contenido o en el panel AARID[212] situado debajo de la vista. Para pasar de una vista a otra, haga clic en el icono **Mostrar atributos**. Para cambiar el orden de los atributos de un elemento, arrastre el atributo fuera del recuadro contenedor y suéltelo cuando aparezca una flecha en la posición en la que desea colocarlo.

- Aserciones (assert)

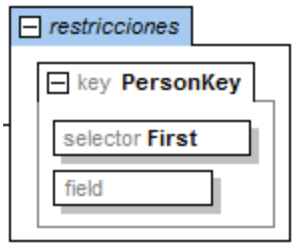


Detalles: se identifican con la palabra "aserciones" en cursiva en un recuadro que se puede expandir. Cada aserción aparece dentro de un pequeño recuadro dentro del recuadro Aserciones.
Nota: las aserciones se pueden editar en el diagrama o en el ayudante de entrada Detalles. Las aserciones se pueden ver y editar en el diagrama de la vista del modelo de contenido o en el panel AARID[212] situado debajo de la vista. Para pasar de una vista a otra, haga clic en el icono **Mostrar**

 © 2016-2022 Altova GmbH

aserciones. Para cambiar el orden de las aserciones de un elemento, arrastre la aserción fuera del recuadro contenedor y suéltela cuando aparezca una flecha en la posición en la que desea colocarla.

- Restricciones de identidad

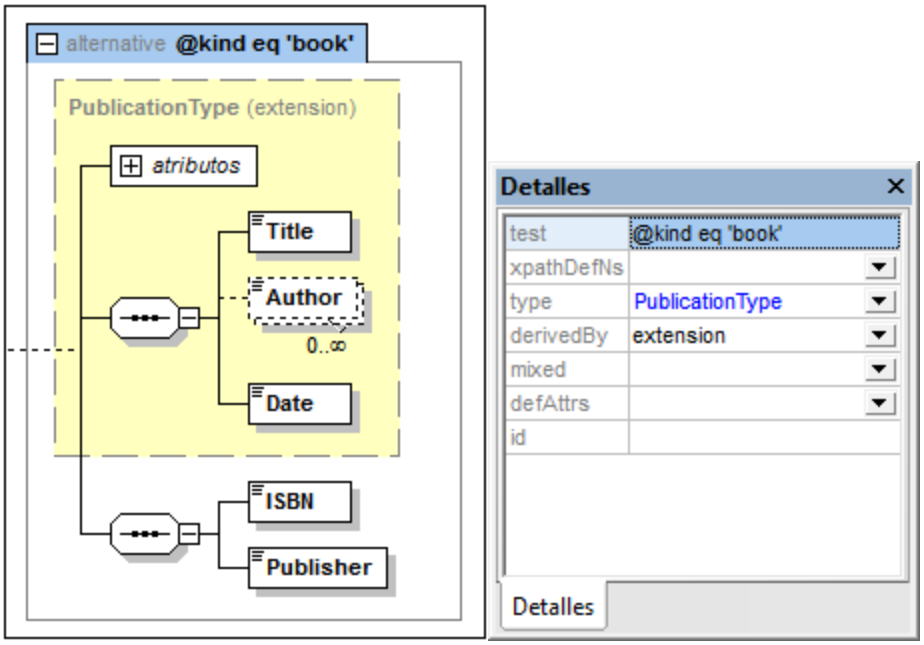


Detalles: se identifican con la palabra "restricciones" en cursiva en un recuadro que se puede expandir.
Nota: las restricciones de identidad que aparecen en el modelo de contenido de un componente muestran las restricciones definidas con los elementos key y keyref y con el elemento unique. Las restricciones de identidad definidas usando el tipo de datos `ID` no aparecen en el diagrama del modelo de contenido sino en el ayudante de entrada Detalles. Las restricciones de identidad se pueden ver y editar en la vista del modelo de contenido o en el panel AARID 212 situado debajo de la vista. Para pasar de una vista a otra, haga clic en el icono **Mostrar restricciones**.

- Asignación de tipos condicionales



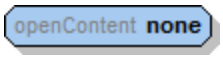
Detalles: el elemento alternative se presenta en un recuadro que contiene la expresión XPath que se probará (*imagen siguiente*). El tipo de elemento `alternative` se especifica en el ayudante de entrada Detalles. Si el tipo es un complexType, esto aparece en el recuadro expandido del elemento `alternative`, donde se puede seguir editando (*imagen siguiente*). Los simpleType no aparecen en el diagrama, pero se pueden definir en la pestaña SimpleType del ayudante de entrada Detalles.



Nota: el elemento `alternative` es nuevo en la especificación XSD 1.1. Si la expresión XPath da como resultado `true`, el tipo especificado por el elemento `alternative` será el tipo seleccionado. Se selecciona el primer elemento `alternative` de los elementos del mismo nivel de `alternative` que dé como resultado `true`. Esto significa que el orden de los elementos `alternative` es importante. Para ordenar los elementos de otra forma, seleccione un recuadro del elemento `alternative` y arrástrelo hasta la posición deseada. Para más información consulte el apartado Asignación de tipos condicionales [205].

- defaultOpenContent, openContent





Detalles: los elementos `defaultOpenContent` y `openContent` se indican en la vista del modelo de contenido por medio de las etiquetas `openContent` y `defOpenContent`. Los elementos comodín se señalan con un recuadro any (*imagen anterior*).
Nota: los elementos `defaultOpenContent` y `openContent` son nuevos en la especificación XSD 1.1. Default Open Content es un componente global, que se crea en la vista general del esquema [177]. En la vista del modelo de contenido del modelo de contenido de un componente puede invalidar el elemento `defaultOpenContent` con un elemento `openContent` del componente que reemplaza al elemento `defaultOpenContent` del esquema. Basta con añadir `openContent` como secundario del componente. El recuadro `defOpenContent` se sustituye por un recuadro `openContent`. En la vista del modelo de contenido puede editar el atributo mode del `openContent` y el espacio de nombres de su elemento comodín, bien en el diagrama o en el ayudante de entrada Detalles. También puede modificar el `defaultOpenContent` (para todo el esquema) desde su representación en el modelo de contenido de cualquier complexType.

Nota:

© 2016-2022 Altova GmbH

- Las líneas de descriptor de propiedad definidas en el cuadro de diálogo Configurar visualización del esquema[1378] se pueden activar o desactivar si hace clic en el icono "Agregar detalles predefinidos" de la barra de herramientas.
- Los atributos, las aserciones y las restricciones de identidad se pueden ver en el diagrama del modelo de contenido o en el panel AARID[212] (situado debajo de la vista del modelo de contenido). Para cambiar de una vista a otra, pulse los iconos Mostrar en el diagrama para atributos, aserciones y restricciones de identidad.
- Desde la vista de modelo de contenido puede pasar al modelo de contenido de otro componente global comprendido en el diagrama activo. Para ello pulse la tecla **Ctrl** mientras hace doble clic en el componente global que le interesa. Para ir al modelo de contenido de un tipo base, haga doble clic en el nombre del tipo base.

4.4.3.2 Edición en la vista del modelo de contenido

Este apartado se divide en varias partes:

- Configurar la vista del modelo de contenido[199]
- Atributos, aserciones y restricciones de identidad[199]
- Iconos de la vista del modelo de contenido[200]
- Operaciones del menú contextual[200]
- Teclas de acceso directo y operaciones de arrastrar y colocar[203]
- Propiedades de los componentes[204]
- Anotaciones[204]
- Comentarios e instrucciones de procesamiento[205]
- Documentación del modelo de contenido[205]

Configurar la vista del modelo de contenido

Puede configurar la vista del modelo de contenido para todo el esquema en el cuadro de diálogo "Configurar visualización del esquema" (**Diseño de esquemas | Configurar opciones de vista**). Para más información consulte el apartado Configurar opciones de vista[1378] de la *Comandos de menú*. Recuerde que las opciones elegidas en este cuadro de diálogo afectan a todo el esquema y también cuando documente o imprima el esquema.

Atributos, aserciones y restricciones de identidad

Los atributos, las aserciones y las restricciones de identidad de un componente pueden aparecer en el panel AARID[212] (situado bajo la vista del modelo de contenido) o en forma de recuadros dentro del modelo de contenido propiamente dicho, es decir, en el diagrama. Para utilizar esta última vista, seleccione la opción correspondiente en el cuadro de diálogo Configurar visualización del esquema[1378]. También puede cambiar de una vista a otra con los botones **Mostrar en el diagrama** de la barra de herramientas Diseño de esquemas (*imagen siguiente*).



Para más información sobre cómo insertar y editar atributos, aserciones y restricciones de identidad, consulte el apartado Atributos, aserciones y restricciones de identidad[212].

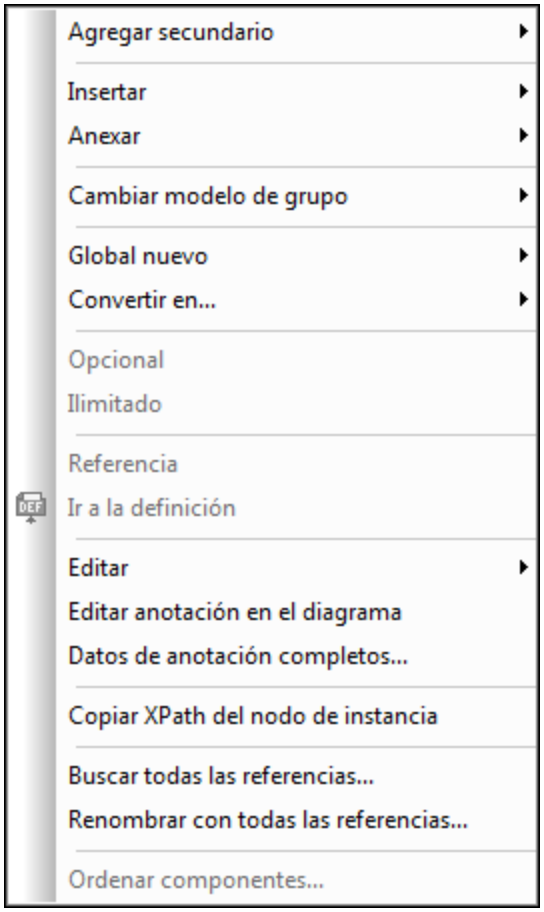
Iconos de la vista del modelo de contenido

	*Mostrar nodos globales:* disponible en la vista del modelo de contenido. Abre la vista general del esquema 177.
	*Agregar detalles predefinidos:* disponible en la vista del modelo de contenido. Muestra/oculta los detalles predefinidos en los recuadros de los componentes los componentes.
	*Mostrar atributos en diagrama:* disponible en la vista del modelo de contenido. Habilita/deshabilita la representación de los atributos en el diagrama, en lugar de en el panel AARID.
	*Mostrar aserciones en diagrama:* disponible en la vista del modelo de contenido. Habilita/deshabilita la representación de las aserciones en el diagrama, en lugar de en el panel AARID.
	*Mostrar restricciones en diagrama:* disponible en la vista del modelo de contenido. Habilita/deshabilita la representación de las restricciones de identidad en el diagrama, en lugar de en el panel AARID.
	*Mostrar restricciones de identidad:* disponible en la vista del modelo de contenido. Habilita/deshabilita la información sobre las IDC.

Operaciones del menú contextual

Varias operaciones de edición de la vista del modelo de contenido se llevan a cabo con la ayuda de un menú contextual (*imagen siguiente*) que aparece al hacer clic con el botón derecho dentro de la vista. En el menú contextual las operaciones de edición que no estén permitidas en ese momento dado se deshabilitan. Las operaciones se llevan a cabo en relación al objeto en el que se hizo clic con el botón derecho. Por ejemplo, si se añade un elemento secundario, se añade en el objeto en el que hizo clic con el botón derecho.

© 2016-2022 Altova GmbH



A continuación describimos las operaciones que se pueden llevar a cabo desde este menú contextual:

- *Agregar compositores y componentes secundarios:* el comando **Agregar secundario** abre un submenú. Seleccione un compositor o componente en este submenú.

- *Insertar/anexar compositores y componentes:* inserta el compositor o componente en el mismo nivel jerárquico que el objeto seleccionado: antes del objeto (**Insertar**) o después del último elemento del mismo nivel del objeto (**Anexar**).

- *Cambiar un compositor:* haga clic con el botón derecho en un compositor y seleccione **Cambiar modelo de grupo | <tipo de compositor>**.

- *Crear componentes globales:* (i) el comando **Global nuevo** está siempre disponible. Abre un submenú donde puede seleccionar el componente global nuevo que desea crear. (ii) Si un objeto puede crearse como componente global, se habilita el comando **Convertir en...** del menú contextual. Este comando abre un submenú donde puede seleccionar en qué tipo de componente global se convierte el objeto seleccionado. El objeto de la vista del modelo de contenido se crea como componente global y el objeto contiene una referencia al componente global recién creado.

- *Definir el número de apariciones:* use los comandos **Opcional** e **Ilimitado** para definir el número de veces que puede aparecer el componente: (i) **Opcional** = 0 o 1; (ii) **Opcional** + **Ilimitado** = de 0 a

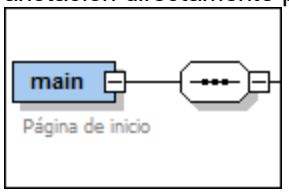
infinito; (iii) **Ilimitado** = de 1 a infinito; (iv) Ni **Opcional** ni **Ilimitado** = 1. (Nota: **Opcional** configura el atributo `minOccurs` del componente, **Ilimitado** configura el atributo `maxOccurs`.)

- *Alternar definiciones locales y globales:* si existe un elemento global cuyo nombre coincide con el de un elemento local, use el comando **Referencia** para alternar la referencia a la definición global (con **Referencia** seleccionado) y la definición local (con **Referencia** sin seleccionar).

- *Pasar a otro modelo de contenido:* haga clic con el botón derecho en el componente cuyo modelo de contenido desea ver y seleccione **Ir a la definición**. (El comando solamente está habilitado si el objeto en el que hizo clic tiene un modelo de contenido). También puede pulsar la tecla **Ctrl** mientras hace doble clic en el componente.

- *Editar detalles predefinidos:* si eligió presentar los detalles predefinidos en el diagrama (con el icono **Agregar detalles predefinidos** de la barra Diseño de esquemas), el comando **Editar** abre un submenú con los detalles predefinidos que se pueden editar. Seleccione una opción y edite el valor en el diagrama.

- *Crear y editar anotaciones para el compositor/componente:* el comando **Editar anotación en el diagrama** crea una línea de anotación bajo el compositor/componente (*imagen siguiente*). En esta línea puede introducir y editar la anotación. Si ya existe la anotación y hace clic en el comando, el texto de la anotación se resalta y puede editarlo. También puede hacer doble clic en el texto de la anotación directamente para empezar a editarlo.

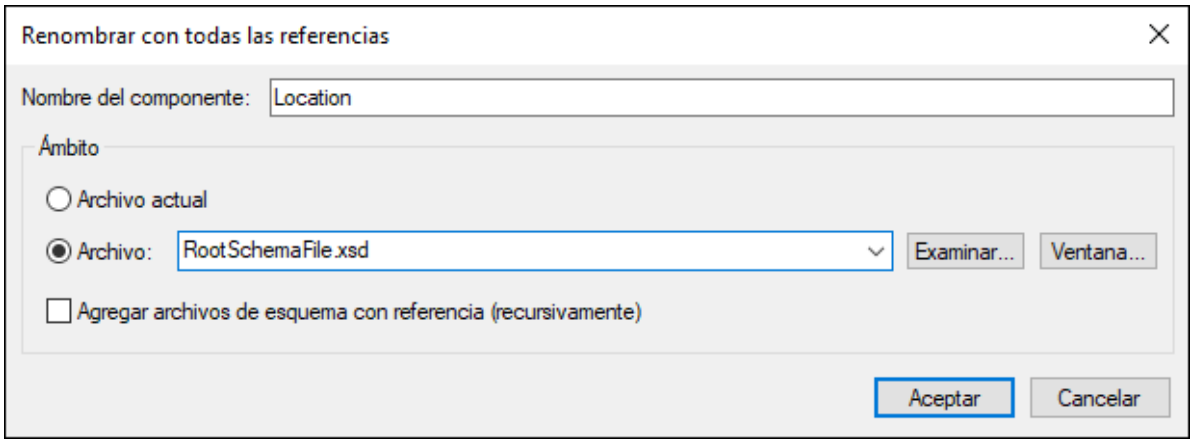
  

  En el documento de esquema XML, la anotación se crea dentro del elemento `annotation` o `documentacion` del compositor o componente (ver más abajo).

- *Copiar XPath del nodo de instancia:* el comando **Copiar XPath del nodo de instancia** se habilita si el elemento/atributo se definió dentro de un elemento o tipo complejo global. Este comando copia en el portapapeles una expresión XPath que localiza el nodo seleccionado. La expresión de ruta de ubicación empieza en el componente global cuyo modelo de contenido aparece actualmente en la vista del modelo de contenido.

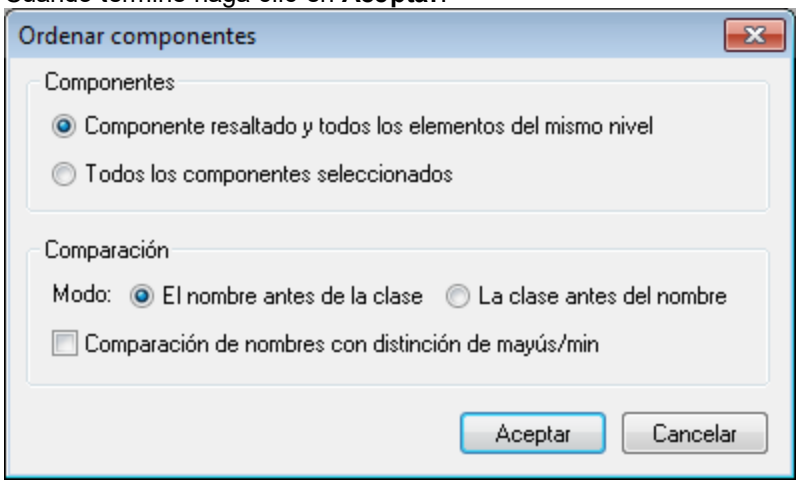
- *Buscar y cambiar el nombre de un componente:* los comandos **Buscar todas las referencias** y **Renombrar con todas las referencias** están disponibles para todos los elementos globales. El primer comando busca todas las apariciones del componente seleccionado en el documento activo y, si lo desea, también en los archivos de esquema relacionados con el documento activo.

© 2016-2022 Altova GmbH



En el ejemplo de la imagen anterior se pretende reemplazar el nombre `Email` con el nombre del componente seleccionado y todas sus referencias en el ámbito de búsqueda. Para más información consulte el apartado Buscar y renombrar componentes globales 487.

- *Ordenar declaraciones y referencias:* con el comando **Ordenar componentes** puede ordenar todos los componentes seleccionados o los elementos del mismo nivel del componente seleccionado. En el cuadro de diálogo "Ordenar componentes" (*imagen siguiente*) puede definir el criterio de ordenación. Cuando termine haga clic en **Aceptar**.



Para seleccionar varios componentes, pulse la tecla **Mayús** o **Ctrl** mientras selecciona los componentes. Puede usar el nombre de los componentes como primera clave de ordenación y la clase de componente como segunda clave de ordenación o viceversa.

Teclas de acceso rápido y operaciones de arrastrar y colocar

En la vista del modelo de contenido puede copiar y pegar elementos con las teclas de acceso rápido **Ctrl+C** y **Ctrl+V** respectivamente. Los objetos se pegan como objetos secundarios del objeto seleccionado. Si por motivos estructurales no es posible pegar el objeto, entonces aparece un mensaje a tal efecto.

También puede arrastrar (i) objetos a otra posición dentro del diagrama y (ii) algunos componentes, como los atributos, desde el ayudante de entrada Componentes hasta el diagrama.

Propiedades de los componentes

Si configuró la vista del modelo de contenido para ver los detalles predefinidos de los componentes en el diagrama, entonces puede editar estos datos desde el diagrama directamente. Para ver los detalles predefinidos en el diagrama pulse el icono **Agregar detalles predefinidos** de la barra de herramientas (*ver la lista de iconos que aparece más arriba*).

Si lo prefiere, puede editar las propiedades del componente en el ayudante de entrada Detalles 232 y los cambios se reflejan inmediatamente en los recuadros del diagrama (si configuró la vista para ver los detalles predefinidos en el diagrama).

Anotaciones

Las anotaciones de los esquemas XML se guardan en el elemento `annotation`. Hay dos tipos de anotaciones, que se almacenan en sendos elementos secundarios del elemento `annotation`:

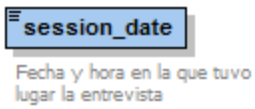
- el secundario `documentation`: contiene información útil destinada a los editores del esquema
- el secundario `appinfo`: permite insertar un script o información que puede ser utilizada por una instrucción de procesamiento

A continuación ofrecemos un ejemplo de un elemento de anotación que contiene ambos tipos de secundarios.

```
<xs:element name="session_date" type="xs:dateTime" nillable="true">
   <xs:annotation>
      <xs:documentation>Fecha y hora en la que tuvo lugar la entrevista</xs:documentation>
      <xs:appinfo source="http://www.altova.com/datehandlers/interviews">separator =
:</xs:appinfo>
   </xs:annotation>
</xs:element>
```

En la vista del modelo de contenido puede crear y editar anotaciones en los compositores y los componentes:

1. Haga clic con el botón derecho en el compositor o componente.



2. Seleccione el comando **Datos de anotación completos** del menú contextual. Se abre el cuadro de diálogo "Anotación" (*imagen siguiente*). Si existe una anotación para ese elemento (ya sea `documentation` o `appinfo`), el diálogo incluye una fila para dicha anotación.

 © 2016-2022 Altova GmbH



3. Para crear un elemento `appinfo` pulse el icono **Anexar** o **Insertar** situado en la esquina superior izquierda para anexar o insertar una fila nueva respectivamente.
4. En el campo *Clase* de la fila nueva seleccione la opción `app` del menú desplegable.
5. En el panel *Contenido* del diálogo escriba el script o la información que debe procesar la instrucción de procesamiento.
6. Si lo desea, en el campo *Source* puede escribir el URI de origen donde la instrucción de procesamiento puede encontrar más información.

Comentarios e instrucciones de procesamiento

Cuando se carga un documento de esquema XML o se cambia de vista en XMLSpy, los comentarios y las instrucciones de procesamiento definidas dentro de los elementos simpleType y complexType se recopilan y se colocan al final del objeto en el que están incluidos. Por eso en estos casos recomendamos usar anotaciones en lugar de comentarios.

Documentación del modelo de contenido

Puede generar documentación detallada[1373] sobre su esquema en formato HTML y MS Word. La documentación abarca todos los componentes globales del esquema y la lista de componentes globales aparece al principio de la documentación. Esta lista incluye enlaces al modelo de contenido de cada componente. Además, los elementos relacionados (como los elementos secundarios o los tipos complejos) también aparecen enlazados por medio de hipervínculos, para que pueda navegar fácilmente de elemento en elemento. Cuando quiera generar la documentación del esquema, seleccione el comando **Diseño de esquemas | Generar documentación**.

4.4.3.3 Asignación de tipos condicionales

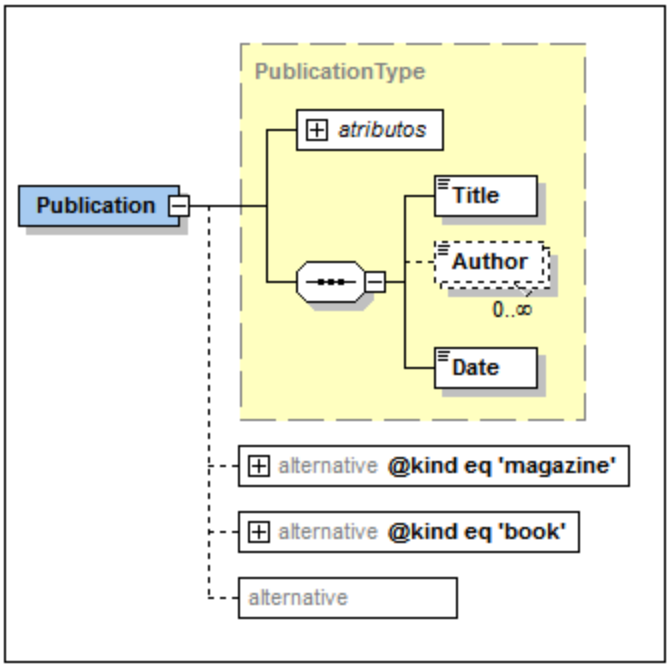
La asignación de tipos condicionales es una característica de la especificación XSD 1.1. Con esta característica podemos permitir que el tipo de un elemento venga determinado por el contenido del documento

XML, en concreto por el valor de los atributos del elemento o por la presencia o ausencia de determinados atributos. Por ejemplo, imaginemos que el documento XML tiene este elemento:

```
<publication kind="magazine">
   ...
</publication>
```

En el esquema, podemos indicar que el tipo del elemento `publication` puede variar dependiendo del valor que tome el atributo `@kind` del elemento en el documento de instancia. En el esquema esto se indica usando el elemento `alternative`, que es nuevo en la especificación XSD 1.1. Podemos especificar varios tipos alternativos, cada uno en un elemento `alternative` diferente.

En el ejemplo de la imagen siguiente, el elemento `Publication` se declaró con tres elementos secundarios `alternative`. Dos de ellos tienen un atributo `test` (`@kind eq 'magazine'` y `@kind eq 'book'`). El tercer elemento `alternative` no tiene el atributo `test`, pero tiene una asignación de tipo simple `xs:error` (esta asignación se hizo en el ayudante de entrada Detalles y no aparece en el diagrama). Si se desencadena, esta asignación devuelve un error de validación XML.



El código de estas declaraciones sería este:

```
<xs:complexType name="PublicationType">
   <xs:sequence>
      <xs:element name="Title" type="xs:string"/>
      <xs:element name="Author" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="Date" type="xs:gYear"/>
   </xs:sequence>
   <xs:attribute name="kind" type="xs:string"/>
</xs:complexType>

<xs:complexType name="MagazineType">
   <xs:complexContent>
      <xs:restriction base="PublicationType">
         <xs:sequence>
```

 © 2016-2022 Altova GmbH

```
            <xs:element name="Title" type="xs:string"/>
            <xs:element name="Date" type="xs:gYear"/>
         </xs:sequence>
      </xs:restriction>
   </xs:complexContent>
</xs:complexType>

<xs:element name="Publication" type="PublicationType">
   <xs:alternative test="@kind eq 'magazine'" type="MagazineType"/>
   <xs:alternative test="@kind eq 'book'">
      <xs:complexType>
         <xs:complexContent>
            <xs:extension base="PublicationType">
               <xs:sequence>
                  <xs:element name="ISBN" type="xs:string"/>
                  <xs:element name="Publisher" type="xs:string"/>
               </xs:sequence>
            </xs:extension>
         </xs:complexContent>
      </xs:complexType>
   </xs:alternative>
   <xs:alternative type="xs:error"/>
</xs:element>
```

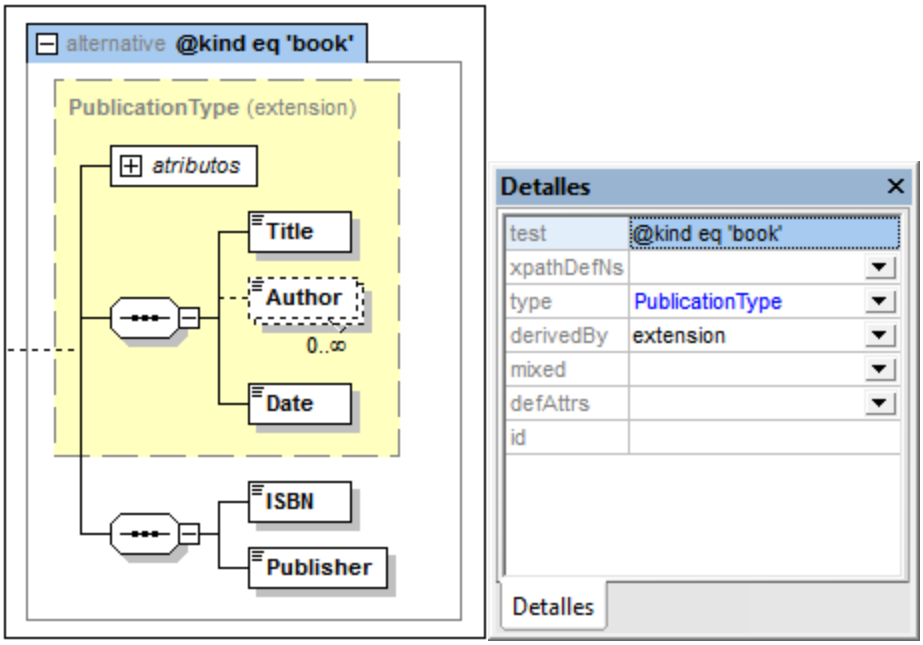
Es importante tener en cuenta que:

- De todos los elementos secundarios `alternative` se selecciona el primero que dé como resultado true. Es decir, el orden de los elementos `alternative` es muy importante. Si desea reordenar los elementos `alternative`, simplemente seleccione uno de los recuadros y arrástrelo a su nueva posición.
- El elemento `Publication` tiene un tipo (`PublicationType`). El tipo sirve de tipo predeterminado si no se usa ninguno de los elementos `alternative`. Sin embargo, en nuestro ejemplo se usará el elemento `alternative` de tipo `xs:error` si los dos elementos condicionales `alternative` anteriores devuelven el valor `false`.
- Si no se cumple con ninguna condición de los elementos `alternative` y el elemento no tiene un tipo predeterminado, entonces se asigna al elemento el tipo `anyType`. En tal caso, el elemento puede tener cualquier contenido que tenga formato XML correcto.
- El elemento `alternative` y el tipo simple `xs:error` son nuevos en la especificación XSD 1.1.

Edición de elementos **`alternative`** en la vista del modelo de contenido

En la vista del modelo de contenido puede añadir un tipo alternativo como secundario de una declaración de elemento con el menú contextual.

El tipo del elemento `alternative` se especifica en el ayudante de entrada Detalles. Si el tipo es un complexType, este aparece en el recuadro expandido del elemento `alternative`, donde se puede seguir editando (*imagen siguiente*). Los tipos simples no aparecen en el diagrama, pero se pueden definir en la pestaña *SimpleType* del ayudante de entrada Detalles.



Nota: en el ayudante de entrada Detalles puede especificar el espacio de nombres de la expresión XPath (en el atributo `xpathDefaultNamespace`). Para más información, siga leyendo.

Uso del atributo `xpathDefaultNamespace`

El espacio de nombres predeterminado declarado en el documento de esquema es el espacio de nombres predeterminado del documento de esquema. Afecta a los nombres de elemento sin prefijo del documento pero no a los nombres de elemento sin prefijo de las expresiones XPath del documento de esquema.

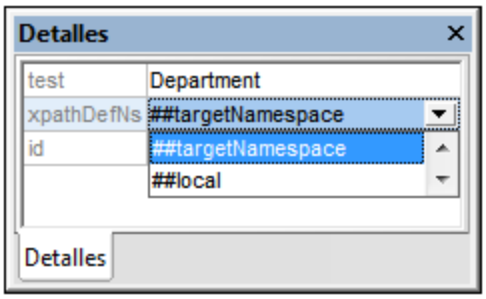
El atributo `xpathDefaultNamespace` es una característica nueva de la especificación XSD 1.1 y es el mecanismo utilizado para especificar el espacio de nombres al que pertenecen los nombres de elemento sin prefijo de las expresiones XPath.

El ámbito de los atributos `xpathDefaultNamespace` son los elementos del esquema XML en los que se decararon. El atributo `xpathDefaultNamespace` puede aparecer en estos elementos de XML Schema 1.1:

- `xs:schema`
- `xs:assert` y `xs:assertion`
- `xs:alternative`
- `xs:selector` y `xs:field` (en restricciones de identidad)

El atributo `xpathDefaultNamespace` de `xs:schema` se configura (en el modo XSD 1.1) en el cuadro de diálogo "Configuración del esquema" (**Diseño de esquemas | Configuración del esquema**). El atributo `xpathDefaultNamespace` de los demás elementos se configura en el ayudante de entrada Detalles de cada elemento (*imagen siguiente*).

© 2016-2022 Altova GmbH



Al declararse el espacio de nombres predeterminado de la XPath en `xs:schema`, se declara el espacio de nombres predeterminado de la XPath para el ámbito de todo el esquema. Puede invalidar esta declaración en los elementos en los que se permite el atributo `xpathDefaultNamespace` (*véase la lista anterior*).

En lugar de incluir un espacio de nombres real, el atributo `xpathDefaultNamespace` incluye una de estas tres palabras clave:

- `##targetNamespace`: el espacio de nombres predeterminado de la XPath será el mismo que el espacio de nombres de destino del esquema
- `##defaultNamespace`: el espacio de nombres predeterminado de la XPath será el mismo que el espacio de nombres predeterminado del esquema
- `##local`: sin espacio de nombres predeterminado de XPath

Si en el documento no se declara ningún espacio de nombres predeterminado de XPath, los elementos sin prefijo de las expresiones XPath no estarán en ningún espacio de nombres.

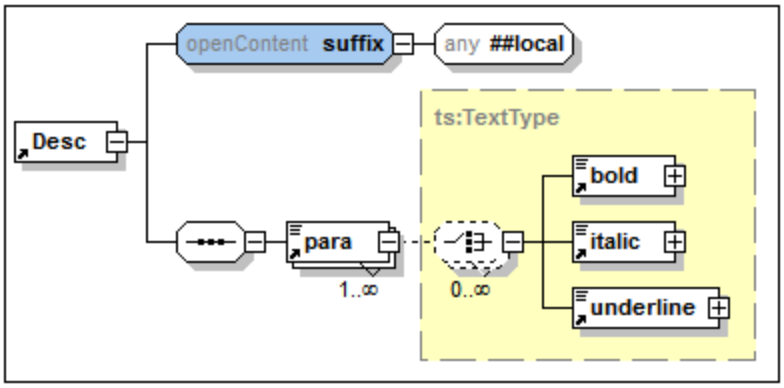
Nota: la declaración de espacio de nombres predeterminado de XPath no afecta a los atributos.

4.4.3.4 Modelos de contenido abierto (openContent)

Los modelos de contenido abierto son una característica nueva de la especificación XSD 1.1. Se declaran en los tipos complejos y permiten que cualquier elemento (es decir, un elemento sin definir del modelo de contenido de un tipo complejo) aparezca tantas veces como se desee bien (i) entre los elementos definidos en el modelo de contenido o bien (ii) después de los elementos definidos en el modelo de contenido.

El elemento `openContent` es un secundario del tipo complejo y aparece una sola vez antes del modelo de contenido del tipo complejo (*imagen siguiente*).

Modo (atributo **mode**)



Estos tres valores tienen implicaciones diferentes:

- Si `mode="interleave"` o `mode="suffix"`, entonces debe estar presente un elemento comodín (`xs:any`) sin un número máximo ni mínimo de apariciones. Esto significa que puede usar tantos elementos sin definir (comodines) como quiera.
- Si `mode="interleave"`, puede usar tantos elementos sin definir como quiera antes o después de los elementos definidos del modelo de contenido. Es decir, se intercalan entre los elementos definidos.
- Si `mode="suffix"`, puede usar tantos elementos sin definir como quiera después del último elemento definido del modelo de contenido.
- Si `mode="none"`, no puede aparecer ningún elemento sin definir y el modelo de contenido no es abierto. El valor `none` se utiliza para reemplazar el elemento `defaultOpenContent` 182, cuyo ámbito es todo el esquema.

En la vista del modelo de contenido puede añadir `openContent` como secundario de un tipo complejo (con el comando **Agregar secundario** del menú contextual). Especifique el modo haciendo doble clic en el recuadro de `openContent` (*imagen anterior*) y seleccionando un valor (`interleave`, `suffix` o `none`). También puede seleccionar el valor en el ayudante de entrada Detalles.

Propiedades comodín **(xs:any)**

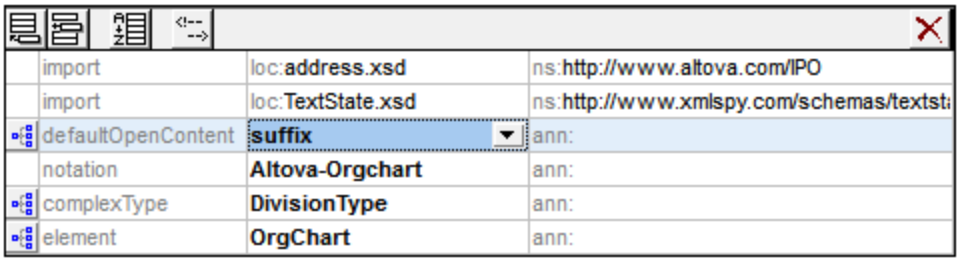
Las propiedades comodín se especifican en el ayudante de entrada Detalles del comodín. Seleccione el comodín en el diagrama e introduzca los valores de sus propiedades en el ayudante de entrada Detalles.

Contenido abierto predeterminado (**defaultOpenContent**)

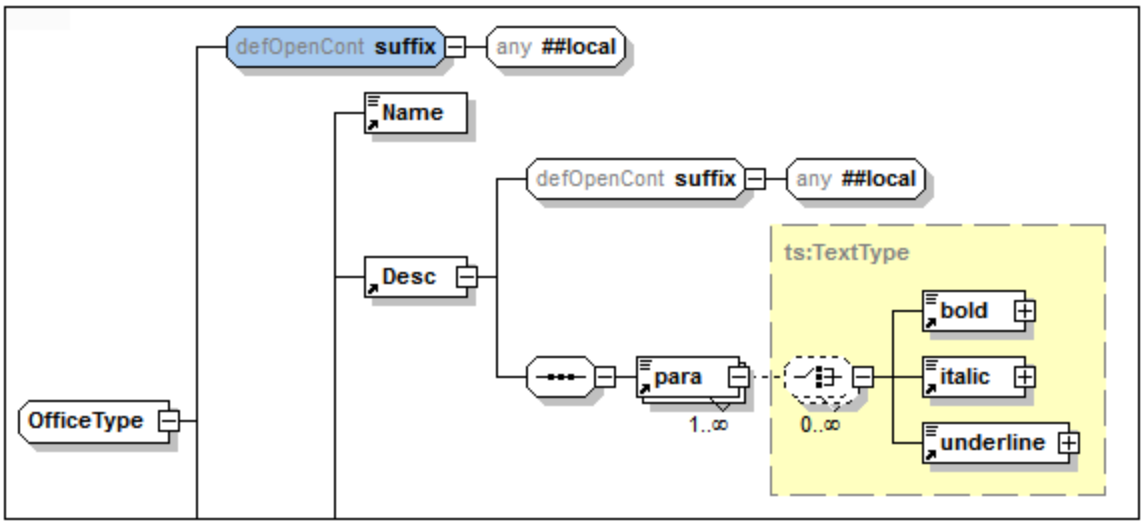
El elemento `defaultOpenContent` es nuevo en la especificación XSD 1.1 y especifica que se pueden añadir elementos sin definir a cualquier tipo complejo de contenido mixto o de solo elemento. Es similar al elemento `openContent` (también nuevo en la especificación XSD 1.1), pero la principal diferencia entre ellos es que, mientras que el elemento `openContent` se aplica a un solo tipo complejo, el elemento `defaultOpenContent` se aplica a todos los tipos complejos del esquema.

El elemento `defaultOpenContent` es un componente global 182 y aparece una sola vez en el documento (*imagen siguiente*), después de los elementos `include`, `import`, `redefine` y `override` y antes de las definiciones de los componentes. Tiene un atributo llamado `mode`, cuyo valor puede ser `interleave` o `suffix`. Su valor predeterminado es `interleave`.

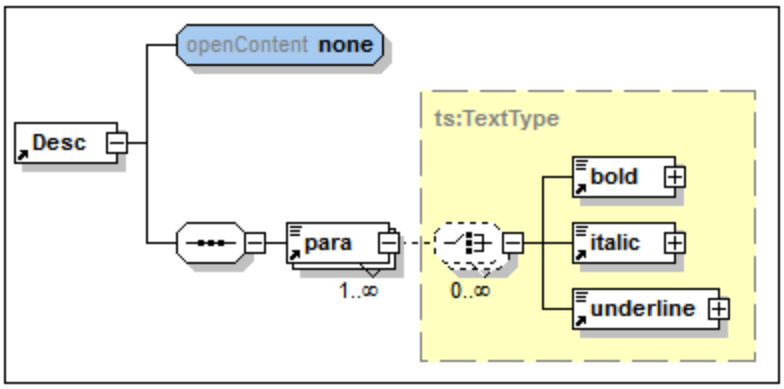
© 2016-2022 Altova GmbH



El elemento `defaultOpenContent` tiene un modelo de contenido, que se puede editar en la vista del modelo de contenido igual que el elemento `openContent` (*ver más arriba*). Una vez declarado, el elemento `defaultOpenContent` se aplica automáticamente a todos los tipos complejos del esquema y aparece en sus modelos de contenido. En la imagen siguiente, puede observar que el elemento `defaultOpenContent` se aplicó automáticamente a los tipos complejos `OfficeType` y `Desc`.

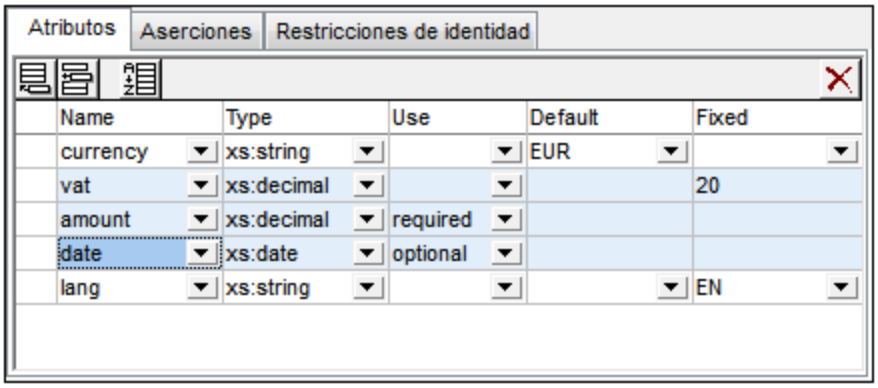


Si desea invalidar un elemento `defaultOpenContent` en un tipo complejo predeterminado, añada un elemento secundario `openContent` al tipo complejo. En la imagen siguiente, por ejemplo, el elemento `Desc` tiene el elemento `defaultOpenContent`, pero se le añadió un elemento `openContent` que invalida al elemento `defaultOpenContent`.



4.4.4 Atributos, aserciones y restricciones de identidad

El panel Atributos/Aserciones/Restricciones de identidad (AARID) (*imagen siguiente*) está situado bajo el panel principal de la vista general del esquema y de la vista del modelo de contenido. El panel y sus pestañas son fijos, pero en la vista del modelo de contenido puede visualizar y editar los componentes de cada pestaña por separado en el diagrama de la vista del modelo de contenido, en lugar de en el panel AARID. Cuando las tres pestañas se representan en el diagrama del modelo de contenido, el panel AARID desaparece.



Para habilitar/deshabilitar la representación de las pestañas del panel AARID en el diagrama, use el cuadro de diálogo Configurar visualización del esquema[1378] (**Diseño de esquemas | Configurar opciones de vista | Elemento**) o haga clic en el icono correspondiente de la barra de herramientas Diseño de esquemas (*ver lista*).

	Mostrar atributos en diagrama: disponible en la vista del modelo de contenido solamente. Habilita/deshabilita la representación de los atributos en el diagrama, en lugar de en el panel AARID.
	Mostrar aserciones en diagrama: disponible en la vista del modelo de contenido solamente. Habilita/deshabilita la representación de las aserciones en el diagrama, en lugar de en el panel AARID.
	Mostrar restricciones en diagrama: disponible en la vista del modelo de contenido solamente. Habilita/deshabilita la representación de las restricciones de identidad en el diagrama, en lugar de en el panel AARID.

Uso de las pestañas del panel AARID

Las pestañas del panel AARID se habilitan o deshabilitan dependiendo del componente seleccionado en el panel superior de la vista general del esquema o de la vista del modelo de contenido. Por ejemplo, a los complexType se les puede añadir atributos. Por eso, si selecciona un complexType en el panel principal, la pestaña *Atributos* está habilitada en el panel AARID.

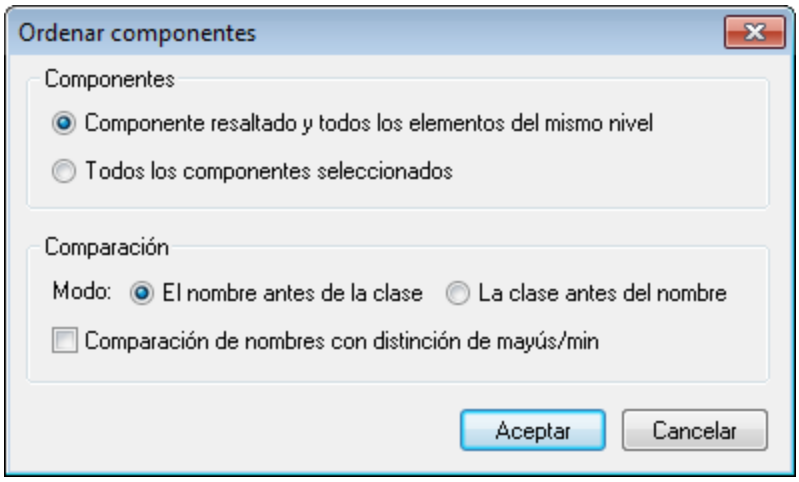
Para aprender a utilizar las pestañas del panel, consulte estos subapartados:

- Atributos, grupos de atributos y comodines de atributo[213]
- Aserciones[216]
- Restricciones de identidad[220]

© 2016-2022 Altova GmbH

Ordenar atributos y restricciones de identidad

Puede ordenar los atributos y las restricciones de identidad de las pestañas haciendo clic en el icono **Ordenar** de la barra de herramientas de la pestaña. Aparece el cuadro de diálogo "Ordenar componentes" (*imagen siguiente*), donde puede elegir si se ordenan el componente resaltado y los componentes del mismo nivel o si se ordenan todos los componentes seleccionados. En el ejemplo de la imagen anterior, se seleccionaron tres atributos (resaltados en azul). Puede mantener pulsada la tecla **Mayús** mientras hace clic en los componentes para seleccionar un grupo de componentes y pulsar la tecla **Ctrl** mientras hace clic para añadir más componentes al grupo seleccionado.



Después de seleccionar los componentes que desea ordenar, elija si se ordenan por orden alfabético (*El nombre antes de la clase*) o según el tipo de componente y por orden alfabético (*La clase antes del nombre*). El criterio de ordenación se implementa inmediatamente en el texto del esquema de documento. No es una mera máscara de la interfaz.

4.4.4.1 Atributos, grupos de atributos y comodines de atributo

En la pestaña *Atributos* (*imagen siguiente*) del panel Atributos/Aserciones/Restricciones de identidad (AARID) puede:

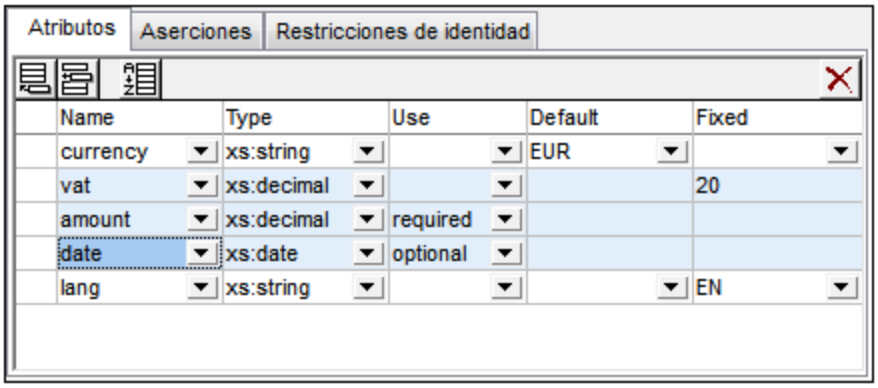
- Declarar atributos localmente en el complexType seleccionado 214
- Hacer referencia a grupos de atributos para usarlos en el complexType seleccionado 215
- Definir comodines de atributo en el complexType seleccionado 215

Nota: si habilitó la representación de atributos en el diagrama del modelo de contenido (**Diseño de esquemas | Configurar opciones de vista**), en lugar de en el panel AARID, puede editar las propiedades de los atributos, de los grupos de atributos y de los comodines de atributo en el diagrama del modelo y en el ayudante de entrada Detalles.

	*Mostrar atributos en diagrama:* disponible en la vista del modelo de contenido solamente. Habilita/deshabilita la representación de los atributos en el diagrama, en lugar de en el panel AARID.

Atributos

En la pestaña *Atributos* (*imagen siguiente*) del panel Atributos/Aserciones/Restricciones de identidad (AARID) puede declarar atributos locales de elementos y de tipos complejos, así como los atributos que constituyen grupos de atributos.



Siga estos pasos para crear atributos:

1. En la vista general del esquema, seleccione el complexType o el attributeGroup para el que desea crear el atributo.
2. En la pestaña *Atributos*, haga clic en el icono **Anexar** o **Insertar** de la barra de herramientas y seleccione **attribute**.
3. En la fila que se crea para ese atributo, escriba los detalles del atributo (nombre, tipo, uso y valor predeterminado o fijo). La propiedad `name` es obligatoria y el valor predeterminado de `use` es `optional`. Las propiedades `Type`, `Default` y `Fixed` son opcionales.

Nota: solamente se pueden añadir atributos a los attributeGroup desde la vista general del esquema[171]. A los complexType se les puede añadir desde la vista general del esquema[171] y desde la vista del modelo de contenido[171].

Nota: si un atributo tiene una definición de tipo, puede llegar a ella haciendo clic en el comando del menú contextual **Ir a la definición del tipo**. En el caso de los tipos simples integrados aparece una caja de texto con información sobre el tipo simple.

Valores predeterminados (default) y valores fijos (fixed)

Si se indica en la declaración de atributo, el valor fijo o predeterminado se aplica al documento de instancia cuando ese atributo no está presente en el documento de instancia. Se puede especificar el valor predeterminado o el valor fijo, pero no ambos (*ver imagen anterior*). El valor predeterminado o fijo debe ser válido con respecto al tipo de datos del atributo. Si la propiedad `use` tiene el valor `required`, entonces no se pueden usar valores predeterminados ni fijos.

Recuerde que:

- *Valores predeterminados:* solamente se inserta el valor predeterminado si falta el atributo. Si el atributo está presente y tiene un valor válido, no se inserta el valor predeterminado. Si el valor no es válido en el documento de instancia, se emite un mensaje de error.

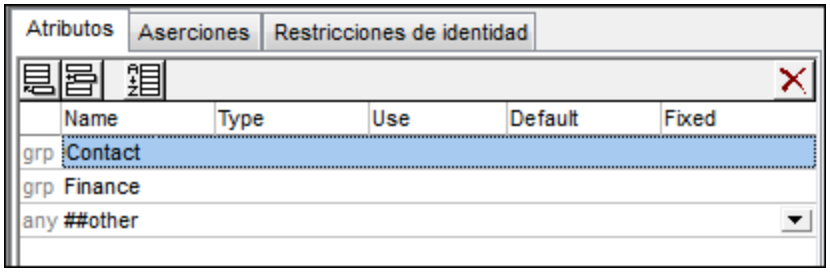
© 2016-2022 Altova GmbH

- *Valores fijos:* el valor fijo no solamente se aplica cuando falta el atributo, sino también cuando el valor que aparece en el documento de instancia no es igual al valor fijo definido en la declaración de atributo.

Nota: puede definir valores predeterminados y fijos tanto en atributos locales como globales. En los atributos locales, estos valores se pueden definir en la pestaña *Atributos* del panel AARID (*imagen anterior*) o en el ayudante de entrada Detalles. En los atributos globales, estos valores se definen en el ayudante de entrada Detalles.

Referencias a grupos de atributos (**attributeGroup**)

Si se declaró un attributeGroup global, puede añadir una referencia a este componente en la definición de un complexType. Para ello, seleccione el componente complexType en la vista general del esquema o en la vista del modelo de contenido, haga clic en el icono **Anexar** o **Insertar** de la pestaña *Atributos* del panel AARID y seleccione **attributeGroup**. En la fila del grupo de atributos que se crea, escriba el nombre del attributeGroup al que desea hacer referencia. También puede añadir varios attributeGroup (en la imagen siguiente, por ejemplo, hay dos referencias a grupos de atributos).

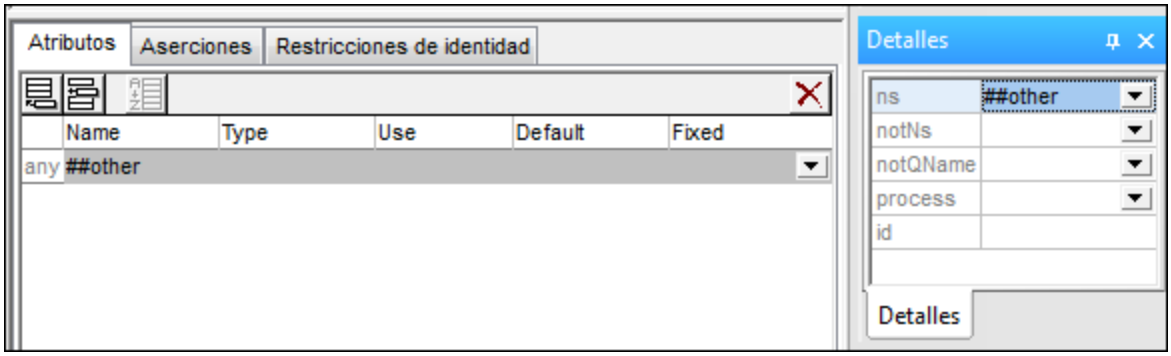


Seleccione el attributeGroup en la pestaña *Atributos* y edite sus propiedades en el ayudante de entrada Detalles.

Comodines de atributo: `anyAttribute`

Puede añadir un comodín de atributo a un complexType para permitir el uso de cualquier atributo en un elemento. Para definir un comodín de atributo se usa un elemento `anyAttribute`, que permite que en el documento de instancia aparezca cualquier atributo del espacio de nombres especificado en el elemento.

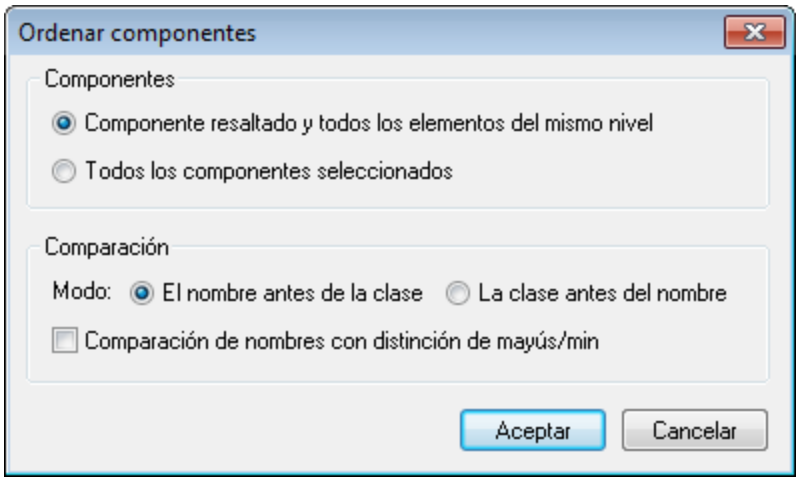
Para añadir un comodín de atributo seleccione el componente complexType en la vista general del esquema o en la vista del modelo de contenido, haga clic en el icono **Anexar** o **Insertar** de la pestaña *Atributos* del panel AARID y selecciona **anyAttribute**. Se crea una fila para el comodín de atributo `anyAttribute` (*imagen siguiente*).



En la pestaña *Atributos* puede configurar la propiedad `namespace` de `anyAttribute`. Si selecciona el comodín en la pestaña *Atributos* también puede editar otras propiedades del comodín en el ayudante de entrada Detalles (*imagen anterior*). Recuerde que las propiedades `notNamespace` y `notQName` son características de la especificación XSD 1.1 [173] y no están disponibles en el modo XSD 1.0 [173].

Ordenar atributos y grupos de atributos

Puede ordenar los atributos y los grupos de atributos de la pestaña Atributos haciendo clic en el icono **Ordenar** de la barra de herramientas de la pestaña. Aparece el cuadro de diálogo "Ordenar componentes" (*imagen siguiente*), donde puede elegir si se ordenan el componente resaltado y los componentes del mismo nivel o si se ordenan todos los componentes seleccionados. Puede mantener pulsada la tecla **Mayús** mientras hace clic en los componentes para seleccionar un grupo de componentes y pulsar la tecla **Ctrl** mientras hace clic para añadir más componentes al grupo seleccionado.



Después de seleccionar los componentes que desea ordenar, elija si se ordenan por orden alfabético (*El nombre antes de la clase*) o según el tipo de componente y por orden alfabético (*La clase antes del nombre*).

El criterio de ordenación se implementa inmediatamente en el texto del esquema de documento. No es una mera máscara de la interfaz.

Nota: los comodines de atributo no se pueden ordenar porque siempre deben aparecer al final de la declaración de tipo complejo y porque las declaraciones de tipo complejo solamente pueden tener un comodín de atributo.

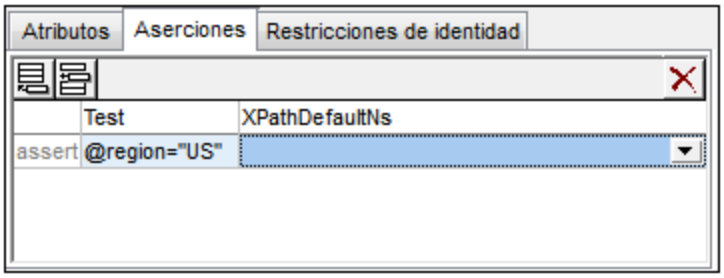
4.4.4.2 Aserciones

Las aserciones descritas en este apartado son **aserciones de tipos complejos**. Dicha aserción se define en un elemento `xs:assert` y sirve de restricción de validez del tipo complejo. (El otro tipo de aserción es una aserción de un tipo simple, que se define en un elemento `xs:assertion` y se crea y edita en el ayudante de entrada Facetas [236] de un tipo simple.)

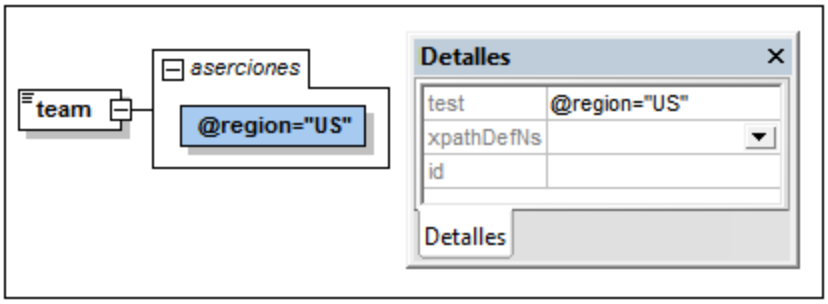
En la vista **Esquema** las aserciones de tipos complejos se pueden crear y editar de varias formas:

- *En la vista general del esquema:* en la pestaña *Aserciones* del panel Atributos/Aserciones/Restricciones de identidad (AARID) (*imagen siguiente*).

 © 2016-2022 Altova GmbH



- *En la vista del modelo de contenido:* las aserciones se pueden editar en la pestaña *Aserciones* (*imagen anterior*) o en el diagrama del modelo (*imagen siguiente*). Para elegir la opción que más le convenga, haga clic en el icono **Mostrar aserciones en diagrama** de la barra de herramientas Diseño de esquemas. En el diagrama del modelo, seleccione el recuadro aserciones del tipo complejo o del elemento de contenido complejo. Después escriba o edite la definición de la aserción en el ayudante de entrada Detalles.



	Mostrar aserciones en diagrama: disponible en la vista del modelo de contenido solamente. Habilita/deshabilita la representación de las aserciones en el diagrama, en lugar de en el panel AARID.

Nota: los elementos `assert` son una característica de la especificación XSD 1.1. Esto significa que las funciones de edición de aserciones solamente están disponibles en el modo XSD 1.1 [173].

Ámbito de la aserción

La expresión XPath utilizada para definir la restricción de la aserción debe estar dentro del ámbito del tipo complejo en el que se define el elemento `assert`. De modo que, si la expresión XPath necesita acceder a un nodo, el elemento `assert` debe definirse en un nodo antecesor de dicho nodo.

Añadir y eliminar aserciones

Un tipo complejo puede tener varias aserciones. La expresión XPath de cada aserción debe dar como resultado el valor booleano `true` para que el elemento del documento de instancia sea válido. Siga estos pasos para añadir una aserción a un tipo complejo:

- *En la vista general del esquema:* seleccione el tipo complejo. En la pestaña *Aserciones* del panel AARID (*primera imagen de este subapartado*), haga clic en el icono **Agregar** o **Insertar** de la parte superior izquierda. Puede agregar varias aserciones si quiere. Para eliminar una aserción, selecciónela y haga clic en el icono **Eliminar** de la parte superior derecha.
- *En la vista del modelo de contenido (imagen anterior):* haga clic con el botón derecho en el tipo complejo y seleccione **Agregar secundario | Aserción**. También puede hacer clic con el botón

derecho en una de las aserciones ya existentes del diagrama del tipo complejo y seleccionar **Anexar | Aserción** o **Insertar | Aserción**. Puede añadir varias aserciones si quiere. Para eliminar una aserción, selecciónela y pulse la tecla **Suprimir**.

Definir la expresión XPath de la aserción

La expresión XPath de una aserción de tipo complejo define la restricción de validez que se debe aplicar al elemento de tipo complejo en el documento de instancia. Por ejemplo, en las imágenes anteriores, la aserción está en el elemento de tipo complejo `team` y la expresión XPath de la aserción es: `@region="US"`. En el documento de esquema XML la aserción aparece así:

```
<xs:assert test='@region="US"'/>
```

La aserción indica que, en el documento de instancia, el elemento `team` debe tener un atributo `region` cuyo valor es `US`. Si no es así, el documento no es válido.

Tenga en cuenta que:

- Las expresiones XPath deben estar escritas en XPath 2.0
- Los nodos probados en la expresión XPath deben estar dentro del ámbito de la aserción (*véase más arriba*)
- Si una expresión no da como resultado el valor booleano `true/false`, el valor devuelto se convierte en un valor booleano. Las secuencias no vacías se convierten en `true` y las secuencias vacías se convierten en `false`.
- Si la expresión tiene un error de sintaxis, la expresión aparece en color rojo. Los errores de contexto no se señalizan. Por ejemplo, si la expresión XPath prueba un atributo y el atributo no está definido en el esquema, no se emite ningún error.

El mensaje de la aserción

Lo más útil es aportar una explicación de la aserción junto con su definición, para que en caso de que la aserción no se cumpla cuando se valide el documento de instancia XML, la aplicación pueda emitir el mensaje adecuado. Como la especificación XML Schema no prevé ese tipo de mensajes, XMLSpy permite suministrar un mensaje en el espacio de nombres xml-schema-extensions de Altova http://www.altova.com/es/xml-schema-extensions (o en cualquier otro espacio de nombres) junto con la definición de la aserción y usarla en la validación del documento XML de instancia. Por ejemplo:

```
<xs:assert test="count(//MiNodo) ge 1" altova:message="Se necesita un elemento MiNodo como mínimo"/> o
<xs:assertion test="count(//MiNodo) ge 1" altova:message="Se necesita un elemento MiNodo como mínimo"/>
```

Si la restricción especificada por la aserción no se cumple, además del mensaje de error de validación el motor de validación de XMLSpy emitirá un mensaje sobre la aserción a modo de sugerencia. El validador informará del valor de un atributo `assert/@message` o `assertion/@message` independientemente del espacio de nombres en el que esté el atributo `message`. No obstante, recuerde que en la vista **Esquema** solamente podrá editar atributos `message` que estén en el espacio de nombres xml-schema-extensions de Altova. Para editar atributos `message` de otros espacios de nombres debe usar la vista **Texto**.

Para más información consulte el apartado Mensajes de aserción 240.

 © 2016-2022 Altova GmbH

Uso del atributo `xpathDefaultNamespace`

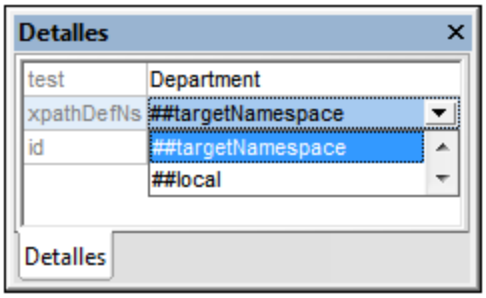
El espacio de nombres predeterminado declarado en el documento de esquema es el espacio de nombres predeterminado del documento de esquema. Afecta a los nombres de elemento sin prefijo del documento pero no a los nombres de elemento sin prefijo de las expresiones XPath del documento de esquema.

El atributo `xpathDefaultNamespace` es una característica nueva de la especificación XSD 1.1 y es el mecanismo utilizado para especificar el espacio de nombres al que pertenecen los nombres de elemento sin prefijo de las expresiones XPath.

El ámbito de los atributos `xpathDefaultNamespace` son los elementos del esquema XML en los que se decararon. El atributo `xpathDefaultNamespace` puede aparecer en estos elementos de XML Schema 1.1:

- `xs:schema`
- `xs:assert` y `xs:assertion`
- `xs:alternative`
- `xs:selector` y `xs:field` (en restricciones de identidad)

El atributo `xpathDefaultNamespace` de `xs:schema` se configura (en el modo XSD 1.1) en el cuadro de diálogo "Configuración del esquema" (**Diseño de esquemas | Configuración del esquema**). El atributo `xpathDefaultNamespace` de los demás elementos se configura en el ayudante de entrada Detalles de cada elemento (*imagen siguiente*).



Al declararse el espacio de nombres predeterminado de la XPath en `xs:schema`, se declara el espacio de nombres predeterminado de la XPath para el ámbito de todo el esquema. Puede invalidar esta declaración en los elementos en los que se permite el atributo `xpathDefaultNamespace` (*véase la lista anterior*).

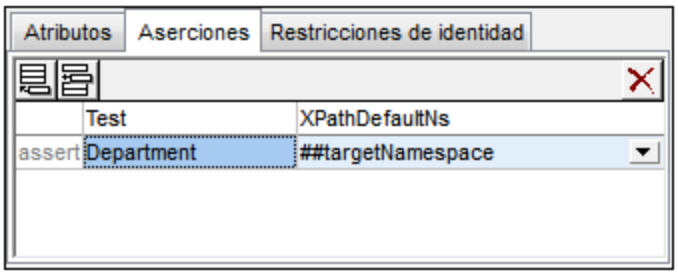
En lugar de incluir un espacio de nombres real, el atributo `xpathDefaultNamespace` incluye una de estas tres palabras clave:

- `##targetNamespace`: el espacio de nombres predeterminado de la XPath será el mismo que el espacio de nombres de destino del esquema
- `##defaultNamespace`: el espacio de nombres predeterminado de la XPath será el mismo que el espacio de nombres predeterminado del esquema
- `##local`: sin espacio de nombres predeterminado de XPath

Si en el documento no se declara ningún espacio de nombres predeterminado de XPath, los elementos sin prefijo de las expresiones XPath no estarán en ningún espacio de nombres.

Nota: la declaración de espacio de nombres predeterminado de XPath no afecta a los atributos.

Para las expresiones XPath de las aserciones también puede especificar el espacio de nombres predeterminado de la XPath en la definición de la aserción. En la pestaña *Aserciones* del panel AARID (imagen siguiente) seleccione una opción de la lista desplegable del campo *XPathDefaultNS*.



El espacio de nombres seleccionado estará en el ámbito de la aserción.

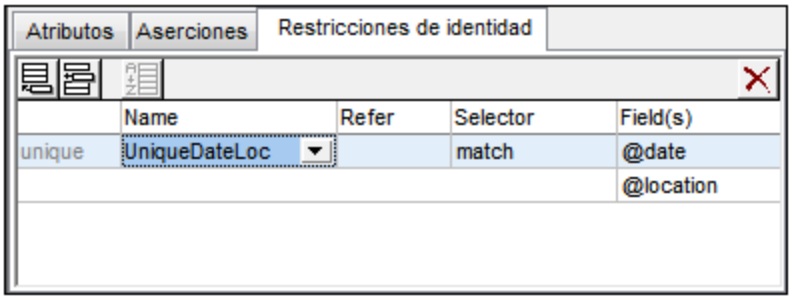
4.4.4.3 Restricciones de identidad

Las restricciones de identidad (IDC) se pueden definir en declaraciones de elementos locales o globales. Estas restricciones especifican el carácter único de los nodos y permiten hacer referencia correctamente a los nodos únicos.

Mecanismos de declaración

Puede definir restricciones de identidad (`unique`, `key`, `keyref`) de varias formas:

- En la vista general del esquema[177] puede declarar las restricciones en los elementos globales. Seleccione un elemento global y defina las IDC en la pestaña *Restricciones de identidad* del panel Atributos/Aserciones/Restricciones de identidad (panel AARID) (*imagen siguiente*).

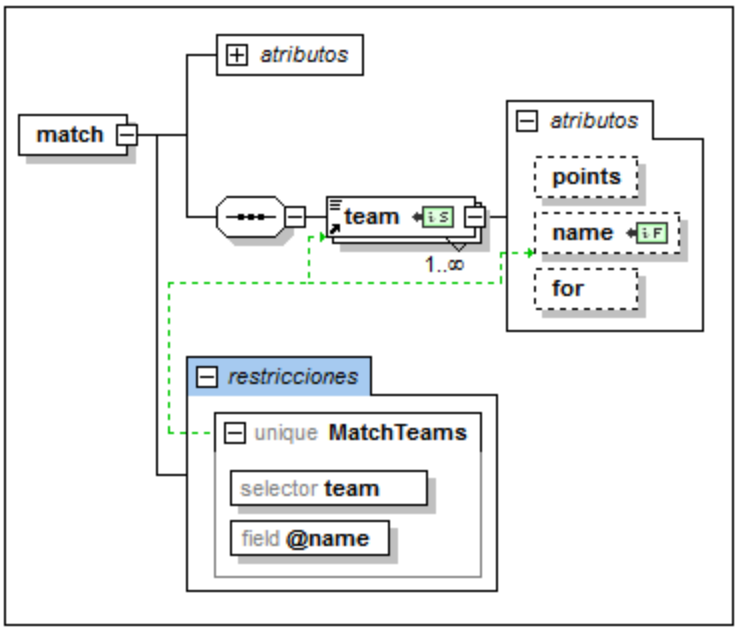


  Añada una IDC (`unique`, `key`, `keyref`) con el icono **Insertar** o **Anexar** de la pestaña *Restricciones de identidad*. Estos iconos también sirven para añadir la propiedad `field` a la restricción. Use el icono **Eliminar** para eliminar la restricción o la propiedad `field` seleccionada.

- En la vista del modelo de contenido[190] de un elemento global puede definir las IDC del elemento global o de un elemento local descendiente. En esta vista, las restricciones se pueden editar en la pestaña *Restricciones de identidad* (*imagen anterior*) o en el recuadro *restricciones* del elemento del diagrama (observe que en la imagen siguiente el elemento `match` tiene una restricción de unicidad con un selector `team`). Para editar las IDC en el diagrama del elemento directamente, seleccione la opción pertinente en el cuadro de diálogo "Configurar visualización del esquema" (**Diseño de esquemas | Configurar opciones de vista**). O puede hacer clic en el icono **Mostrar restricciones en diagrama**

 © 2016-2022 Altova GmbH

de la barra de herramientas Diseño de esquemas. A partir de ese momento, el diagrama incluye una representación gráfica de las IDC.



Para añadir una IDC (`unique`, `key`, `keyref`) en el diagrama cuando las restricciones están representadas en el diagrama directamente, haga clic con el botón derecho en el elemento que desea restringir y seleccione **Agregar secundario | [IDC]** en el menú contextual. El elemento `field` solamente está disponible en el menú contextual si seleccionó una IDC en el diagrama. Para eliminar la propiedad `field` o la IDC, selecciónelas y pulse la tecla **Suprimir**.

La expresión XPath se puede introducir en el recuadro `selector` y en el recuadro `field` de tres maneras diferentes: (i) escribiéndola directamente, (ii) seleccionando el nodo en la lista desplegable que aparece al hacer clic en el recuadro `selector` o `field`, o (iii) arrastrando el nodo de destino hasta el recuadro `selector` o `field` y soltándolo cuando el recuadro aparezca resaltado. La expresión XPath se crea automáticamente.

Nota: en la pestaña *Restricciones de identidad* del ayudante de entrada Componentes se ofrece un resumen de todas las restricciones de identidad[229] del esquema.

Iconos relacionados con las restricciones de identidad

Mostrar restricciones en diagrama: disponible en la vista del modelo de contenido solamente. Habilita/deshabilita la representación de las restricciones de identidad en el diagrama, en lugar de en el panel AARID.

Mostrar restricciones de identidad: disponible en la vista del modelo de contenido solamente. Habilita/deshabilita la información sobre las IDC.

Nodo selector, nodo field: aparecen en los recuadros del diagrama e identifican el nodo seleccionado (en la IDC) por la expresión XPath de `selector` y de `field` (respectivamente).

Representación de las restricciones de identidad (IDC)

Si está activado el icono **Mostrar restricciones de identidad** IDC, la información sobre las IDC aparece en el diagrama y se puede ver mejor. Los nodos seleccionados por las expresiones XPath de `selector` y `field` se señalan con unos pequeños iconos (*ver la lista anterior*) y el recuadro de la IDF está conectado al recuadro de sus selector y field por medio de líneas de color verde (*imagen anterior*).

El icono **Mostrar restricciones de identidad** también activa la función de validación de IDC de la vista Esquema. Si una expresión XPath o la restricción IDC no es correcta, los elementos erróneos se indican en color rojo, mientras que las advertencias se señalan en color naranja. Cuando se valida el documento de esquema XML, los mensajes de error y advertencia aparecen en la ventana Mensajes.

Para deshabilitar esta función de validación, haga clic otra vez en el icono **Mostrar restricciones de identidad** IDC.

Ejemplo de código XML

Los ejemplos de restricciones de identidad que aparecen más abajo en este apartado están basados en este documento de instancia válido:

```
<results xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="Scores.xsd">
   <!----- Grupos ----->
   <group id="A">
          <team name="Brazil"/>
          <team name="Germany"/>
          <team name="Italy"/>
          <team name="Holland"/>
   </group>
   <group id="B">
          <team name="Argentina"/>
          <team name="France"/>
          <team name="England"/>
          <team name="Spain"/>
   </group>
   <!----- Partidos ----->
   <match group="A" date="2012-06-12" location="Munich">
          <team name="Brazil" for="2" points="3"/>
          <team name="Germany" for="1" points="0"/>
   </match>
   <match group="A" date="2012-06-12" location="Frankfurt">
          <team name="Italy" for="2" points="1"/>
          <team name="Holland" for="2" points="1"/>
   </match>
   <match group="B" date="2012-06-13" location="Munich">
          <team name="Argentina" for="2" points="3"/>
          <team name="France" for="0" points="0"/>
   </match>
   <match group="B" date="2012-06-13" location="Berlin">
          <team name="England" for="0" points="1"/>
```


© 2016-2022 Altova GmbH

```
            <team name="Spain" for="0" points="1"/>
        </match>
    </results>
```

Restricciones de unicidad (`unique`)

Una restricción de unicidad especifica que el valor de un elemento o atributo (o de un conjunto de elementos o atributos) debe ser único dentro de un ámbito determinado. En el código XML del ejemplo, queremos asegurarnos de que los dos equipos que se enfrentan en un partido no son el mismo equipo. De modo que, dentro del ámbito de cada elemento `match`, restringimos el valor de los nodos `team/@name` como valor único. Para hacer esto es necesario seguir estos pasos:

1. En la vista general del esquema, seleccione el elemento `match`. El elemento `match` será el ámbito de la definición de restricción de identidad.
2. En la pestaña *Restricciones de identidad*, haga clic en el icono **Anexar** o **Insertar** de la parte superior izquierda. Haga clic en **Unique** en el menú que aparece. Se añade una fila para la restricción de unicidad (*imagen siguiente*).

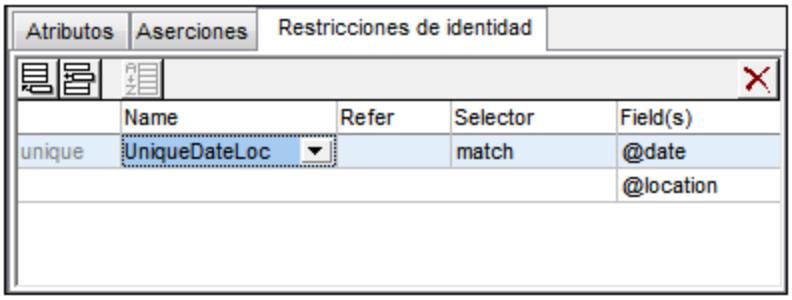


3. Ponga un nombre a la restricción de unicidad. (En el ejemplo de la imagen se usó el nombre `MatchTeams`.)
4. Introduzca una expresión XPath en el campo *Selector* para seleccionar el elemento `team`. Recuerde que el elemento `match` es el nodo de contexto (el ámbito). Ahora el elemento `team` es el selector de la IDC, es decir, el nodo al que afecta la restricción de unicidad.
5. En el campo *Field*, escriba el nodo `@name` que debe ser único. El valor de este nodo es el valor que debe ser único.

La restricción de unidad que acabamos de describir especifica que dentro del ámbito de cada elemento `match`, cada elemento `team` debe tener un valor único en el atributo `@name`.

Puede usar el resto de los campos de la pestaña para comprobar la unicidad. Por ejemplo, puede definir una restricción de unicidad en el elemento `results` para comprobar si todos los partidos tienen una combinación única de fecha y lugar: no puede haber más de un partido en un mismo lugar en la misma fecha. La restricción de unicidad debe tener, por cada elemento `match` (el selector), una combinación única de valores para los atributos `@date` y `@location` dentro del ámbito del elemento `results`.

Defina la restricción de unicidad en el elemento `results` tal y como hizo en el paso anterior. En este caso el *Selector* es `match` y en la columna *Field* introducimos `@date` y `@location` (*imagen siguiente*). Para añadir otro elemento en la columna Field, haga clic en el icono **Anexar** y después seleccione **Field**.

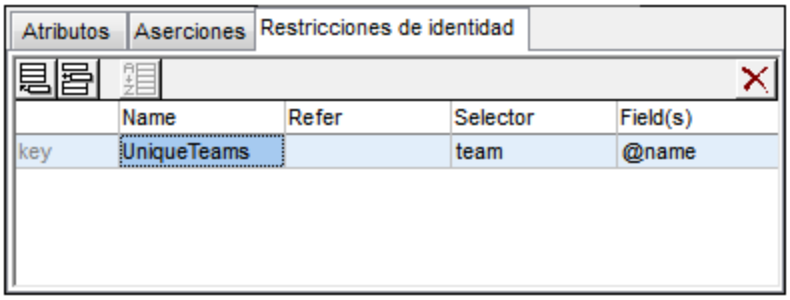


Nota: la columna *Refer* de la pestaña *Restricciones de identidad* se habilita solamente en las restricciones de identidad `keyref`.

Restricciones de clave (`key`)

Una restricción de clave `key` especifica que (i) el valor de un elemento o atributo (o de un conjunto de elementos o atributos) debe ser único dentro de un ámbito determinado y que (ii) estos elementos o atributos deben estar presentes en el documento de instancia XML. Por tanto, los elementos o atributos opcionales no se deben seleccionar como `field` de una restricción `key`. En otras palabras, una restricción de clave `key` es una restricción de unicidad (`unique`) con una restricción adicional: que los atributos o elementos seleccionados como `field` deben estar presentes a la fuerza en el documento de instancia XML.

En la imagen siguiente aparece una restricción de clave `key` definida en un elemento `match`, similar a la primera restricción de unicidad del ejemplo.



Esta restricción de clave `key` especifica que dentro del ámbito de cada elemento `match`, cada elemento `team` debe tener un valor único para el atributo `@name`. Además, especifica que el atributo `@name` debe estar presente en cada elemento `match/team`.

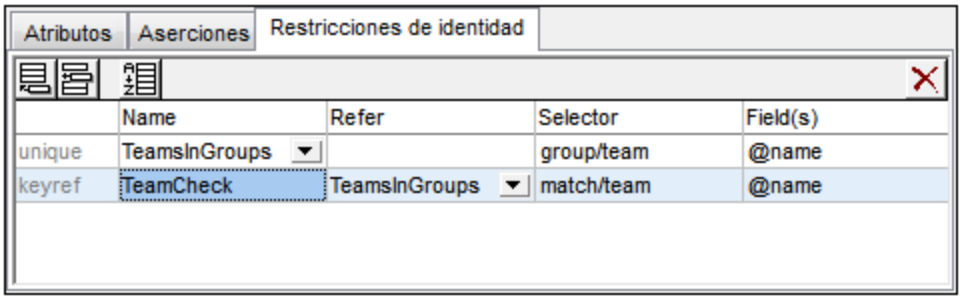
Nota: la columna *Refer* de la pestaña *Restricciones de identidad* se habilita solamente en las restricciones `keyref`.

Referencias de clave (`keyref`)

Las referencias de clave comparan un conjunto de valores de un documento de instancia con otro conjunto de valores. En nuestro código XML de ejemplo (*ver más arriba*) podemos usar una referencia de clave para comprobar si los equipos que juegan partidos se enumeran en la lista de grupos. Si no es así, el documento XML no es válido.

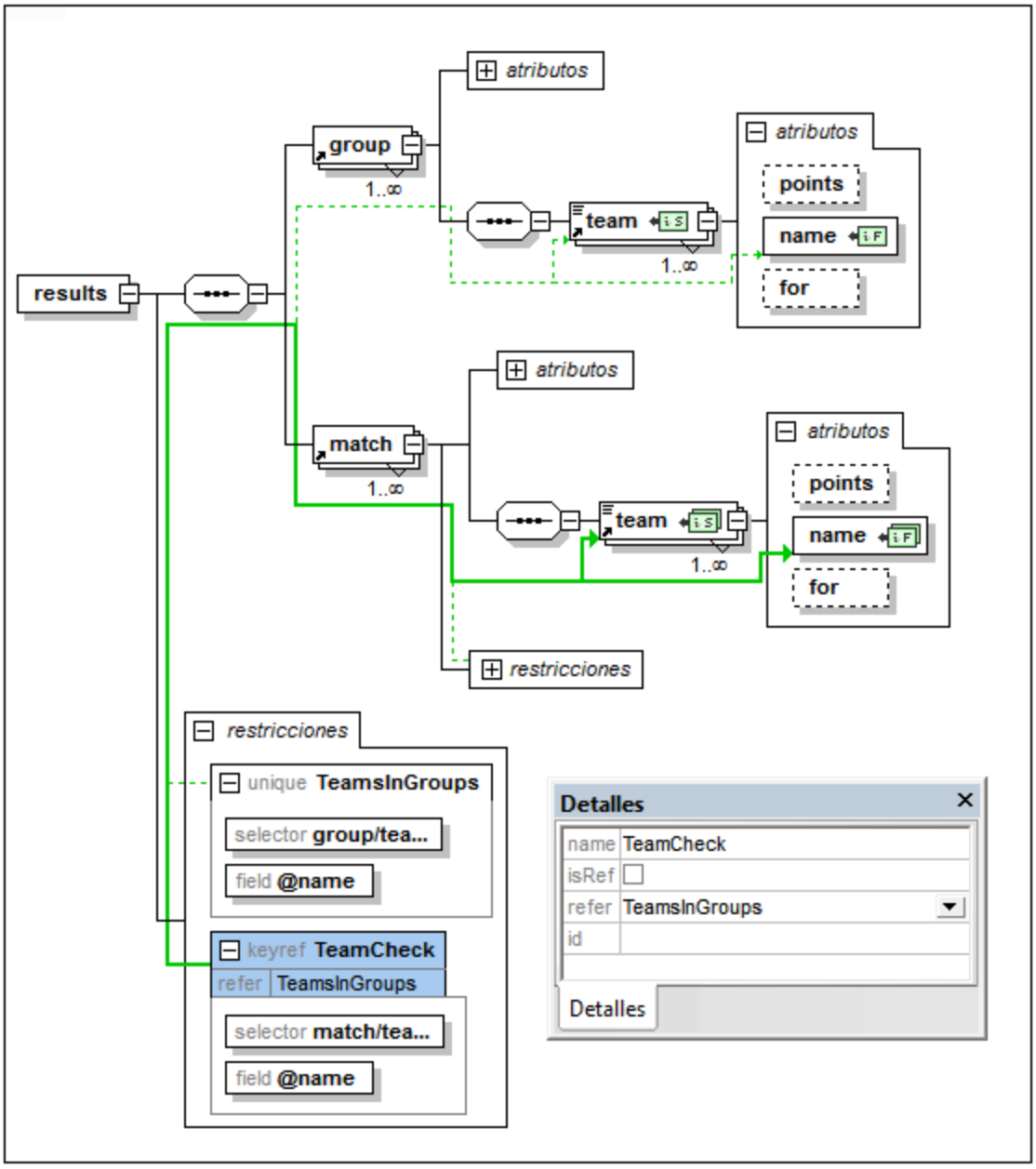
Primero creamos una constante de unicidad o una restricción de clave `key`. En la imagen siguiente aparece la restricción de unicidad (`unique`) `TeamsInGroups`, que se creó en el elemento `results`. Esta restricción estipula que cada elemento `team` de `group` debe tener un atributo `@name` único.

 © 2016-2022 Altova GmbH



Después creamos la referencia de clave (`keyref`) `TeamCheck`, que selecciona el secundario `team` del elemento `match` y comprueba si el valor del atributo `@name` está presente entre los valores devueltos por `TeamsInGroups`, al que hace referencia (en la columna *Refer*).

En la imagen siguiente puede ver una representación gráfica de esta referencia de clave (resaltada en azul) junto con el ayudante de entrada Detalles (donde puede seleccionar también la restricción a la que se hace referencia). Las relaciones de la restricción seleccionada aparecen señaladas con una línea sólida de color verde. Las restricciones no seleccionadas se representan con una línea de puntos verdes. Además, por cada restricción de identidad, los nodos seleccionados por la expresión XPath para `selector` y `field` se identifica con los iconos y respectivamente. Si el nodo está contraído, la línea de relación termina en puntos suspensivos.



Uso del atributo `xpathDefaultNamespace`

El espacio de nombres predeterminado declarado en el documento de esquema es el espacio de nombres predeterminado del documento de esquema. Afecta a los nombres de elemento sin prefijo del documento pero no a los nombres de elemento sin prefijo de las expresiones XPath del documento de esquema.

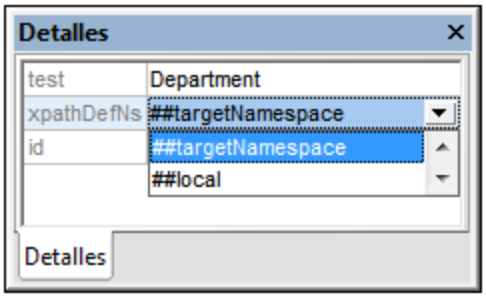
El atributo `xpathDefaultNamespace` es una característica nueva de la especificación XSD 1.1 y es el mecanismo utilizado para especificar el espacio de nombres al que pertenecen los nombres de elemento sin prefijo de las expresiones XPath.

El ámbito de los atributos `xpathDefaultNamespace` son los elementos del esquema XML en los que se decararon. El atributo `xpathDefaultNamespace` puede aparecer en estos elementos de XML Schema 1.1:

© 2016-2022 Altova GmbH

- `xs:schema`
- `xs:assert` y `xs:assertion`
- `xs:alternative`
- `xs:selector` y `xs:field` (en restricciones de identidad)

El atributo `xpathDefaultNamespace` de `xs:schema` se configura (en el modo XSD 1.1) en el cuadro de diálogo "Configuración del esquema" (**Diseño de esquemas | Configuración del esquema**). El atributo `xpathDefaultNamespace` de los demás elementos se configura en el ayudante de entrada Detalles de cada elemento (*imagen siguiente*).



Al declararse el espacio de nombres predeterminado de la XPath en `xs:schema`, se declara el espacio de nombres predeterminado de la XPath para el ámbito de todo el esquema. Puede invalidar esta declaración en los elementos en los que se permite el atributo `xpathDefaultNamespace` (*véase la lista anterior*).

En lugar de incluir un espacio de nombres real, el atributo `xpathDefaultNamespace` incluye una de estas tres palabras clave:

- `##targetNamespace`: el espacio de nombres predeterminado de la XPath será el mismo que el espacio de nombres de destino del esquema
- `##defaultNamespace`: el espacio de nombres predeterminado de la XPath será el mismo que el espacio de nombres predeterminado del esquema
- `##local`: sin espacio de nombres predeterminado de XPath

Si en el documento no se declara ningún espacio de nombres predeterminado de XPath, los elementos sin prefijo de las expresiones XPath no estarán en ningún espacio de nombres.

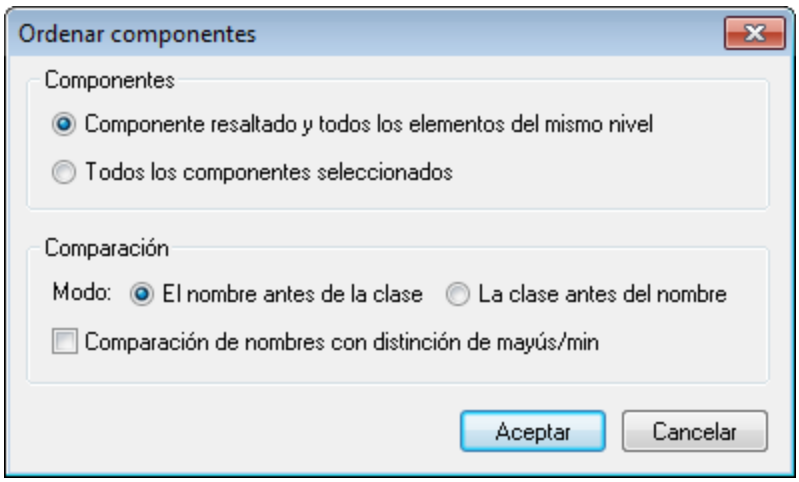
Nota: la declaración de espacio de nombres predeterminado de XPath no afecta a los atributos.

Id. de la restricciones de identidad

Puede asignar un Id. a la restricción de identidad, a su `selector` y a su `field`. Para asignar un Id., seleccione el componente y después escriba el identificador en el campo `id` del ayudante de entrada Detalles.

Ordenar restricciones de identidad

Puede ordenar las restricciones de la pestaña Restricciones de identidad haciendo clic en el icono **Ordenar** de la barra de herramientas de la pestaña. Aparece el cuadro de diálogo "Ordenar componentes" (*imagen siguiente*), donde puede elegir si se ordenan el componente resaltado y los componentes del mismo nivel o si se ordenan todos los componentes seleccionados. Puede mantener pulsada la tecla **Mayús** mientras hace clic en los componentes para seleccionar un grupo de componentes y pulsar la tecla **Ctrl** mientras hace clic para añadir más componentes al grupo seleccionado.



Después de seleccionar los componentes que desea ordenar, puede elija si se ordenan por orden alfabético (*El nombre antes de la clase*) o según el tipo de componente y por orden alfabético (es decir, las restricciones `unique` primero, después las restricciones `key` y por último las `keyref`).

El criterio de ordenación se implementa inmediatamente en el texto del esquema.

4.4.5 Ayudantes de entrada de la vista Esquema

La vista Esquema ofrece tres ayudantes de entrada, que describimos en detalle en este apartado:

- Ayudante de entrada Componentes 229
- Ayudante de entrada Detalles 232
- Ayudante de entrada Facetas 234

Los ayudantes de entrada son los mismos en la vista general del esquema y en la vista del modelo de contenido. Con ellos podrá agregar y editar definiciones de componentes del esquema de forma gráfica. Por ejemplo, puede arrastrar componentes desde un ayudante de entrada o seleccionar un componente en el diseño y definir sus propiedades en uno de los ayudantes.

© 2016-2022 Altova GmbH

4.4.5.1 Componentes

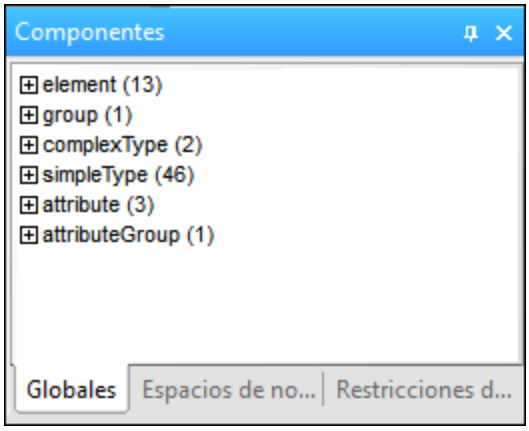
Con el ayudante de entrada Componentes de la vista Esquema puede:

- Organizar los componentes globales en una vista en forma de árbol, según el tipo de componente y su espacio de nombres (*ver las imágenes de este subapartado*). Ofrece un resumen organizado de todos los componentes globales.
- Navegar hasta la vista del modelo de contenido de un componente global y visualizarla (siempre que el componente tenga un modelo de contenido). Si el componente no tiene un modelo de contenido, simplemente se resalta el componente en la vista general del esquema. Los componentes globales que se incluyeron o importaron desde otros esquemas también aparecen en el ayudante de entrada Componentes.
- Ver un resumen de las restricciones de identidad definidas en el documento de esquema. Para más información sobre la pestaña *Restricciones de identidad* del ayudante de entrada Componentes, consulte el apartado Restricciones de identidad 220.

Nota: dependiendo del modo XSD seleccionado (XSD 1.0 o 1.1) el ayudante de entrada muestra los tipos de datos integrados de XSD 1.0 o de XSD 1.1.

Pestaña Globales

La pestaña *Globales* (*imagen siguiente*) del ayudante de entrada ofrece una lista en forma de árbol de los componentes globales, agrupados según el tipo de componente. Junto al tipo de componente aparece el número de componentes globales de ese tipo presentes en el esquema.

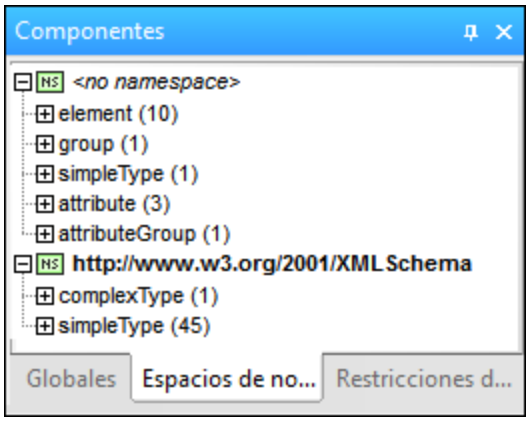


En la lista en forma de árbol, los componentes globales se agrupan en siete categorías. Recuerde que si el esquema no incluye ningún componente global de un tipo determinado, ese tipo de componente no aparece en la lista en forma de árbol.

- Declaraciones de elemento (`element`)
- Modelos de grupo (`group`)
- Tipos complejos (`complexType`)
- Tipos simples (`simpleType`)
- Declaraciones de atributo (`attribute`)
- Grupos de atributos (`attributeGroup`)
- Notaciones (`notation`)

Pestaña Espacios de nombres

En la pestaña *Espacios de nombres* (*imagen siguiente*), los componentes se organizan según su espacio de nombres y después por tipo de componente.



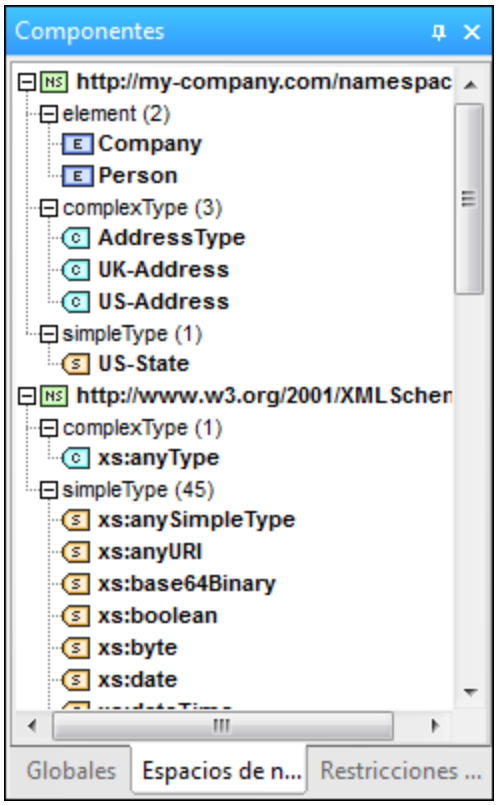
En la lista en forma de árbol de esta pestaña, los componentes globales se agrupan en siete categorías. Recuerde que si el esquema no incluye ningún componente global de un tipo determinado, ese tipo de componente no aparece en la lista en forma de árbol.

- Declaraciones de elemento (`element`)
- Modelos de grupo (`group`)
- Tipos complejos (`complexType`)
- Tipos simples (`simpleType`)
- Declaraciones de atributo (`attribute`)
- Grupos de atributos (`attributeGroup`)
- Notaciones (`notation`)

Tipos de componentes en las pestañas Globales y Espacios de nombres

Si expande un grupo de componentes en la pestaña *Globales* o *Espacios de nombres*, se muestran todos los componentes que pertenecen a ese grupo (*imagen siguiente*). Así puede navegar rápidamente hasta un componente concreto. Si hace doble clic en un componente del ayudante de entrada Componentes, su definición aparece en la ventana principal.

© 2016-2022 Altova GmbH



Si el componente tiene un modelo de contenido (es decir, si es un element, group o complexType), al hacer doble clic en el componente se abre su modelo de contenido en la vista del modelo de contenido (de la ventana principal). Si el componente no tiene un modelo de contenido (es decir, si es un simpleType, attribute, attributeGroup o notation) y hace doble clic en él, el componente aparece resaltado en la vista general del esquema (de la ventana principal).

Nota importante: si el componente está en un esquema incluido o importado, entonces se abre el esquema incluido o importado (si no está abierto ya) y se muestra su modelo de contenido en la vista del modelo de contenido o se resalta el componente en la vista general del esquema.

Pestaña Restricciones de identidad

La pestaña *Restricciones de identidad* del ayudante de entrada Componentes (*imagen siguiente*) ofrece una visión de conjunto de las restricciones de identidad del documento. En esta pestaña las restricciones se enumeran según el tipo de restricción (`unique`, `key`, `keyref`) en una lista en forma de árbol.



Los nodos que aparecen en negrita están presentes en el esquema actual, mientras que el resto son nodos de esquemas secundarios. Si hace doble clic en uno de los nodos de la pestaña *Restricciones de identidad*, el nodo aparece resaltado en la vista del modelo de contenido 190.

Si hace clic con el botón derecho en un nodo de la pestaña *Restricciones de identidad*, aparece un menú contextual con estas opciones:

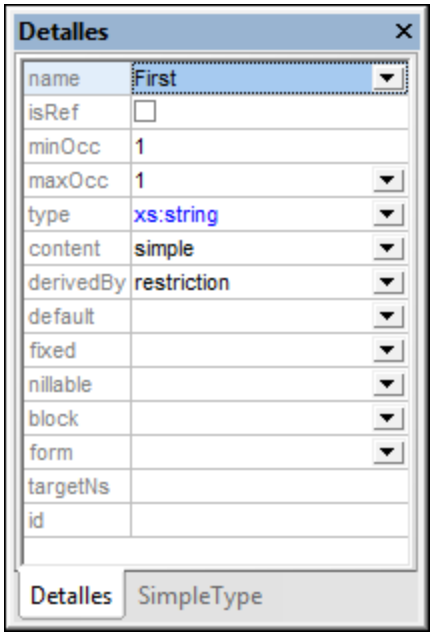
- *Mostrar en el diagrama:* selecciona el componente de esquema en la vista del modelo de contenido 190 .
- *Mostrar destino del selector/field en el diagrama:* selecciona en la vista del modelo de contenido 190 el componente de esquema de destino del selector o field de la restricción de identidad. Si la restricción tiene varios field, aparece un cuadro de diálogo solicitando que seleccione uno de ellos.
- *Ir a la restricción de identidad:* seleccione la restricción de identidad en la vista general del esquema 177.
- *Expandir/contraer todos los secundarios:* expande o contrae todo el árbol.

Para más información, consulte el apartado Restricciones de identidad 220.

4.4.5.2 Detalles

El ayudante de entrada Detalles de la vista Esquema muestra datos editables sobre el componente o compositor seleccionado en la ventana principal. Si está editando un archivo de esquema que contiene extensiones de bases de datos, el ayudante incluye una pestaña con datos sobre las extensiones de BD.

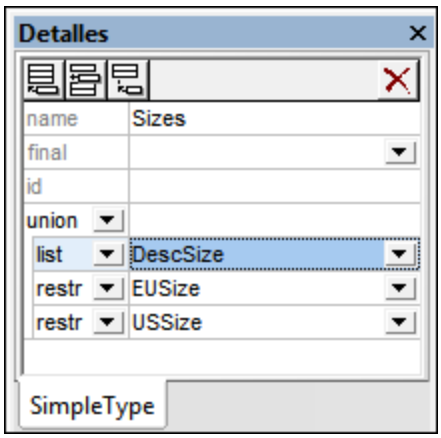
© 2016-2022 Altova GmbH



Para cambiar las propiedades del componente o compositor seleccionado, haga doble clic en el campo que desea editar y edite la propiedad. Si el campo que desea editar incluye un cuadro combinado, seleccione un valor de la lista desplegable. Los cambios realizados en el ayudante de entrada tienen efecto inmediato en el diseño.

Derivaciones de tipos simples

También puede usar el ayudante de entrada Detalles para crear tipos simples derivados: `restriction`, `list` y `union`. Si el componente seleccionado en el diseño es un tipo simple, el ayudante de entrada Detalles incluye una pestaña llamada *SimpleType* (*imagen siguiente*).



En el cuadro combinado para el tipo de derivación, seleccione `restrictiion`, `list` o `union`. Después seleccione un tipo simple en el cuadro combinado para el tipo de miembro. Use los iconos de la barra de herramientas para anexar o insertar un tipo en el mismo nivel, para agregar otro subnivel de derivación o para eliminar un tipo de derivación.

4.4.5.3 Facetas

Para crear un tipo simple (con nombre o anónimo) se restringe el tipo base del tipo simple (que es un tipo simple ya existente). Dicha restricción se lleva a cabo añadiendo facetas para restringir los valores del tipo base. El ayudante de entrada Facetas de la vista **Esquema** permite editar las facetas de un tipo simple de forma gráfica y sencilla. Por cada tipo de faceta disponible el ayudante de entrada incluye una pestaña diferente:

Pestaña	Facetas disponibles
Facetas 234	`minInclusive`, `maxInclusive`, `minExclusive`, `maxExclusive`, `length`, `minLength`, `maxLength`, `totalDigits`, `fractionDigits`, `whiteSpace`, `explicitTimezone`
Patrones 235	`pattern`
Enumeraciones 236	`enumeration`
Aserciones 236	`assertion`
Muestras 238	`altova:exampleValues` es una anotación, **no una faceta**. Esta anotación se usa para generar valores de muestra en el documento XML de instancia que XMLSpy genera a partir del esquema XML.

A continuación describimos todas estas pestañas.

Cómo seleccionar el tipo simple en el diseño

Puede seleccionar el tipo simple (con nombre o anónimo) en dos entornos diferentes:

- En la vista general del esquema 177 (bien en la lista de componentes globales, o bien en la pestaña *Atributos* del panel inferior) o
- En la vista del modelo de contenido 190 (bien en el diagrama, o bien en la pestaña Atributos del panel inferior)

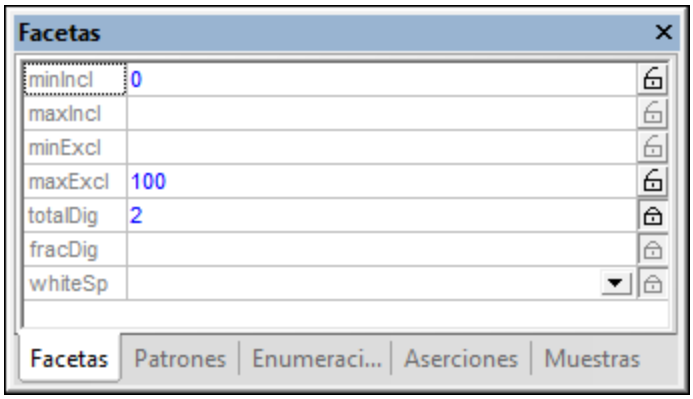
Cuando seleccione un tipo simple en el diseño (en uno de los dos entornos mencionados anteriormente), las facetas pertinentes se habilitan en el ayudante de entrada Facetas y se pueden editar.

Pestaña Facetas

En la pestaña *Facetas* solamente aparecen las facetas apropiadas para el tipo seleccionado en el diseño. Por ejemplo, si desea restringir el tipo `xs:string`, las facetas como `totalDigits` no aparecen en el ayudante de entrada porque no se le pueden aplicar al tipo seleccionado.

- Las cuatro facetas de límite (`minInclusive`, `maxInclusive`, `minExclusive`, `maxExclusive`) solamente se pueden aplicar a tipos numéricos y de fecha/hora y a tipos derivados de ellos.
- Las tres facetas de longitud (`length`, `minLength`, `maxLength`) solamente se pueden aplicar a tipos basados en cadenas, a los tipos binarios y a `anyURI`.
- La faceta `totalDigits` se puede aplicar a `xs:decimal` y a tipos enteros y a los tipos derivados de ellos. La faceta `fractionDigits` solamente se puede aplicar a `xs:decimal`.

© 2016-2022 Altova GmbH

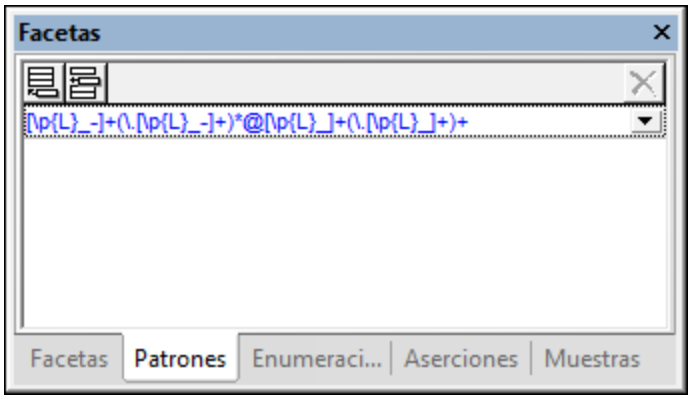


Para introducir un valor, seleccione un valor de la lista desplegable del cuadro combinado (si lo hay) o haga doble clic en el campo del valor y escriba el valor nuevo. Si introduce un valor no válido, los conflictos resultantes se señalan en color rojo. Los valores válidos aparecen en azul. Por ejemplo, no se pueden usar al mismo tiempo las facetas `minInclusive` y `maxInclusive`. Por eso, si escribe un valor en estas dos facetas, los valores aparecen en rojo.

Para especificar una **faceta fija** (dándole a la faceta un valor de atributo `fixed="true"`), haga clic en el símbolo en forma de candado que aparece a la derecha de la faceta para cerrar el candado. En la imagen anterior, por ejemplo, la faceta `totalDigits` se configuró como faceta fija. Puede haber más de una faceta fija. Para revertir la faceta a un estado no fijo, haga clic otra vez en el símbolo en forma de candado.

Pestaña Patrones

En la pestaña *Patrones* (*imagen siguiente*) puede añadir facetas `pattern` a una restricción. El patrón (de una faceta `pattern`) se especifica con una expresión regular. El patrón de la imagen siguiente, por ejemplo, especifica el patrón de las direcciones de correo electrónico.

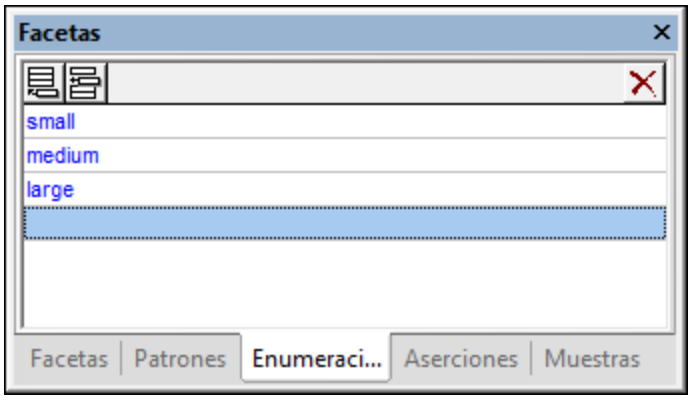


Si se especifican varias facetas `pattern`, entonces el valor de la instancia XML debe coincidir con uno de ellos. Por ejemplo, si creamos un patrón para restringir los códigos postales, el patrón podría tener dos facetas `pattern`, una para los códigos postales de EE UU y otra para los códigos postales europeos. El valor de la instancia XML debe coincidir con uno de estos dos patrones. De lo contrario el valor no es válido.

Para añadir una faceta `pattern` haga clic en el icono **Anexar** o **Insertar** de la esquina superior izquierda y escriba una expresión regular que defina el patrón. Para eliminar una faceta `pattern`, selecciónela y haga clic en el icono **Eliminar** de la esquina superior derecha.

Pestaña Enumeraciones

En la pestaña *Enumeraciones* (*imagen siguiente*) puede añadir facetas `enumeration` a una restricción. Cada faceta `enumeration` especifica un valor válido para el tipo. Un grupo de facetas `enumeration` especifica un rango de valores permitidos. En la imagen siguiente, por ejemplo, las facetas `enumeration` especifican el rango permitido de valores para la restricción.



Para añadir una faceta `enumeration` haga clic en el icono **Anexar** o **Insertar** de la esquina superior izquierda y escriba su valor. Para eliminar una faceta `enumeration`, selecciónela y haga clic en el icono **Eliminar** de la esquina superior derecha.

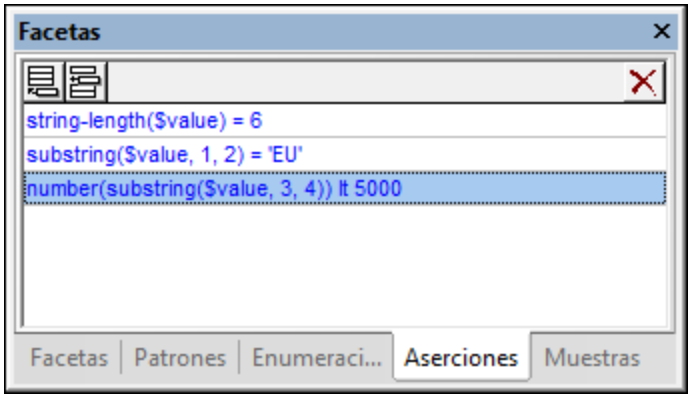
Pestaña Aserciones

Las aserciones son una característica de la especificación XSD 1.1. Esto significa que la pestaña *Aserciones* se habilita solamente en el modo XSD 1.1 173. Las facetas de aserción que se definen en la pestaña *Aserciones* del ayudante de entrada Facetas son **aserciones para tipos simples** (en lugar de aserciones para tipos complejos, que se definen y editan 216 en la vista general del esquema o en la vista del modelo de contenido, **no** en el ayudante de entrada Facetas).

Cuando seleccione un tipo simple en el diseño (un elemento o atributo de contenido simple), puede asignarle una aserción haciendo clic en la pestaña *Aserciones* (*imagen siguiente*) y después en el icono **Anexar** o **Insertar** de la barra superior. Aparece una fila donde puede escribir la expresión XPath 2.0 que se usará para definir el elemento `assertion`. Para que la expresión XPath conserve el valor del tipo simple es necesario usar una variable especial llamada **`$value`**. (Recuerde que, al no haber descendientes con los que realizar pruebas sino solamente un valor, el paso normal `self::node()` (o la abreviatura "." de este paso) no se puede usar en la expresión XPath.)

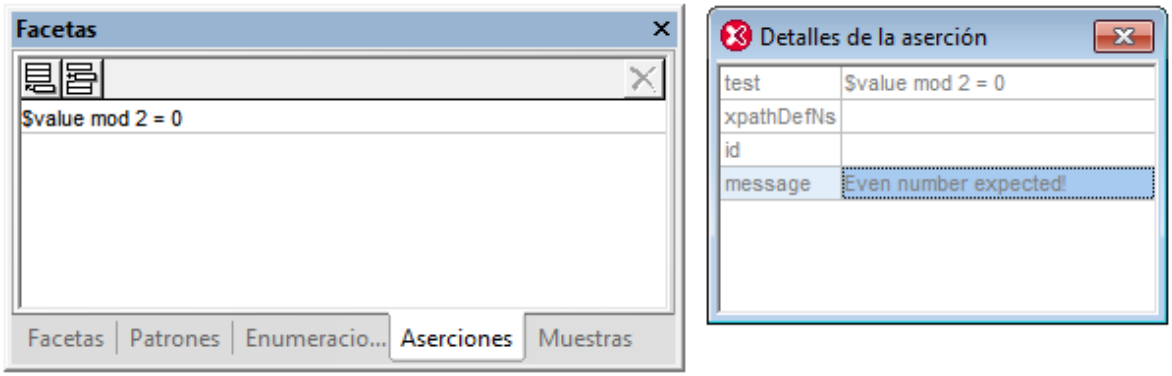
En la imagen siguiente, por ejemplo, la expresión XPath `string-length($value) = 6` prueba si el valor del tipo simple tiene seis caracteres. Si tiene seis caracteres, el elemento o atributo del documento de instancia es válido de acuerdo con la aserción.

© 2016-2022 Altova GmbH



Nota: si hay un error sintáctico, la expresión XPath se vuelve de color rojo. No obstante, recuerde que el tipo de datos se establece en tiempo de ejecución y, por tanto, los errores de tipo no se detectan mientras se teclea la expresión XPath. Por ello debe prestar atención a la hora de construir los tipos. Para ver un ejemplo de construcción de tipo, consulte la tercera expresión XPath de la imagen anterior, que convierte un valor de cadena (siempre que la aserción esté definida en un tipo simple `xs:string`) en un número antes de hacer una comparación numérica.

En un tipo simple puede especificar varias aserciones, igual que en la imagen anterior. Si usa varias aserciones, el elemento o atributo del documento de instancia debe cumplir con todas ellas. De lo contrario, el elemento o atributo no es válido. Las aserciones de la imagen anterior especifican que el valor del documento de instancia debe ser una cadena de seis caracteres que empiece por los caracteres EU y cuyos cuatro últimos caracteres sean caracteres numéricos con un valor comprendido entre `0000` y `4999`. Para editar sus detalles, haga clic con el botón derecho en la aserción en el ayudante de entrada Facetas y después elija **Detalles...** en el menú contextual. Esto abre la ventana modal "Detalles de la aserción" (*imagen siguiente*).



Lo más útil es aportar una explicación de la aserción junto con su definición, para que en caso de que la aserción no se cumpla cuando se valide el documento de instancia XML, la aplicación pueda emitir el mensaje adecuado. Como la especificación XML Schema no prevé ese tipo de mensajes, XMLSpy permite suministrar un mensaje en el espacio de nombres xml-schema-extensions de Altova http://www.altova.com/es/xml-schema-extensions (o en cualquier otro espacio de nombres) junto con la definición de la aserción y usarla en la validación del documento XML de instancia. Por ejemplo:

```
<xs:assert test="count(//MiNodo) ge 1" altova:message="Se necesita un elemento MiNodo
como mínimo"/> o
<xs:assertion test="count(//MiNodo) ge 1" altova:message="Se necesita un elemento
MiNodo como mínimo"/>
```

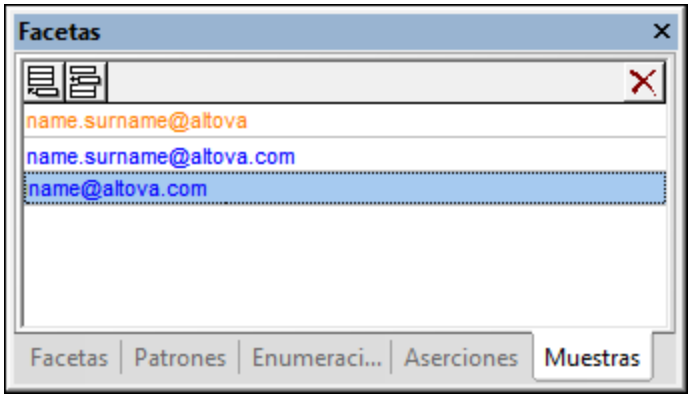
Si la restricción especificada por la aserción no se cumple, además del mensaje de error de validación el motor de validación de XMLSpy emitirá un mensaje sobre la aserción a modo de sugerencia. El validador informará del valor de un atributo `assert/@message` o `assertion/@message` independientemente del espacio de nombres en el que esté el atributo `message`. No obstante, recuerde que en la vista **Esquema** solamente podrá editar atributos `message` que estén en el espacio de nombres xml-schema-extensions de Altova. Para editar atributos `message` de otros espacios de nombres debe usar la vista **Texto**.

Para más información consulte el apartado Mensajes de aserción 240.

Nota: si puede, siempre es recomendable usar otras facetas en lugar de aserciones. Por ejemplo, la restricción especificada por la primera aserción de la imagen anterior se podría especificar con la faceta `length` (en la pestaña *Facetas*).

Pestaña Muestras

En la pestaña *Muestras* (*imagen siguiente*) puede especificar valores de muestra para generar un archivo XML a partir del esquema XML (con el comando de menú **DTD/Esquema | Generar archivo XML de muestra**). Si uno de los valores de muestra introducidos no es válido, el valor aparece en naranja. En la imagen siguiente, por ejemplo, el primer valor no es válido porque no coincide con la faceta `pattern` definida para las direcciones de correo eletrónico (véase la descripción de la pestaña *Patrones* más arriba).



Los valores de muestra se colocan en un elemento de anotación `altova:example` que está en el espacio de nombres http://www.altova.com/es/xml-schema-extensions. Añada una anotación `altova:ejemplo` haciendo clic en el icono **Anexar** o **Insertar** de la esquina superior izquierda y escriba el valor de `altova:ejemplo`. Para eliminar una anotación `altova:ejemplo`, selecciónela y haga clic en el icono **Eliminar** de la esquina superior derecha.

4.4.6 Validación y corrección rápida

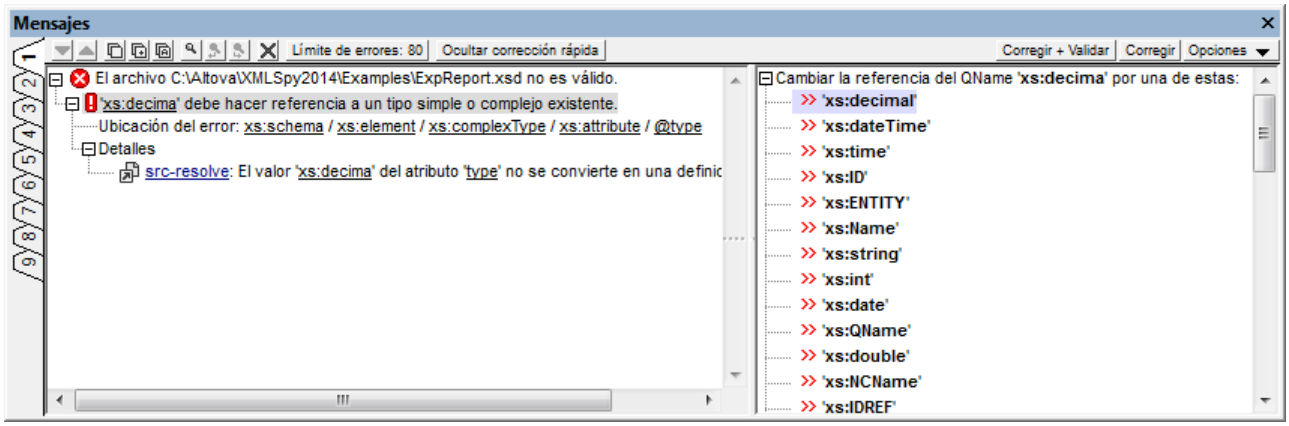
Los documentos de esquema XML se validan con el comando de menú **XML | Validar el documento XML (F8)**.

Si el documento es válido, aparece un mensaje confirmándolo en la ventana Mensajes.

Si el documento no es válido, la ventana Mensajes se divide en dos paneles (*imagen siguiente*). El panel izquierdo (el panel de errores) enumera los primeros `n` errores o todos los errores. El panel derecho es el panel

 © 2016-2022 Altova GmbH

de corrección rápida y ofrece soluciones para corregir el error seleccionado en el panel izquierdo. Por ejemplo, en la imagen siguiente está seleccionado el segundo error del panel de errores y el panel derecho muestra soluciones para corregir ese error. Si selecciona una de las posibles soluciones y hace clic en **Corregir + Validar** o en **Corregir**, el documento se corrige usando la solución elegida.



Panel de errores

La barra de herramientas de la ventana ofrece estas funciones:

- Las flechas **Arriba** y **Abajo** sirven para recorrer la lista de errores.
- Copiar un mensaje, copiar un mensaje y sus descendientes o copiar todos los mensajes.
- Buscar palabras con la función Buscar, Buscar siguiente y Buscar anterior. Esto es práctico si hay múltiples errores.
- Borrar todos los errores del panel.
- Establecer cuántos errores se muestran como máximo (de `1` a `999`). El valor predeterminado es `100`. Haga clic en el botón para establecer un límite de errores.
- Mostrar/Ocultar el panel de corrección rápida. Cuando el panel de corrección rápida está oculto, la barra de herramientas incluye el botón **Mostrar corrección rápida**. Haga clic en este botón para abrir el panel de corrección rápida (entonces el botón pasa a llamarse **Ocultar corrección rápida**). Si el botón **Mostrar/Ocultar corrección rápida** está deshabilitado, significa que no hay ninguna opción de corrección rápida disponible.

Panel de corrección rápida

La barra de herramientas de la ventana ofrece estas funciones:

- Haga clic en el botón **Corregir + Validar** para corregir el error seleccionado con la solución seleccionada y para volver a validar el documento. Si hay más errores, estos se enumeran en el panel de errores.
- Haga clic en el botón **Corregir** para corregir el error y no volver a validar el documento.
- Haciendo clic en el botón **Opciones** puede indicar qué ocurre cuando se hace doble clic en una solución del panel de corrección rápida: se ejecuta el comando **Corregir + Validar** o solo **Corregir**.

4.4.7 Mensajes de aserción

En XML Schema 1.1 se pueden definir aserciones para tipos complejos (usando elementos `xs:assert`) y tipos simples (usando elementos `xs:assertion`).

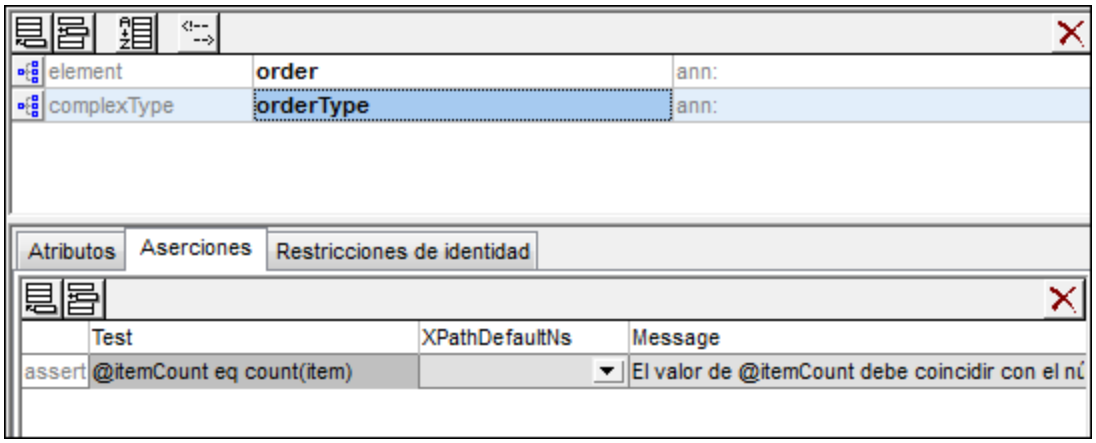
Lo más útil es aportar una explicación de la aserción junto con su definición, para que en caso de que la aserción no se cumpla cuando se valide el documento de instancia XML, la aplicación pueda emitir el mensaje adecuado. Como la especificación XML Schema no prevé ese tipo de mensajes, XMLSpy permite suministrar un mensaje en el espacio de nombres xml-schema-extensions de Altova http://www.altova.com/es/xml-schema-extensions (o en cualquier otro espacio de nombres) junto con la definición de la aserción y usarla en la validación del documento XML de instancia. Por ejemplo:

```
<xs:assert test="count(//MiNodo) ge 1" altova:message="Se necesita un elemento MiNodo como mínimo"/> o
<xs:assertion test="count(//MiNodo) ge 1" altova:message="Se necesita un elemento MiNodo como mínimo"/>
```

Si la restricción especificada por la aserción no se cumple, además del mensaje de error de validación el motor de validación de XMLSpy emitirá un mensaje sobre la aserción a modo de sugerencia. El validador informará del valor de un atributo `assert/@message` o `assertion/@message` independientemente del espacio de nombres en el que esté el atributo `message`. No obstante, recuerde que en la vista **Esquema** solamente podrá editar atributos `message` que estén en el espacio de nombres xml-schema-extensions de Altova. Para editar atributos `message` de otros espacios de nombres debe usar la vista **Texto**.

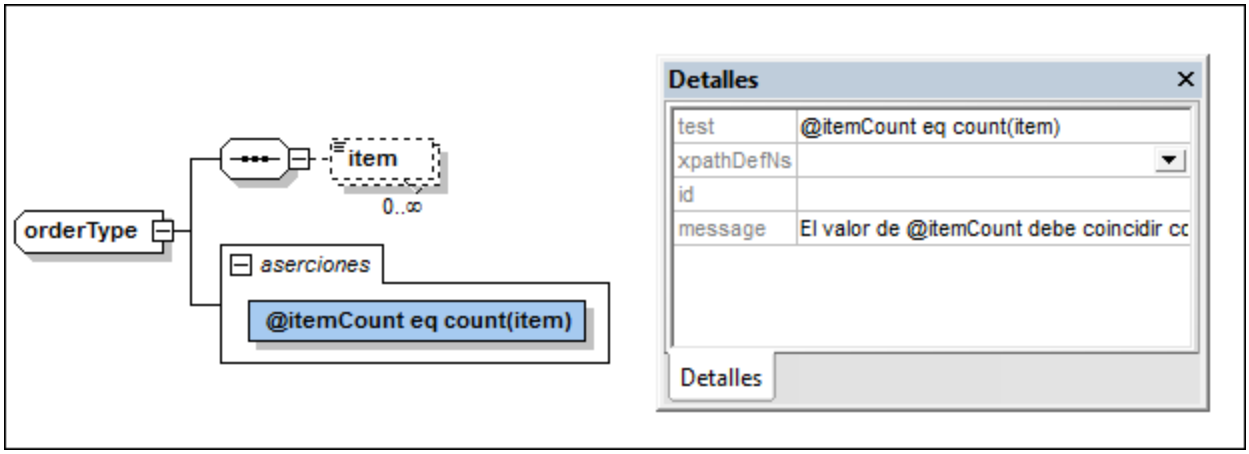
Editar mensajes para `xs:assert`

En la vista **Esquema** los elementos `xs:assert` (para tipos complejos) se pueden crear y editar en el panel Atributos/Aserciones/Restricciones de identidad (AARID) 216 o en el ayudante de entrada Detalles 216 del tipo complejo correspondiente. La imagen que aparece a continuación muestra una aserción para el tipo complejo `orderType`. La aserción (en este caso un elemento `xs:assert`) se definió en la pestaña *Aserciones* (de la vista general del esquema) junto con un mensaje de aserción.



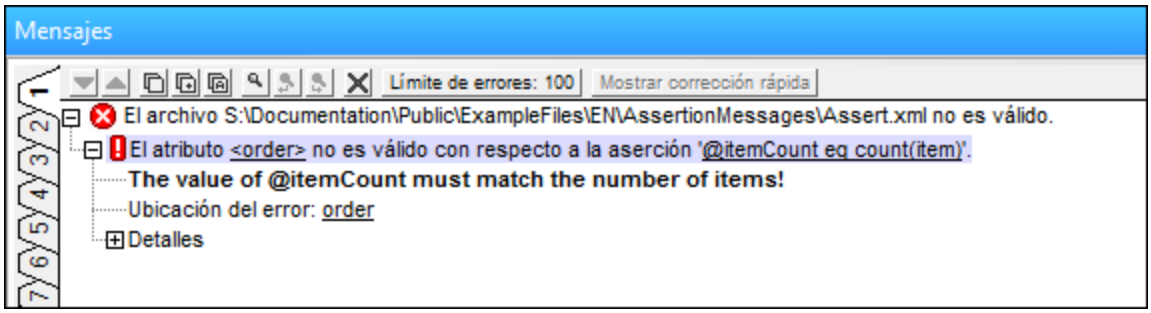
Si está seleccionada la opción **Mostrar aserciones en el diagrama** 196, las aserciones de los tipos complejos también se pueden crear y editar en la vista del modelo de contenido. Para agregar o editar un mensaje de aserción, seleccione la aserción y escriba el mensaje de aserción en el ayudante de entrada Detalles (*imagen siguiente*).

© 2016-2022 Altova GmbH



Recuerde que los mensajes de aserción creados de esta manera están en el espacio de nombres http://www.altova.com/es/xml-schema-extensions. Cuando se añade el primer mensaje de aserción en el esquema XML desde el panel AARID 216 o desde el ayudante de entrada Detalles 216, el espacio de nombres xml-schema-extensions de Altova se declara en el elemento `xs:schema` automáticamente.

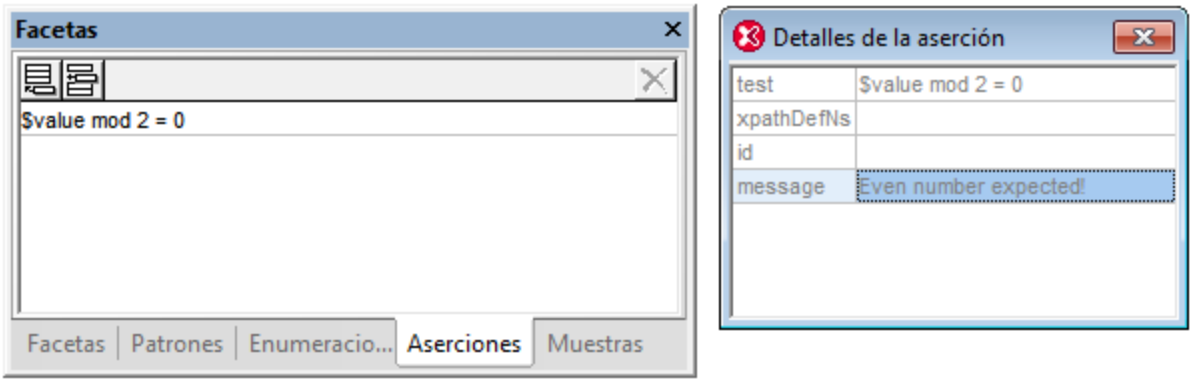
Si al validar un archivo XML la prueba de la aserción da un resultado negativo, la aplicación emite el mensaje definido para la aserción junto con un mensaje de error (*imagen siguiente*).



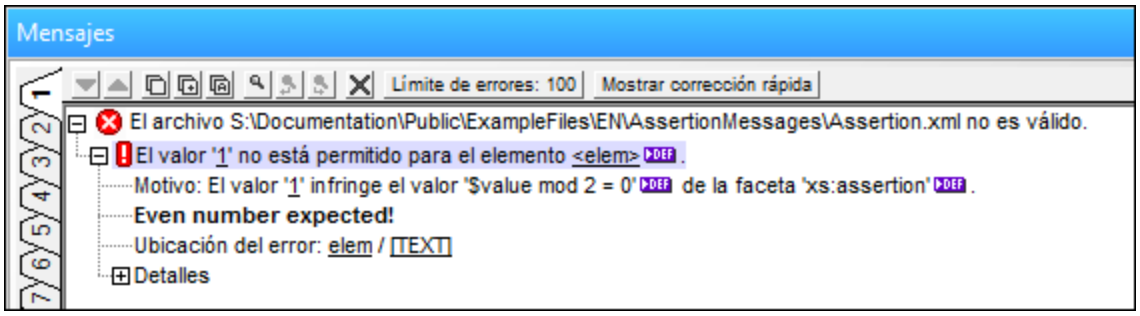
Editar mensajes para `xs:assertion`

En la vista **Esquema** puede crear y editar elementos `xs:assertion` (para tipos simples) en el ayudante de entrada Facetas 234 del tipo simple correspondiente. Para editar el mensaje de aserción haga clic con el botón derecho en la aserción en el ayudante de entrada Facetas (*imagen siguiente, izda*) y elija **Detalles...** en el menú contextual. En la ventana modal que aparece puede editar el mensaje (*imagen siguiente, dcha*).

Recuerde que los mensajes de aserción creados de esta manera están en el espacio de nombres http://www.altova.com/es/xml-schema-extensions. Cuando se añade el primer mensaje de aserción en el esquema XML desde la ventana modal Detalles de la aserción, el espacio de nombres xml-schema-extensions de Altova se declara en el elemento `xs:schema` automáticamente.



Si al validar un archivo XML la prueba de la aserción da un resultado negativo, la aplicación emite el mensaje definido para la aserción junto con un mensaje de error (*imagen siguiente*).



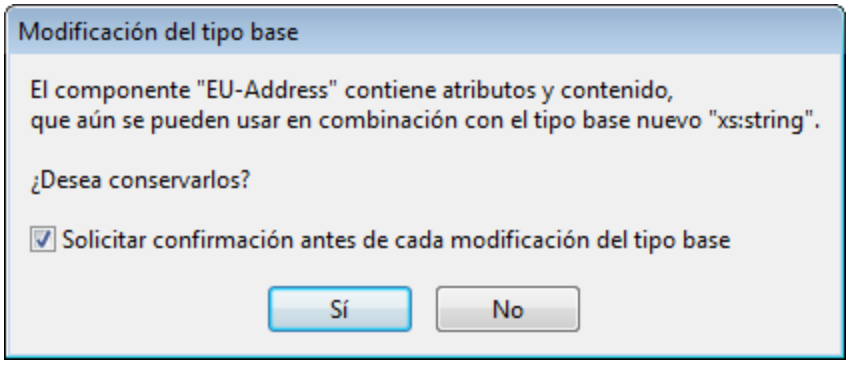
4.4.8 Modificación del tipo base

Si se cambia el tipo base de un tipo derivado en la vista Esquema, el contenido, los atributos, las facetas y los valores de muestra del tipo derivado se pueden abordar de dos maneras diferentes:

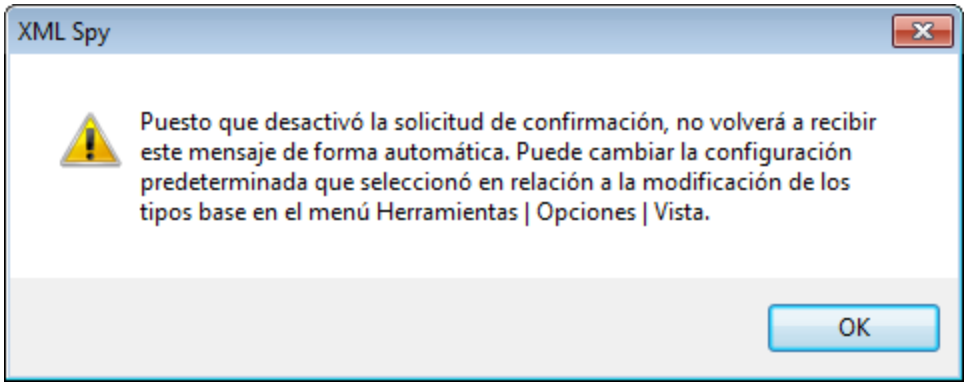
- Se pueden mantener si aún son pertinentes en combinación con el tipo base nuevo.
- Se pueden quitar automáticamente independientemente de si aún son pertinentes o no en combinación con el tipo base nuevo.

Cuando se cambia el tipo base de un tipo derivado que contiene contenido, atributos, facetas o valores de muestra, aparece el cuadro de diálogo "Modificación del tipo base" (*imagen siguiente*).

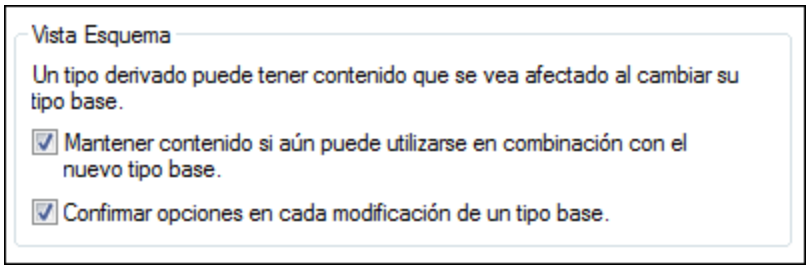
© 2016-2022 Altova GmbH



Si desactiva la casilla "Solicitar confirmación", aparece un mensaje emergente (*imagen siguiente*) indicando que puede volver a activar la confirmación en la sección *Vista* del cuadro de diálogo "Opciones" (**Herramientas | Opciones | Vista**[1597]).



En el panel *Vista Esquema* (*imagen siguiente*) de la sección *Vista* del cuadro de diálogo "Opciones" (**Herramientas | Opciones | Vista**[1597]), puede indicar si el contenido se debe mantener y si se debe solicitar la confirmación del usuario cada vez que se modifique el tipo base.



Marque ambas casillas si desea que se mantenga el contenido por defecto y si desea confirmar las opciones cada vez que se modifique el tipo base.

4.4.9 Restricciones inteligentes

Cuando se restringe un tipo complejo, partes del modelo de contenido del tipo base se vuelven a escribir en el tipo derivado. Esto puede ser algo difícil si el modelo de contenido es complejo. El motivo es que puede ser complicado recordar qué modelo de contenido tenía el tipo base mientras se edita el tipo derivado.

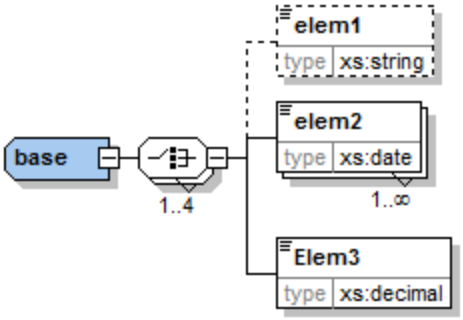
La función de restricción inteligente combina y correlaciona el modelo de contenido del tipo base con el del tipo derivado en la vista gráfica del modelo de contenido derivado. En el tipo complejo derivado se pueden ver todas las partículas del tipo complejo base y sus relaciones con el tipo derivado. Además, la restricción inteligente ofrece sugerencias visuales sobre todas las maneras posibles de restringir el tipo base. Con esta función es más fácil restringir correctamente el tipo derivado.

Para activar la función de restricción inteligente:

- Haga clic en el icono "Habilitar restricción inteligente"  de la barra de herramientas Diseño de esquemas.

El ejemplo que aparece a continuación muestra las características de la restricción inteligente.

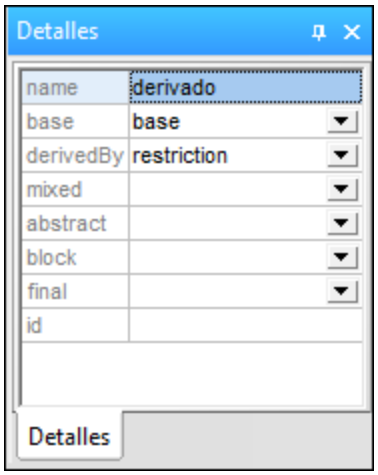
En este ejemplo el tipo base utilizado es el tipo complejo que aparece a continuación:



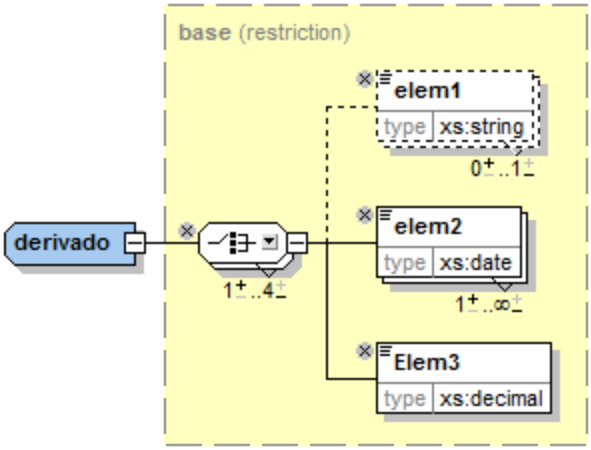
El tipo complejo derivado se deriva del tipo base de la siguiente manera:

1. Cree un tipo complejo nuevo en el esquema y llámelo derivado.
2. En el ayudante de entrada Detalles, seleccione base de la lista desplegable del campo **base** y restriction de la lista desplegable del campo **derivedBy** (es decir, lo derivamos por restricción).

© 2016-2022 Altova GmbH

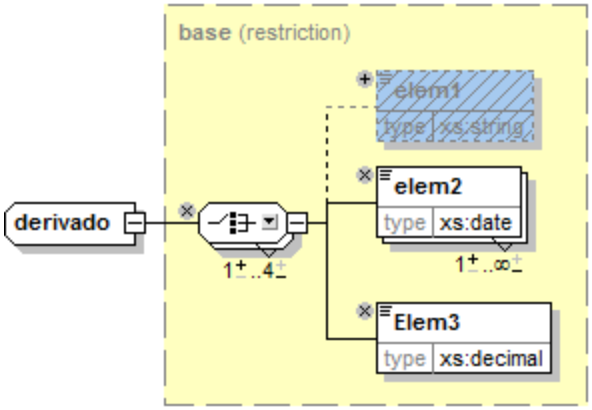


En la vista del modelo de contenido del tipo derivado, active la función de restricción inteligente y el tipo derivado nuevo tendrá este aspecto:

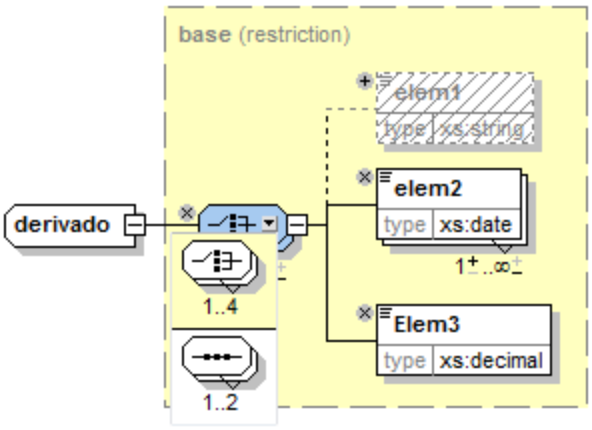


Observe que el modelo de contenido incluye ahora unos controles que sirven para restringir el tipo derivado de este ejemplo:

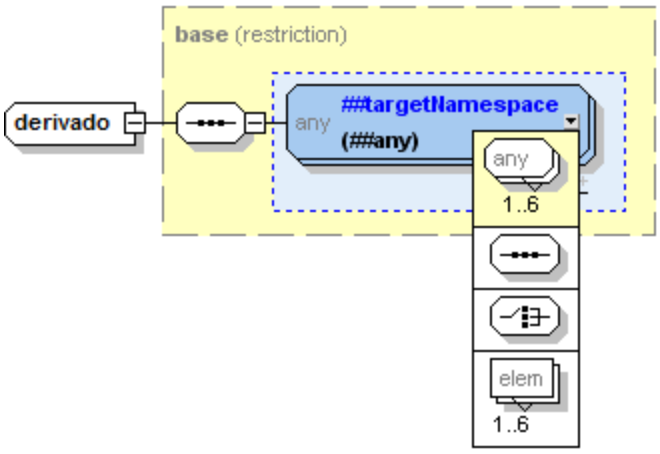
- El icono sirve para quitar del tipo derivado elementos que están en el tipo base. En este ejemplo, usamos este icono para quitar elem1 del tipo derivado. Para volver a añadirlo, haga clic en este icono . Para volver a añadirlo, haga clic en este icono .



- En el compositor de opción Choice haga clic en el icono en forma de flecha y aparece una lista con la que puede cambiar el modelo de grupo de Choice a Sequence:

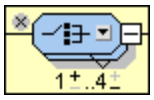


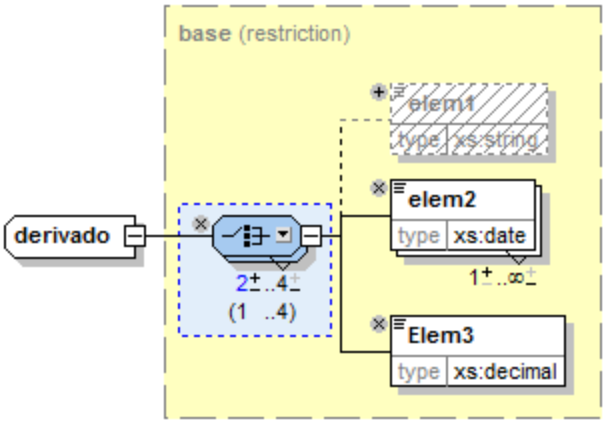
Los comodines también se pueden cambiar así, con el icono en forma de flecha. Por ejemplo:



La especificación XML Schema contiene una lista que detalla qué partículas se pueden sustituir y con qué partículas.

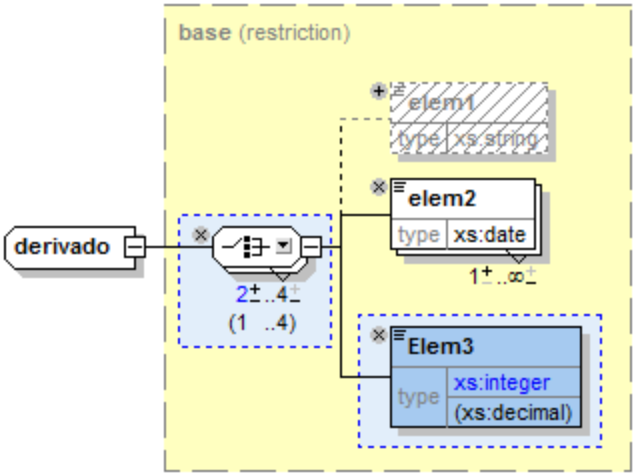
© 2016-2022 Altova GmbH

- Cambie el número de repeticiones del modelo de grupo con ayuda de este control  e incremente el número mínimo de repeticiones pulsando el signo + situado junto al número 1. O reduzca el número máximo de repeticiones pulsando el signo - situado junto al número 4. Estos controles están disponibles si el intervalo de repetición del tipo base describe un intervalo real (p. ej. 2-5) y no una cantidad concreta (p. ej. 4-4). Los controles también están disponibles si el rango de repetición no es correcto.

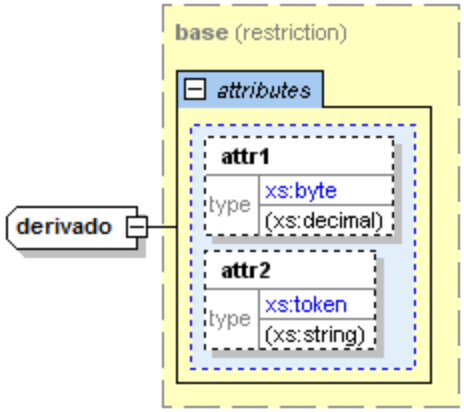


En este ejemplo puede ver que se cambió el número mínimo de repeticiones de este elemento a 2. Observe que el modelo de grupo ahora tiene un fondo de color azul. Esto indica que ya no es igual que el modelo de grupo del tipo complejo base. Además, el intervalo de repetición permitido del modelo de grupo de la partícula base aparece entre paréntesis.

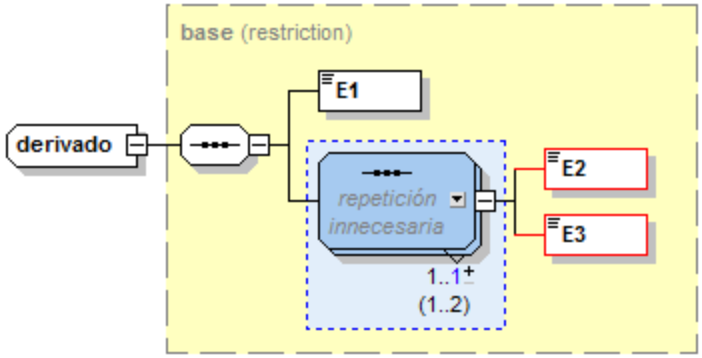
- Puede cambiar los tipos de datos de atributos o elementos si el tipo de datos nuevo es una restricción válida del tipo de datos base, de acuerdo con lo definido en la especificación XML Schema. Por ejemplo, puede cambiar el tipo de datos de elem3 del tipo de datos derived de decimal a integer. Tras los cambios, el elemento tiene un color de fondo azul, indicando que es diferente al elemento del tipo base. Además, el tipo que el elemento tiene en el tipo base aparece entre paréntesis:



Este ejemplo muestra atributos cuyos tipos de datos se restringieron en el tipo complejo derivado:



- La función de restricción inteligente le avisa si el modelo de contenido incluye *repeticiones innecesarias*. Una repetición innecesaria tiene lugar si, por ejemplo, una secuencia del modelo de contenido no es necesaria. Este ejemplo muestra una repetición innecesaria:



Nota: las repeticiones innecesarias sólo aparecen si el modelo de contenido incluye un error. Un modelo de contenido puede incluir una repetición innecesaria y ser válido, en cuyo caso la repetición innecesaria no se señaliza (para evitar confundir al usuario).

Consulte la especificación XML Schema para obtener más información sobre repeticiones innecesarias.

4.4.10 xml:base, xml:id, xml:lang, xml:space

Según la especificación XML Namespaces el espacio de nombres `http://www.w3.org/XML/1998/namespace` está enlazado por definición al prefijo `xml`. Esto significa que este es el espacio de nombres que se debe usar con el prefijo `xml`y que está reservado para este prefijo. En este espacio de nombres hay cuatro atributos que pueden ser secundarios de cualquier elemento XML de cualquier documento XML (ya sea esquema o instancia):

- `xml:base` (para establecer el URI base de un elemento)
- `xml:id` (para especificar el identificador único de un elemento)

 © 2016-2022 Altova GmbH

- `xml:lang` (para identificar el idioma usado dentro del elemento)
- `xml:space` (para especificar cómo se deben tratar los espacios en blanco en el elemento)

En la vista Esquema, después de importar en el documento de esquema XML el espacio de nombres de XML Namespaces, se puede hacer referencia a estos cuatro atributos `xml:` en cualquier elemento del esquema.

Para declarar uno de estos atributos en un elemento, siga estos pasos:

1. Declare el espacio de nombres de XML Namespaces para el documento de esquema y enlace el espacio de nombres al prefijo `xml:`. Cuando los cuatro atributos `xml:` se usen en el documento, su nombre se anexará para incluir la parte del espacio de nombres correspondiente.
2. Importe el espacio de nombres de XML Namespaces. El validador de XMLSpy reconocerá el espacio de nombres y creará cuatro atributos `xml:` como atributos globales, a los que se puede hacer referencia en ese esquema.
3. Inserte el atributo `xml:` que desea usar como secundario de un elemento. El atributo se declara como referencia al atributo global "importado".

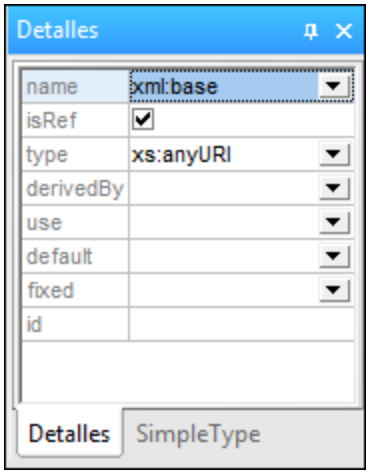
Declarar el espacio de nombres de XML Namespaces

Para declarar el espacio de nombres de XML Namespaces (`http://www.w3.org/XML/1998/namespace`) utilice el diálogo "Configuración del esquema", donde se almacenan y editan todos los espacios de nombres declarados para el esquema. El espacio de nombres debe enlazarse al prefijo `xml:`. (Otra opción es abrir el esquema en la vista Texto y declarar el espacio de nombres, con el prefijo `xml:`, en el elemento `xs:schema`.)

Importar el espacio de nombres de XML Namespaces

En la vista general del esquema, cree una declaración de importación global. Para ello, haga clic en el icono Insertar o Anexar (situados en la esquina superior izquierda de la vista general del esquema) y seleccione **Import**. Escriba el espacio de nombres de XML Namespaces. La declaración de importación tiene este aspecto en la vista Texto:

```
<xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/XML/1998/namespace"/>.
```



XInclude y xml:base

Cuando el elemento `include` de XInclude se sustituye por el archivo XML indicado en el atributo `href` del elemento de inclusión, el elemento de nivel superior del documento XML analizado se incluye con un atributo `xml:base`. Si se va a validar este documento, el esquema debe definir un atributo `xml:base` en los elementos correspondientes.

4.4.11 Adelante y atrás: navegar de una posición a otra

Los comandos **Adelante** y **Atrás** de la vista Esquema sirven para volver a posiciones en las que estuvo previamente en la vista Esquema. Esto puede ser muy útil cuando, por ejemplo, pasa de un componente de esquema a otro y desea ver otra vez el componente anterior. Al hacer clic una vez en el botón **Atrás** de la barra de herramientas, volverá a la posición que visualizó antes. Si hace clic repetidas veces en el botón **Atrás**, podrá ver hasta 500 posiciones anteriores. Después de volver atrás a posiciones anteriores, puede usar el botón **Adelante** de la barra de herramientas para ver las posiciones más actuales.

Las teclas de acceso rápido para estos dos comandos son:

- ***Atrás:*** **Alt + Flecha izquierda**
- ***Adelante:*** **Alt + Flecha derecha**

Diferencias entre Atrás/Adelante y Deshacer/Rehacer

Es necesario mencionar que los comandos **Atrás** y **Adelante** no son lo mismo que los comandos **Deshacer** (**Ctrl+Z**) y **Rehacer** (**Ctrl+Y**). Estos dos pares de comandos llevan a cabo dos series de pasos diferentes. Al hacer clic una vez en el comando **Atrás**, volverá al último componente de esquema que visualizó. Al hacer clic una vez en el comando **Deshacer**, se deshace el último cambio de edición que realizó, independientemente de cuándo se hiciese el cambio.

Notas adicionales

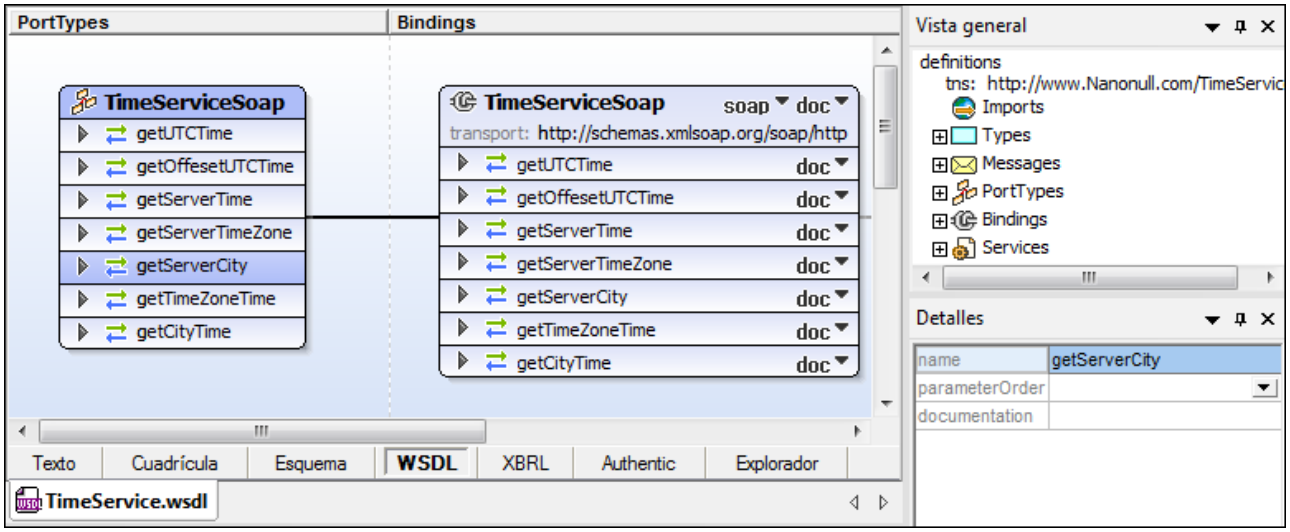
Es necesario tener en cuenta que:

- El botón **Atrás** permite volver a visualizar hasta 500 posiciones anteriores.
- la función Adelante/Atrás está habilitada en todos los esquemas con los que esté trabajando. Si cierra uno de los esquemas o visualiza uno de los esquemas en otra vista, los comandos abrirán el esquema en la vista Esquema o cambiarán a la vista Esquema.
- Si un componente que visualizó con anterioridad se elimina, los comandos no podrán llevarle a ese componente. Por tanto, al hacer clic en el botón **Atrás**, volverá a la posición anterior al componente que ya no está disponible. Si el componente formaba parte de una posición anterior, esta posición se mostrará pero no contendrá el componente eliminado.

© 2016-2022 Altova GmbH

4.5 Vista WSDL

La vista WSDL (*ver imagen siguiente*) ofrece una interfaz para editar documentos WSDL 1.1 y WSDL 2.0 de forma gráfica. La vista WSDL se habilita cuando está activo un documento WSDL. Para abrir la vista, haga clic en la pestaña WSDL. La estructura y los componentes del documento WSDL se crean en la ventana principal[252] mediante mecanismos de diseño gráfico, mientras que los ayudantes de entrada[256] ofrecen funciones de edición.



La ventana principal[252] (formada por secciones para portTypes (WSDL 1.1) o interfaces (WSDL 2.0), bindings (enlaces) y servicios) y los ayudantes de entrada Vista general[256] y Detalles[263] se describen en los distintos apartados de esta sección. Para más información sobre cómo trabajar con proyectos, consulte la sección Menú Proyecto[1298] de la *Comandos de menú*.

Funciones disponibles en la vista WSDL

Estas son las funciones de la vista WSDL:

- Una presentación gráfica en la ventana principal de todos los documentos WSDL, organizados en grupos de portTypes (WSDL 1.1) o interfaces (WSDL 2.0), enlaces (bindings) y servicios.
- Manipulación directa de elementos WSDL mediante operaciones de arrastrar y colocar.
- Posibilidad de añadir, insertar y eliminar los elementos WSDL visibles en la vista gráfica (con ayuda de un menú contextual).
- Posibilidad de introducir y editar valores en el ayudante de entrada Detalles.
- Validación WSDL con el borrador de trabajo del W3C.
- Cambio automático a la vista Esquema si se descubre que un esquema importado o referenciado no es válido.
- Edición de tipos de esquemas desde la vista WSDL directamente.
- Generación de documentación WSDL en formato MS Word o HTML.
- Generación de un diagrama (imagen PNG) del documento WSDL visible en la ventana principal.
- Impresión de la vista de la ventana WSDL.

Visualización de archivos

Recuerde estos puntos a la hora de visualizar archivos en la vista WSDL:

- Cuando se abre un archivo WSDL, el archivo se abre automáticamente en la vista WSDL.
- También puede ver documentos WSDL en la vista Texto o Cuadrícula. Para ello, basta con hacer clic en la pestaña correspondiente.
- Si el archivo WSDL contiene una referencia a un esquema XML, dicho esquema se puede ver y editar con solo seleccionar el comando de menú **WSDL | Tipos | Editar esquema en la vista Esquema**. Como resultado el archivo de esquema se abre en la vista Esquema.
- Si está abierto un archivo de esquema asociado, entonces no puede cambiar la vista del archivo WSDL (p. ej. de la vista WSDL a la vista Texto). Antes de cambiar de vista, guarde los cambios en el archivo de esquema y ciérrelo.

La vista WSDL ofrece dos ayudantes de entrada que le asistirán en la edición de documentos WSDL: Vista general 256 y Detalles 263. Para acoplar o desacoplar los ayudantes de entrada, haga doble clic en su barra de título. Mientras están acoplados, puede ocultarlos automáticamente si hace clic en el icono en forma de chincheta de la barra de título. Cuando se ocultan automáticamente, el ayudante de entrada se minimiza en forma de pestaña y aparece al borde de la ventana de la aplicación. Para volver a acoplar el ayudante de entrada ocultado automáticamente, pase el cursor por encima de la pestaña y haga clic otra vez en el icono en forma de chincheta de la barra de título.

4.5.1 Ventana principal

La ventana principal es donde se edita el documento WSDL. Está formada por tres secciones verticales: (i) portTypes (WDL 1.1) o interfaces (WSDL 2.0) 253; (ii) enlaces (bindings) 254 y (iii) servicios 255. La relación entre un portType y un enlace y entre un enlace y un servicio se indica mediante líneas de conexión. A continuación describimos en detalle cada una de estas tres secciones.

Símbolos utilizados en la ventana principal

En la ventana principal se utilizan estos símbolos:

 Tipo de puerto (portType) en WSDL 1.1 o interfaz en WSDL 2.0

 Enlace (binding)

 Servicio

 Elemento de error (fault)

 Operación. La flecha verde representa entradas y la azul representa salidas. Dependiendo del tipo de operación, se usa el símbolo correspondiente.

 Mensaje

 Parte de mensaje (parámetro)

 Elemento XSD *element*

 Elemento XSD *simpleType* o *complexType*

 Puerto

Agregar portTypes, interfaces, enlaces y servicios nuevos

Para agregar un portType nuevo (en documentos WSDL 1.1), una interfaz nueva (en documentos WSDL 2.0), un enlace nuevo o un servicio nuevo, haga clic con el botón derecho en la ventana principal (pero fuera del recuadro de un componente) y seleccione el comando pertinente del menú contextual que aparece.

© 2016-2022 Altova GmbH

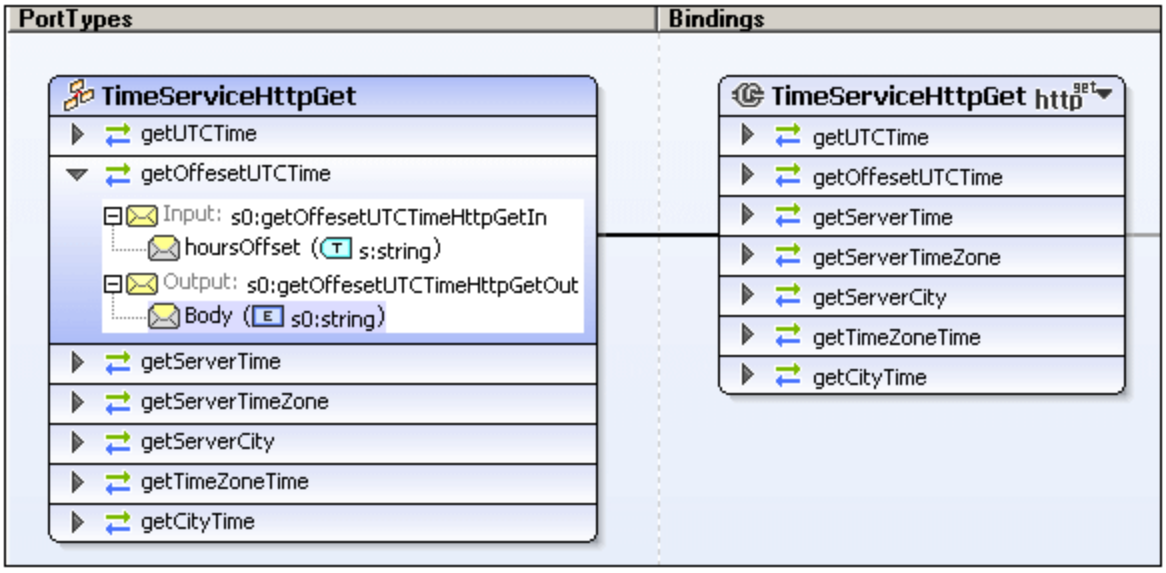
Operaciones de arrastrar y colocar

Puede utilizar operaciones de arrastrar y colocar en estos dos casos:

- En la ventana principal puede arrastrar y colocar asociaciones entre portTypes (WSDL 1.1) o interfaces (WSDL 2.0) y enlaces y entre enlaces y servicios.
- En documentos WSDL 2.0 los elementos del ayudante de entrada Vista general se pueden arrastrar hasta elementos de error (fault) de interfaces tanto en la ventana principal como en el ayudante de entrada Vista general.

PortTypes (WSDL 1.1) e interfaces (WSDL 2.0)

La sección **PortTypes** (documentos WSDL 1.1) contiene todos los tipos de puerto definidos en el documento WSDL (*la imagen siguiente muestra un solo portType en la sección PortTypes*). La sección **Interfaces** (en documentos WSDL 2.0) contiene todas las interfaces definidas en el documento WSDL.



Cada portType o interfaz se representa en forma de recuadro, que contiene las operaciones definidas para el portType o la interfaz. Los componentes se pueden editar dentro del recuadro directamente. A continuación describimos las principales funciones disponibles en los recuadros de portTypes e interfaces:

- Las operaciones se pueden expandir para ver sus mensajes. Para ello haga clic en el icono situado a la izquierda del nombre de la operación.
- En WSDL 1.1 un mensaje puede contener una parte de mensaje (parámetro). Estos mensajes se pueden expandir para mostrar la parte de mensaje.
- Al hacer clic con el botón derecho en un componente del recuadro de un portType (ya sea en un portType, una operación, un mensaje o una parte de mensaje), aparece un menú contextual con varias opciones. Por ejemplo, si hace clic con el botón derecho en el nombre de un portType, el menú contextual que aparece permite anexar un portType nuevo, anexar una operación al portType seleccionado o crear un enlace para el portType seleccionado.
- Las propiedades opcionales WSDL 2.0 de las interfaces (*extends, styleDefault* y *documentation)* se ocultan si están vacías. Para editarlas, seleccione el comando **Editar** del menú contextual de la interfaz.

- En documentos WSDL 2.0 puede editar las propiedades de las operaciones si selecciona el comando **Editar** del menú contextual de la operación. El recuadro ofrece un cuadro combinado para la propiedad *style*, que contiene todos los valores posibles para la propiedad.
- Recuerde que cuando un componente está seleccionado, sus detalles se pueden editar en el ayudante de entrada Detalles.
- La documentación para portTypes e interfaces aparece debajo de sus recuadros.

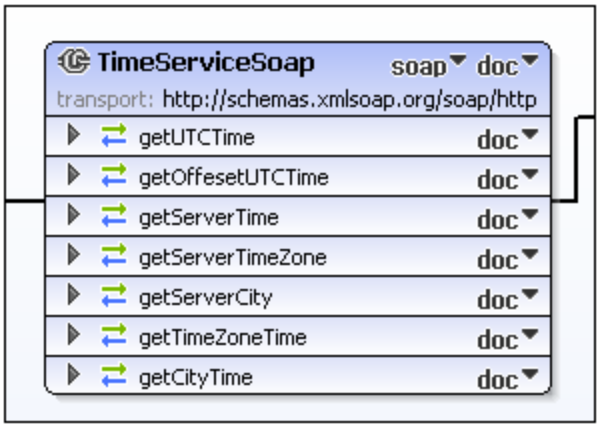
La asociación de un portType o una interfaz con un enlace se indica en la ventana principal mediante una línea de conexión negra que vincula el recuadro del portType o de la interfaz con el recuadro del enlace. El recuadro del enlace está situado en la sección **Bindings** de la ventana principal.

Enlaces (Bindings)

Un enlace define los formatos de mensaje y los detalles de protocolo para:

- las operaciones definidas por un portType determinado (WSDL 1.1)
- o las operaciones y elementos de error (fault) definidos por una interfaz determinada (WSDL 2.0).

En WSDL 1.1 se pueden crear enlaces para extremos SOAP 1.1 o 1.2 o para verbos GET y POST de HTTP 1.1. En WSDL 2.0 se pueden crear enlaces para extremos SOAP 1.1 o 1.2 o para HTTP. Cada enlace se representa en un recuadro (*ver imagen siguiente*) en la sección **Bindings** de la ventana principal. El recuadro del enlace contiene todas las operaciones y elementos de error (fault) del portType o de la interfaz asociados (*ver imagen siguiente*).



Para asociar un enlace a un portType o a una interfaz:

- Haga clic con el botón derecho en un portType o en una interfaz y seleccione el comando **Crear enlace para el portType** o **Crear enlace para la interfaz**, según corresponda.
- Haga clic con el botón derecho en un enlace WSDL 1.1 y edite la propiedad *PortType*.
- Haga clic con el botón derecho en un enlace WSDL 2.0 y seleccione el comando **Editar | Interface**.

Para definir el enlace, en el primer cuadro combinado situado a la derecha del nombre del enlace (*imagen siguiente*), seleccione el protocolo deseado. En WSDL 1.1 puede elegir entre estos protocolos: `soap 1.1`, `soap 1.2`, `http-get` o `http-post`. Si selecciona un protocolo SOAP, además podrá definir (con el segundo cuadro combinado) si el estilo debe ser `doc` o `rpc`. En los documentos WSDL 2.0, la propiedad `wsoap:protocol` se puede añadir o editar con ayuda del comando **Editar** del menú contextual del enlace.

© 2016-2022 Altova GmbH



En WSDL 1.1 las codificaciones MIME (conocidas también como enlaces MIME) se definen a nivel de mensaje. Para definir una codificación MIME, haga clic con el botón derecho en el mensaje (*imagen siguiente*) y anexe la definición MIME correspondiente. En la imagen siguiente puede ver definiciones MIME que se crearon para el mensaje `Output`.



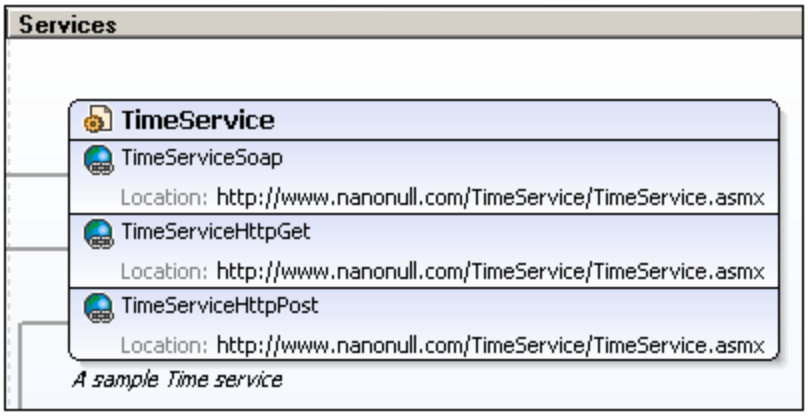
Al hacer clic con el botón derecho en cualquier componente del recuadro de un enlace, aparece un menú contextual. Dicho menú incluye comandos para anexar o eliminar enlaces, editar elementos de extensibilidad o definir mensajes. Tenga en cuenta que cuando está seleccionado el recuadro de un enlace o un componente del enlace, las definiciones se presentan en el ayudante de entrada Detalles, donde se pueden editar.

Para crear un puerto para un enlace, haga clic con el botón derecho en la barra de título del recuadro de un enlace y seleccione el comando **Crear puerto para el enlace** (documentos WSDL 1.1) o **Crear extremo para el enlace** (documentos WSDL 2.0). El puerto o extremo asociado se crea dentro de un recuadro de servicio (en la sección **Services** de la ventana principal). La asociación entre un enlace y un puerto se representa mediante una línea de conexión negra.

La documentación de los enlaces aparece debajo de los recuadros de los enlaces.

Servicios

Un servicio agrupa un conjunto de puertos relacionados (WSDL 1.1) o de extremos relacionados (WSDL 2.0). Se representa en forma de recuadro en la sección **Services** de la ventana principal (*imagen siguiente*). Cada recuadro incluye varias declaraciones de puerto o extremo (*ver imagen siguiente*).



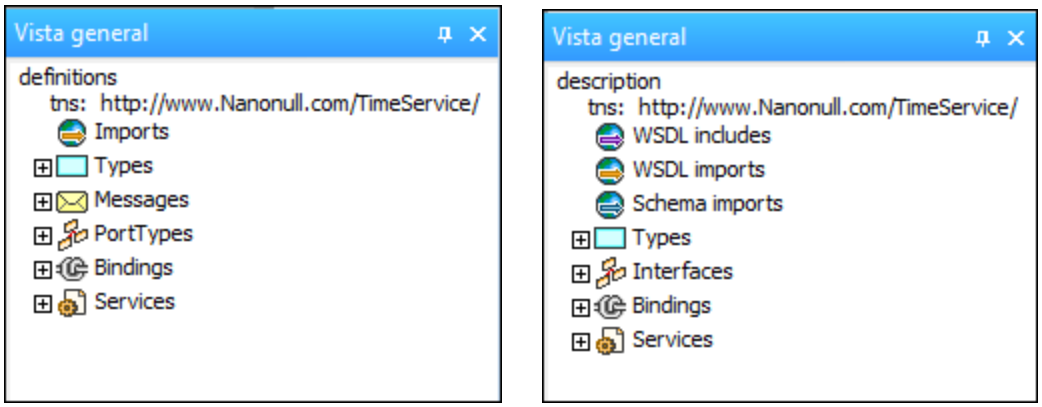
El nombre del servicio, del puerto o extremo, el enlace asociado a un puerto o extremo y la información de dirección de un puerto o extremo se pueden editar en el recuadro del servicio o en el ayudante de entrada Detalles. Al hacer clic con el botón derecho en el recuadro de un servicio o en un componente del recuadro, aparece un menú contextual con comandos.

La documentación de los servicios aparece debajo de los recuadros de los servicios.

4.5.2 Ventana de vista general

El **ayudante de entrada Vista general** (*imagen siguiente*) ofrece una visión de conjunto del documento WSDL, al agrupar los componentes del documento en categorías estructurales y mostrar el espacio de nombres de destino, los esquemas importados y los documentos WSDL incluidos e importados. Además de portTypes (o interfaces en WSDL 2.0), mensajes (WSDL 1.1), enlaces y servicios, la vista general incluye también los tipos definidos en el documento.

En el ayudante de entrada Vista general también puede gestionar las acciones de importación e inclusión de archivos WSDL y de esquema XML.



Ayudante de entrada Vista general en WSDL 1.1 (izquierda) y WSDL 2.0 (derecha)

En cada categoría, los componentes se representan en forma de árbol. Cada componente del árbol se puede expandir para mostrar su contenido (o contraer para ocultar su contenido). Al seleccionar un componente en el

 © 2016-2022 Altova GmbH

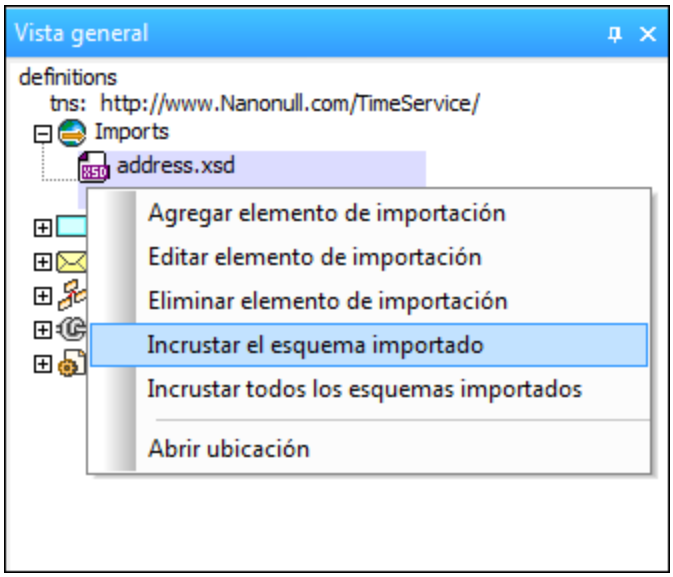
ayudante de entrada Vista general, las propiedades del componente aparecen en el ayudante de entrada Detalles 263, donde se pueden editar Los nombres de los componentes de esquema y WSDL que aparecen en el árbol se pueden editar en él directamente. Sin embargo, los componentes definidos de manera externa (es decir, los que están en documentos WSDL o esquemas incluidos o importados) se presentan en color gris y no se pueden editar. A continuación describimos las categorías estructurales del ayudante de entrada Vista general.

Espacio de nombres de destino (WSDL 1.1 y 2.0):

Se representa en el árbol con las siglas tns. El espacio de nombres de destino se puede editar en el ayudante de entrada Vista general. El resto de los espacios de nombres se deben editar en la vista Texto.

Imports (WSDL 1.1):

Si lo desea, puede importar archivos de esquema XML (XSD) y archivos WSDL al documento WSDL activo. Para importar un archivo WSDL o de esquema XML, haga clic con el botón derecho en el nodo *Imports* o en un archivo ya importado y seleccione **Agregar elemento de importación** en el menú contextual. Si hace clic con el botón derecho en un archivo importado ya disponible de la lista Imports, aparece un menú contextual con opciones para agregar un elemento de importación (**Agregar elemento de importación**), reemplazar el archivo seleccionado con otro archivo (**Editar elemento de importación**) o eliminar el archivo importado. También puede abrir el archivo desde su ubicación. El archivo se abre en la vista WSDL (archivo `.wsdl`) o en la vista Esquema (archivo `.xsd`), donde puede editarlo.



Después de importarlos, los esquemas XML se pueden incrustar en el archivo WSDL. Al incrustar un esquema se crea el esquema como esquema alineado dentro del elemento `types` y se elimina el elemento `import`. Para incrustar un esquema importado, haga clic con el botón derecho en el esquema de la lista *Imports* y seleccione el comando Incrustar esquema importado o Incrustar todos los esquemas. Este último comando, que afecta a todos los esquemas importados, también está disponible en el menú contextual del nodo *Imports*.

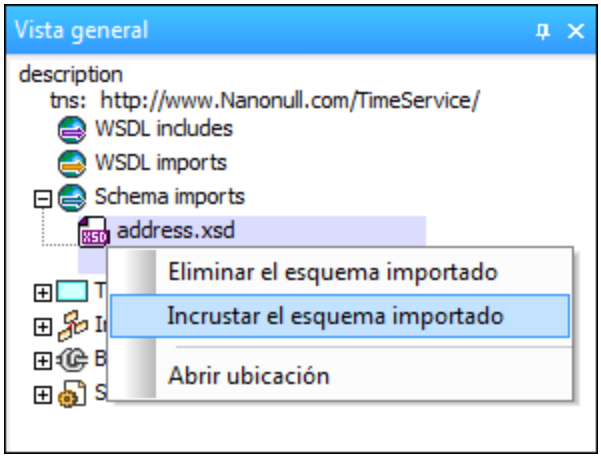
WSDL includes, WSDL imports, Schema imports (WSDL 2.0):

Si lo desea puede incluir o importar archivos de esquema XML (XSD) y archivos WSDL en el documento WSDL activo. Para incluir o importar un archivo, haga clic con el botón derecho en el nodo correspondiente (*WSDL*

Includes, WSDL Imports o Schema Imports), busque el archivo que desea incluir o importar y añádalo. El espacio de nombres de un archivo importado se genera automáticamente a partir del espacio de nombres de destino del archivo importado.



Al hacer clic con el botón derecho en un archivo incluido o importado, aparece un menú contextual con opciones para eliminar el archivo o abrirlo en XMLSpy. El archivo se abre en la vista WSDL (archivo `.wsdl`) o en la vista Esquema (archivo `.xsd`), donde puede editarlo. Después de importarlos, los esquemas XML se pueden incrustar en el archivo WSDL (*ver imagen siguiente*).



Al incrustar un esquema se crea el esquema como esquema alineado dentro del elemento `types` y se elimina el elemento `import`. Para incrustar un esquema importado, haga clic con el botón derecho en el esquema de la lista *Imports* y seleccione el comando Incrustar esquema importado o Incrustar todos los esquemas. Este último comando, que afecta a todos los esquemas importados, también está disponible en el menú contextual del nodo *Imports*.

Types (WSDL 1.1 y 2.0):

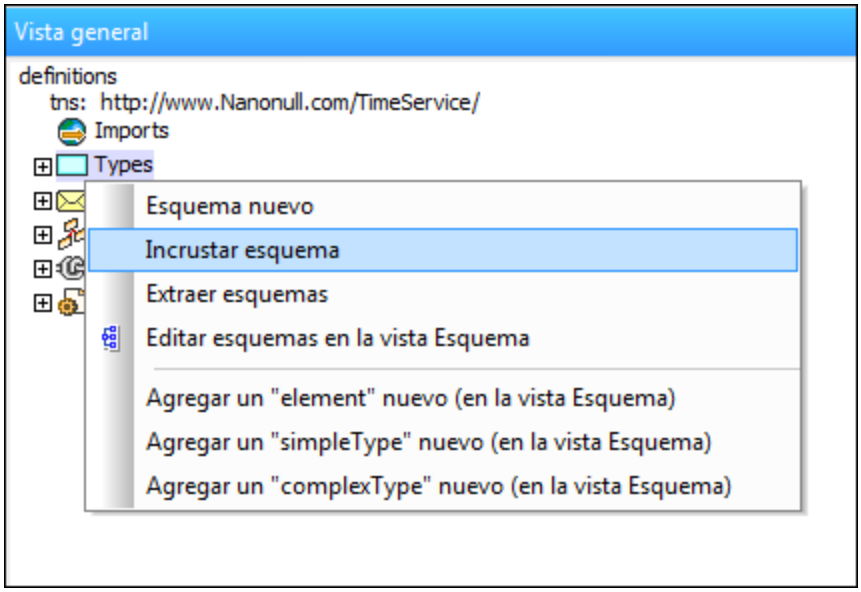
Ofrece una lista con todos los tipos definidos en el documento WSDL (en negro) y en los esquemas o documentos WSDL importados (en gris).

© 2016-2022 Altova GmbH



Estas son las funciones disponibles para los tipos:

- *Crear un esquema nuevo* haga clic con el botón derecho en el nodo *Types* y seleccione la opción **Esquema nuevo** (*ver imagen siguiente*). Los esquemas incrustados nuevos y vacíos se crean en el archivo WSDL y todos los esquemas incrustados, incluido el esquema nuevo vacío, se abren en la vista Esquema. Los esquemas incrustados nuevos y vacíos se crean en el archivo WSDL y todos los esquemas incrustados, incluido el esquema nuevo vacío, se abren en la vista Esquema. Cuando termine de editar el documento de esquema y guarde los cambios, estos cambios se escribirán también en el documento WSDL. Después de esto puede guardar el documento. Esta función es muy práctica si desea crear un esquema nuevo incrustado en el documento WSDL.



- *Incrustar esquemas:* haga clic con el botón derecho en el nodo *Types* y seleccione la opción **Incrustar esquema**. Se abre un cuadro de diálogo en el que puede buscar el archivo de esquema que desea incrustar. Al hacer clic en **Aceptar** se crea el esquema como esquema alineado dentro del elemento `types`. Si el esquema seleccionado ya se importó, aparece un diálogo preguntando si desea incrustar el esquema ya importando.

- *Extraer esquemas:* haga clic con el botón derecho en el nodo *Types* y seleccione la opción **Incrustar esquema**. Cada esquema incrustado se abre en un archivo temporal en la vista Esquema y por cada archivo se abre un cuadro de diálogo "Guardar como". Si decide guardar el archivo de esquema, el esquema se extrae, se guarda en la ubicación indicada en el diálogo y después se importa al archivo

WSDL. El archivo de esquema ya no existe como esquema alineado, sino como esquema externo importado.

- *Editar esquemas:* puede editar los esquemas incrustados en la vista Esquema. Haga clic con el botón derecho en el nodo *Types* o en un componente de esquema de la lista *Types* y después seleccione la opción **Editar esquemas** en la vista Esquema. Como resultado se genera un archivo XSD temporal a partir de las definiciones de `types` del documento WSDL. Este documento XSD se muestra en la vista Esquema, donde se puede editar. Cuando termine de editar el documento XSD y guarde los cambios, los cambios se guardarán en las definiciones de `types` del documento WSDL. Si cierra el documento XSD sin guardar los cambios, las definiciones de `types` del documento WSDL no se modificarán.

- *Incorporar componentes de esquema:* puede agregar un elemento element (WSDL 1.1 y 2.0), simpleType (WSDL 1.1) o complexType (WSDL 1.1). Para ello haga clic con el botón derecho en el nodo *Types* o en un componente de esquema de la lista *Types* y seleccione el correspondiente comando **Agregar**. Como resultado se genera un archivo XSD temporal a partir de las definiciones de `types` del documento WSDL y se abre en la vista Esquema. Este archivo contiene el componente nuevo y sin nombrar. Edite el componente en la vista Esquema y guarde el archivo temporal. El componente nuevo se escribirá en las definiciones de `types` del documento WSDL.

- *Eliminar componentes de esquema:* para eliminar un componente de esquema de la lista *Types*, haga clic con el botón derecho en el componente y seleccione la opción **Eliminar** del menú contextual.

Messages (WSDL 1.1)

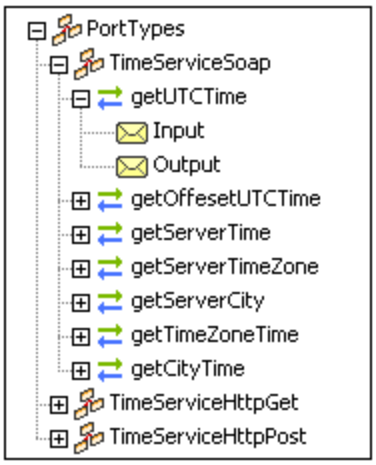
Cuando está seleccionado un mensaje o un subcomponente del mensaje, sus propiedades aparecen en el ayudante de entrada Detalles 263, donde se pueden editar. También puede realizar todas estas acciones:

- Haga clic con el botón derecho en un mensaje del ayudante de entrada Vista general. Aparece un menú contextual con opciones para agregar una parte de mensaje al mensaje, eliminar el mensaje o agregar un mensaje nuevo.
- Haga clic con el botón derecho en una parte de mensaje en el ayudante de entrada Vista general. Aparece un menú contextual con opciones para agregar otra parte de mensaje o eliminar la parte de mensaje seleccionada.
- El menú contextual incluye el comando **Sincronizar**, que resalta el mensaje o parte de mensaje en el recuadro del portType pertinente.

PortTypes (WSDL 1.1):

Los menús contextuales de los portTypes permiten realizar estas operaciones:

© 2016-2022 Altova GmbH

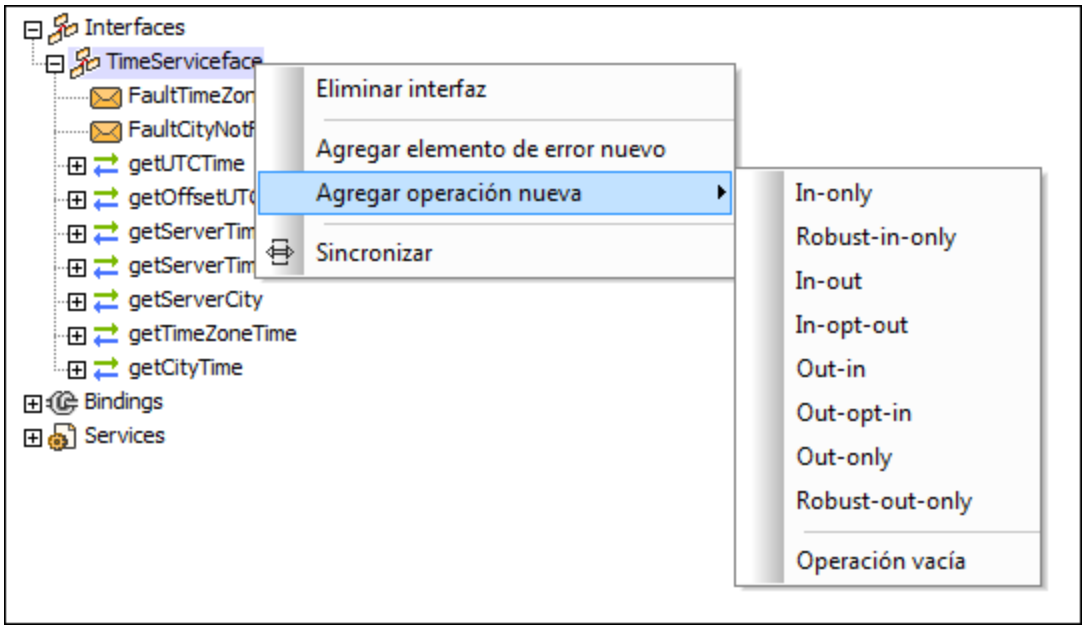


- Al seleccionar el nodo *PortTypes*, el menú contextual permite añadir un portType nuevo.
- Al seleccionar un portType, el menú contextual permite añadir portTypes nuevos, eliminar el portType seleccionado y añadir operaciones al portType seleccionado.
- Al seleccionar una operación, el menú contextual permite anexar más operaciones, eliminar la operación seleccionada y añadir elementos (de entrada, salida o error) a la operación seleccionada.
- Al seleccionar un elemento de mensaje (de entrada, salida o error), el menú contextual permite añadir más mensajes y eliminar el elemento de mensaje seleccionado.
- El comando **Sincronizar** del menú contextual resalta en la ventana principal el portType, la operación o el mensaje seleccionados.

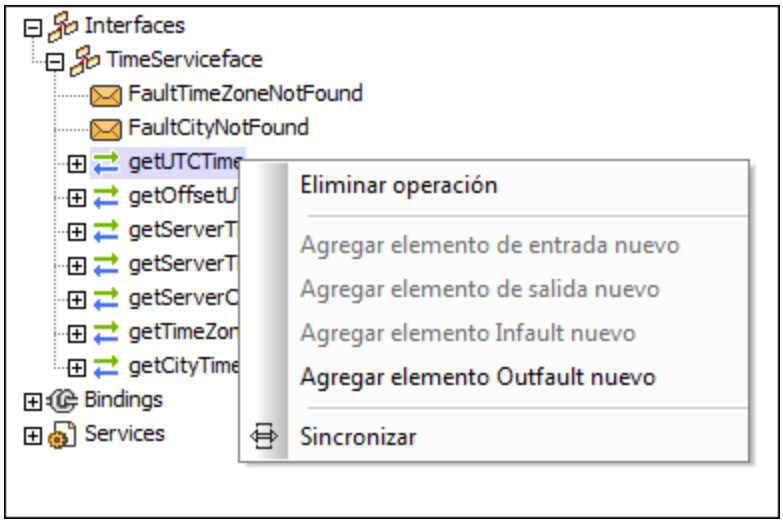
Interfaces (WSDL 2.0):

Los menús contextuales de las interfaces permiten realizar estas operaciones:

- Para añadir una interfaz, haga clic con el botón derecho en el nodo *Interfaces* y seleccione el comando **Agregar interfaz nueva** del menú contextual.
- Al hacer clic con el botón derecho en una interfaz aparece un menú contextual (*imagen siguiente*) con comandos para eliminar la interfaz seleccionada y añadir elementos de error (fault) u operaciones a la definición de la interfaz seleccionada. Utilice el submenú del comando **Agregar operación nueva** para especificar el tipo de operación que desea añadir. La operación de enlace nueva se añade a todos los enlaces que hacen referencia a la interfaz. Del mismo modo, si se elimina una operación, las operaciones de enlace que hacen referencia a la interfaz también se eliminan.



- Al hacer clic con el botón derecho en una operación aparece un menú contextual (*imagen siguiente*) con comandos para eliminar la operación seleccionada y añadir elementos (p. Ej. `infault` y `outfault`) a la operación.



- Al hacer clic con el botón derecho en un elemento de mensaje aparece un menú contextual con comandos para eliminar el mensaje seleccionado.
- El comando **Sincronizar** del menú contextual resalta en la ventana principal la interfaz, la operación o el mensaje seleccionados.

Bindings (WSDL 1.1 and 2.0)

A hacer clic con el botón derecho en un enlace, podrá anexar más enlaces, eliminar el enlace seleccionado e insertar operaciones para el enlace. Al hacer clic con el botón derecho en una operación o mensaje, estarán disponibles las mismas funciones que se describen en el subapartado dedicado a las operaciones y mensajes

 © 2016-2022 Altova GmbH

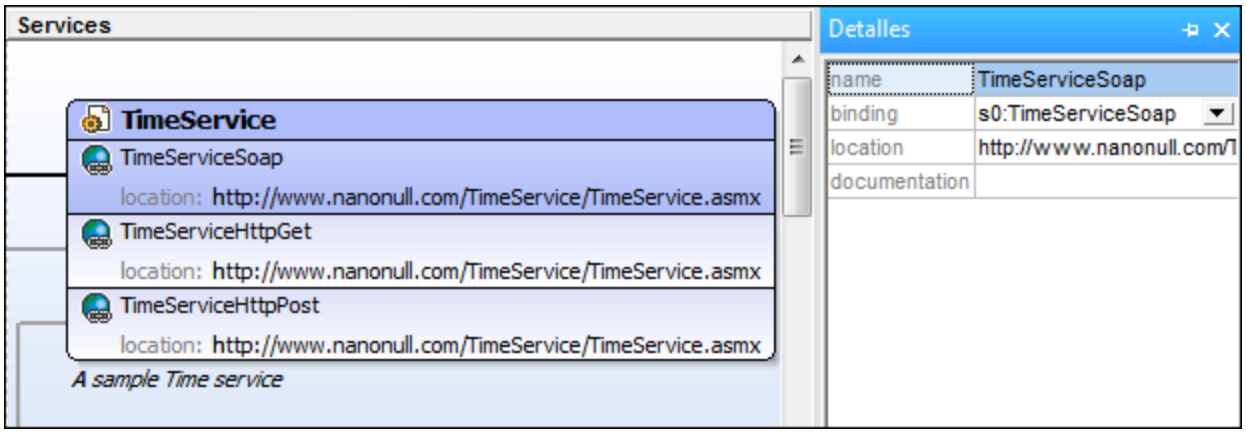
de la categoría PortTypes (WSDL 1.1) o Interfaces (WSDL 2.0). El comando **Sincronizar**resalta en la ventana principal el enlace, la operación o el mensaje seleccionados.

Services (WSDL 1.1 and 2.0)

A hacer clic con el botón derecho en un servicio, podrá añadir más servicios, eliminar el servicio seleccionado y añadir puertos al servicio. El comando **Sincronizar** resalta en la ventana principal el servicio o puerto seleccionado.

4.5.3 Ayudante de entrada de detalles

El ayudante de entrada Detalles presenta las propiedades del componente seleccionado tanto en la ventana principal como en el ayudante de entrada Vista general (*imagen siguiente*). Estas propiedades se pueden editar en el ayudante de entrada Detalles directamente.



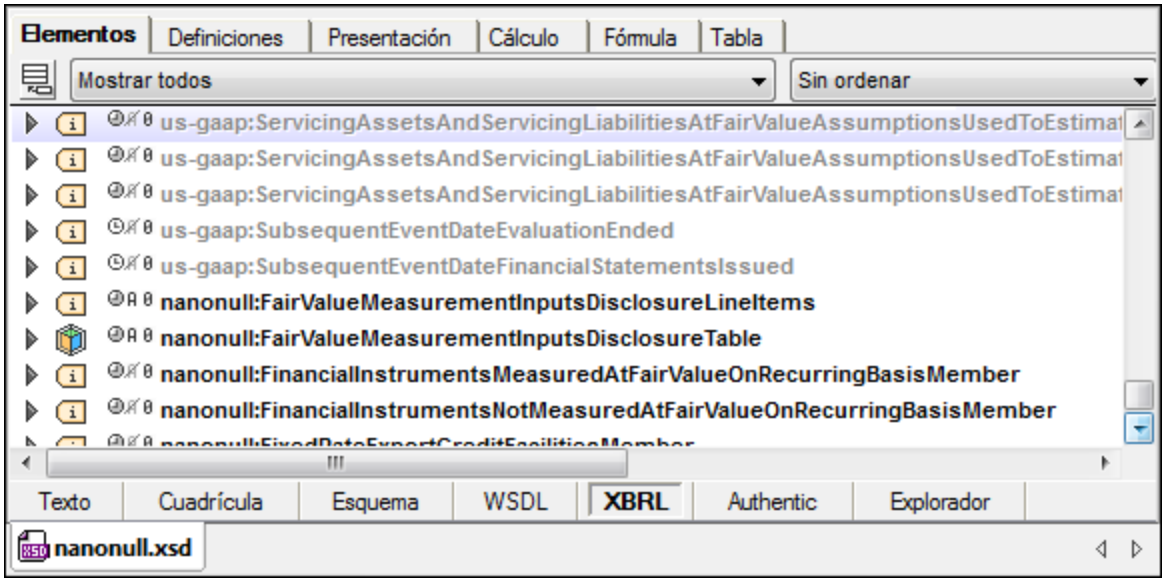
En este ejemplo vemos que al seleccionar el puerto `TimeServiceSoap` (del servicio `TimeService`) en la ventana principal, las propiedades de `TimeServiceSoap` se presentan en el ayudante de entrada Detalles, donde se pueden editar. Para editar un campo de texto como `name` o `description`, haga doble clic en el campo y edite el texto. Algunas propiedades, como la propiedad `binding` de este ejemplo, incluyen un cuadro combinado con varias opciones para que seleccione la opción deseada.

4.6 Vista XBRL

La vista XBRL es una interfaz gráfica que sirve para editar taxonomías XBRL. La vista ofrece mensajes de error y validación rápida, que le ayudarán a desarrollar taxonomías con rapidez y precisión.

La vista XBRL está formada por estos componentes:

- Una ventana principal dividida en seis pestañas: Elementos 265, Definiciones 269, Presentación 269, Cálculo 269, Fórmula 269 y Tabla 269 (*imagen siguiente*). Las principales características de estas pestañas se describen en los diferentes apartados de esta sección.



- Tres potentes ayudantes de entrada 272: Vista general, Elementos globales y Detalles. Estos ayudantes de entrada le permitirán gestionar los archivos de taxonomía y editar la taxonomía en la ventana principal.
- Una ventana de mensajes 125 que presenta los mensajes de validación de la taxonomía.

Esta sección describe la ventana principal y los ayudantes de entrada de la vista XBRL y ofrece información sobre cómo utilizarlos. Para más información, consulte la sección XBRL 817 y la descripción de los comandos del menú XBRL 1515.

Funciones adicionales de la vista XBRL

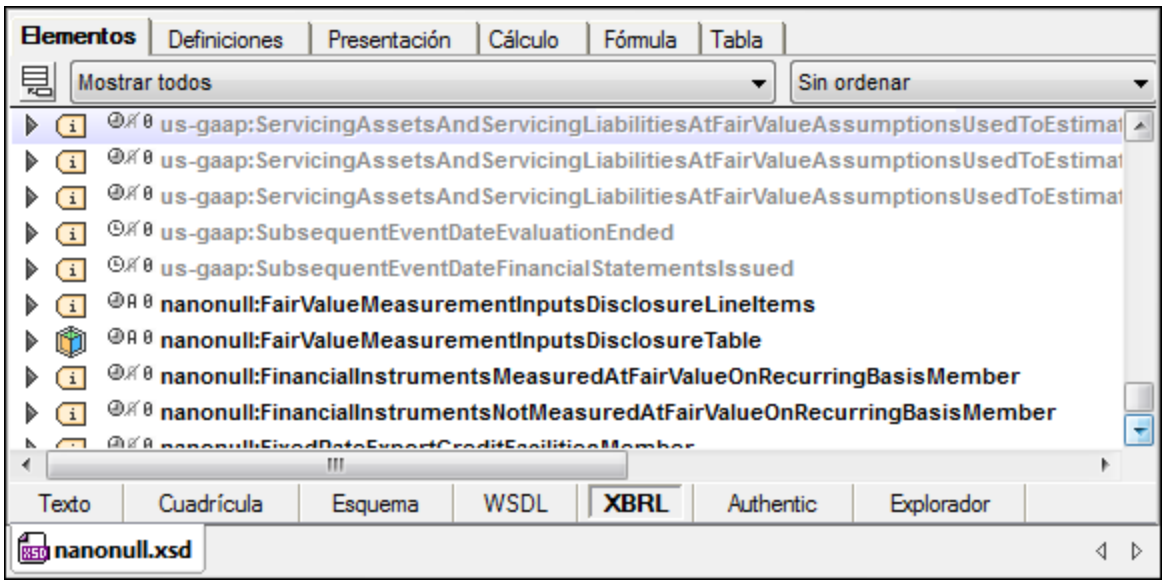
Además de las funciones de edición que se describen en esta sección, la vista XBRL también ofrece funciones para:

- Generar documentación 1523: cree documentos detallados de documentación en formato HTML, Word y RTF.
- Imprimir la vista actual 1275: imprima la vista actual con el comando **Archivo | Imprimir** de XMLSpy.

© 2016-2022 Altova GmbH

4.6.1 Ventana principal: Elementos

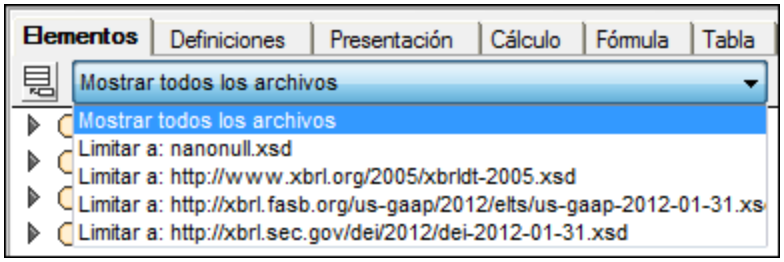
La pestaña **Elementos** de la ventana principal (*imagen siguiente*) presenta los conceptos de la taxonomía, incluidos los conceptos de taxonomías importadas. Los conceptos de la taxonomía actual aparecen en negro, mientras que los de las taxonomías importadas aparecen en gris.



Para más información consulte la sección XBRL[817] y la descripción de comandos en el Menú XBRL[1515].

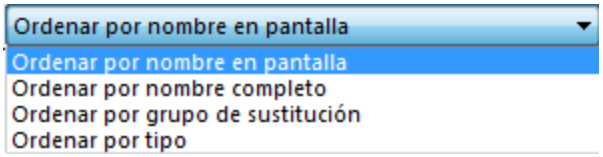
Seleccionar la taxonomía que desea visualizar

En el cuadro combinado *Archivo* situado a la izquierda, debajo de las pestañas de la ventana principal (*imagen siguiente*), puede seleccionar si desea visualizar sólo los conceptos de la taxonomía actual o los conceptos de la taxonomía actual y de las taxonomías importadas. Seleccione *Mostrar todos los archivos* para visualizar las taxonomías importadas. La edición será más rápida si quita de la vista (filtra) las taxonomías importadas de mayor tamaño.



Ordenar elementos

En el cuadro combinado *Ordenar* situado a la derecha, debajo de las pestañas de la ventana principal (*imagen siguiente*), puede ordenar los elementos en la ventana principal.



Estos son los criterios de ordenación disponibles:

- *Ordenar por nombre en pantalla:* los elementos aparecen por orden alfabético, tomando su nombre como criterio de ordenación. Los nombres incluyen el prefijo. P. ej. **abc:yname** aparecerá antes de **bcd:xname**.
- *Ordenar por nombre completo:* los nombres de elemento se resuelven completamente (es decir, sus espacios de nombres se expanden) y los nombres expandidos aparecen por orden alfabético. Así que si el prefijo de espacio de nombres **abc** está vinculado con el espacio de nombres **http:people.altova.com**, entonces el nombre de elemento **abc:yname** se resolverá como **http:people.altova.com:yname**.
- *Ordenar por grupo de sustitución:* hay cuatro grupos de sustitución: ítems, tuplas, hypercubeItem (hipercubo), dimensionItem (dimensión).
- *Ordenar por tipo:* hace referencia al atributo `type` del elemento XBRL.

Buscar elementos en la ventana principal

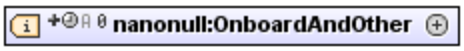
Para buscar un elemento en la ventana principal, pulse la combinación de teclas **Ctrl+F**. Aparece el cuadro de diálogo Buscar, donde puede escribir la cadena de texto que desea buscar. También puede buscar en los prefijos de espacio de nombres usados en la taxonomía.

Conceptos (los elementos <element>)

Cada concepto de la taxonomía se define en un elemento `<element>` de un esquema XML (*ver el código que aparece a continuación*). Puede ver esta definición de texto sin formato si cambia a la vista Texto 145 del documento de taxonomía. (Para ver las definiciones de texto de un documento de taxonomía importado, es necesario abrir el documento de taxonomía importado en XMLSpy.)

```
<xs:element id="icui_UnrealizedHoldingLoss"
       name="UnrealizedHoldingLoss"
       substitutionGroup="xbrli:item"
       type="xbrli:monetaryItemType"
       xbrli:balance="credit"
       xbrli:periodType="instant"
       abstract="false"
       nillable="true"/>
```

Cada elemento (o concepto) se presenta en la pestaña Elementos con un icono que indica su grupo de sustitución (ítem, tupla, hipercubo o dimensión). Además, a la izquierda del nombre del concepto se incluyen iconos que indican los valores de los atributos `balance`, `periodType`, `abstract` y `nillable` del concepto (*ver imagen siguiente*).



Para editar el nombre del concepto, haga doble clic en su nombre y edítelo.

 © 2016-2022 Altova GmbH

Grupo de sustitución (substitutionGroup)

El valor del grupo de sustitución de un concepto se indica mediante estos iconos:

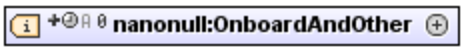

`xbrli:item`


`xbrli:tuple`


`xbrldt:hypercubeItem`


`xbrldt:dimensionItem`

La imagen siguiente muestra un elemento cuyo `substitutionGroup` tiene el valor `xbrli:item`.



Atributos **balance**, **periodType**, **abstract** y **nillable**

El resto de iconos situados a la izquierda del nombre del elemento (*ver imagen anterior*) indica los valores de los principales atributos del concepto, que son, de izquierda a derecha:

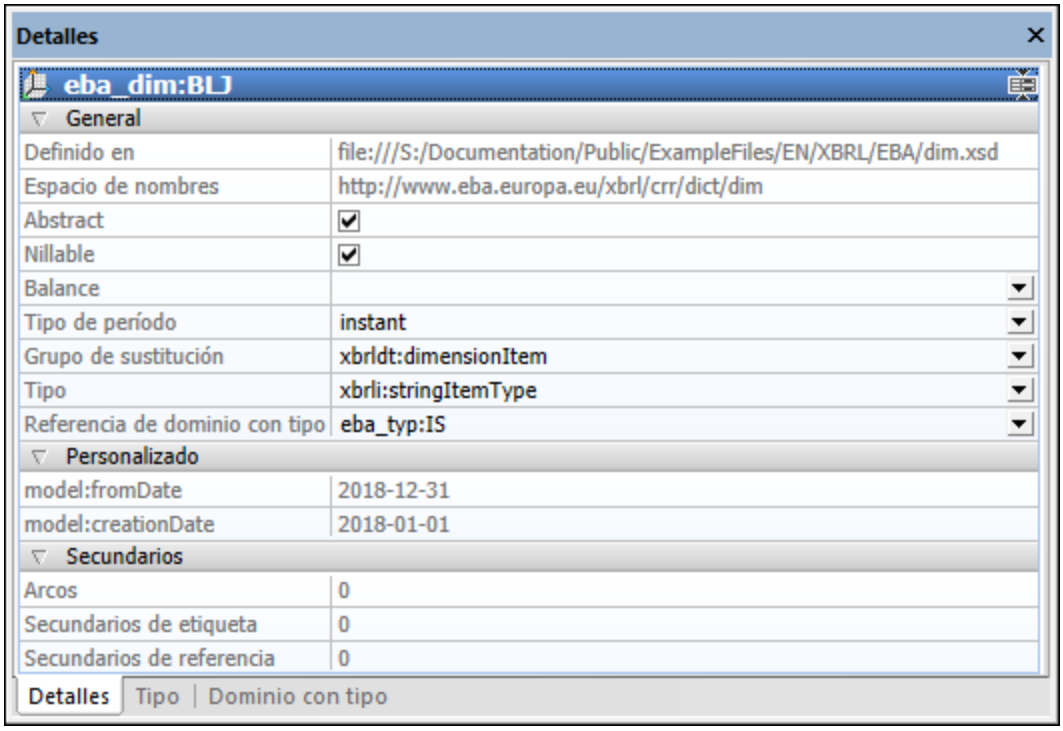
- Atributo `xbrli:balance`: con un icono + si el valor es `credit` y un icono - si el valor es `debit`
- Atributo `xbrli:periodType`: con un icono en forma de reloj de color gris si el valor es `duration` y un reloj de color blanco si el valor es `instant`
- Atributo `xs:abstract`: un icono `A` de color negro si el valor es `true` y de color gris si el valor es `false`
- Atributo `xs:nillable`: un icono `0` de color negro si el valor es `true` y de color gris si el valor es `false`

Tenga en cuenta que los atributos `xbrli:` que acabamos de mencionar son atributos del esquema XBRL, mientras que los atributos `xs:` son del esquema XML (XSD).

En la imagen anterior, el icono + indica que el valor del atributo `xbrli:balance` es `credit`. Los valores de los demás atributos del concepto de la imagen anterior son: `duration` (para el atributo `xbrli:periodType`), `false` (para el atributo `xs:abstract`, es decir no es abstracto) y `true` (para el atributo `xs:nillable`, es decir puede ser nulo).

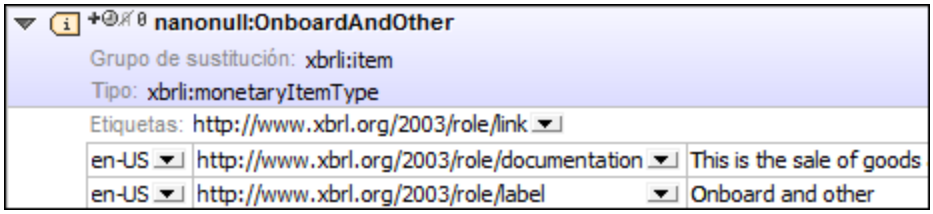
Los valores de estos cuatro atributos se presentan en un mensaje emergente de información rápida al pasar el puntero por encima de alguno de los iconos. Al hacer clic en un icono, aparece un cuadro de lista con los valores permitidos para ese atributo. Puede seleccionar uno de los dos valores permitidos y terminar rápidamente de editar el valor del atributo.

Estos valores de atributo también se pueden editar en el ayudante de entrada Detalles (*ver imagen siguiente*).



Atributos `substitutionGroup` y `type`

Al hacer clic en el icono en forma de punta de flecha situado a la izquierda del nombre del elemento, se expande la presentación del elemento y se ofrece una representación gráfica de los valores de los atributos `xs:substitutionGroup` y `xs:type` (*ver imagen siguiente*).



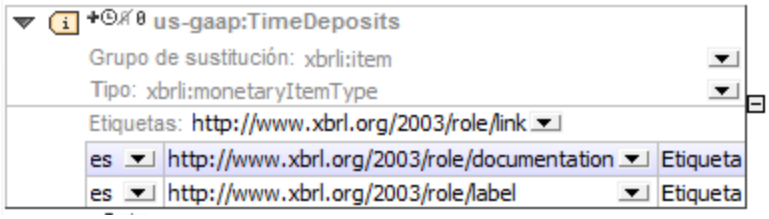
En la imagen anterior puede observar que el valor de los atributos `xs:substitutionGroup` y `xs:type` es `xbrli:item` y `xbrli:monetaryItemType` respectivamente. Estos valores se pueden editar con solo seleccionar otro valor distinto en la lista desplegable del cuadro combinado de cada atributo. Los valores de estos atributos también se pueden editar en el ayudante de entrada Detalles (*ver la imagen anterior a esta*). Para ello, haga doble clic en el campo del valor pertinente en el ayudante de entrada Detalles e introduzca los datos.

Vínculos de etiqueta (Label)

Para agregar un elemento secundario `label` (etiqueta) haga clic con el botón derecho en un elemento y seleccione el comando **Agregar función de vínculo de etiqueta** en el menú contextual. Se añade un recuadro para la etiqueta (*ver imagen siguiente*) y en él puede indicar las propiedades de la etiqueta (el idioma,

 © 2016-2022 Altova GmbH

el tipo de etiqueta y su valor). Puede expandir y contraer las etiquetas con los iconos + y - situados en el borde derecho del recuadro.



Las propiedades de idioma y tipo de etiqueta de cada etiqueta se pueden editar con solo seleccionar el valor deseado para la propiedad en los cuadros combinados (*ver imagen anterior*).

Vínculos de referencia (Reference)

Para agregar un elemento secundario `reference` (referencia) haga clic con el botón derecho en un elemento y seleccione el comando **Agregar función de vínculo de referencia** en el menú contextual. Se añade un recuadro para la referencia, donde se puede editar la propiedad URI de la referencia. Puede expandir y contraer las referencias con los iconos + y - situados en el borde derecho del recuadro.

4.6.2 Ventana principal: Definiciones, Presentación, Cálculo, Fórmula y Tabla

La ventana principal contiene una pestaña por cada tipo de relación existente entre los conceptos:

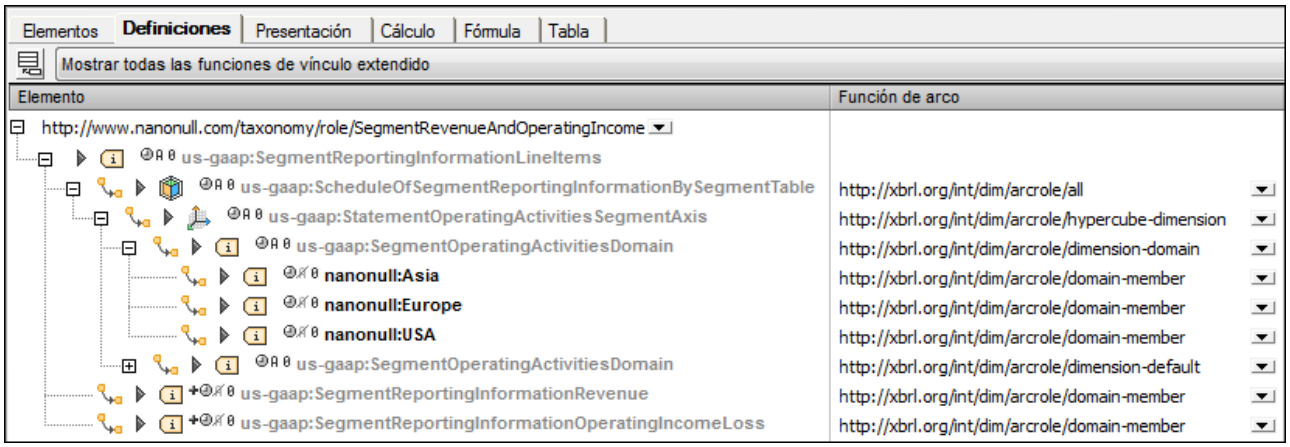
- Relaciones de definición, que aparecen en la pestaña **Definiciones**.
- Relaciones de presentación, que se muestran en la pestaña **Presentación**.
- Relaciones de cálculo, que se presentan en la pestaña **Cálculo**.
- Definiciones y relaciones de fórmulas, que se presentan en la pestaña **Fórmula**.
- Definiciones de tabla, que se presentan en la pestaña **Tabla**.

  Cada pestaña (Definición, Presentación, Cálculo, Fórmula y Tabla) muestra las relaciones de la taxonomía de ese tipo concreto (*ver imagen siguiente*) y sirven para editar las relaciones de forma gráfica. La relación entre dos conceptos (ya sea una relación de definición, presentación o cálculo) se crea al construir un arco desde un concepto a otro. Este arco de un concepto a otro se representa gráficamente en la vista con una flecha curva . La relación entre los dos conceptos se conoce con el nombre de *función de arco*. La función de arco de un arco se muestra en el ayudante de entrada Detalles si está seleccionado el destino de la relación en la columna "Función de arco" de la pestaña **Definiciones** (*ver imagen siguiente*).

Los arcos de relación se representan con el mismo icono en todos los tipos de relación (definición, presentación y cálculo). En este apartado, describiremos por separado cada pestaña pero damos una descripción básica de la presentación de las relaciones sólo en el subapartado dedicado a la pestaña **Definiciones**. Además ofrecemos información sobre los arcos de presentación y cálculo en los subapartados respectivos. Para más información consulte las secciones Crear relaciones: Parte 1 853 y Crear relaciones: Parte 2 855.

Pestaña Definiciones

La pestaña **Definiciones** muestra todas las definiciones de la taxonomía. Estas definiciones se especifican en arcos de definición que se ubican en archivos XML de relaciones de definición. En la pestaña **Definiciones** de la vista **XBRL**, la estructura resultante del conjunto de arcos de definición se presenta en forma de árbol, cuyos nodos se pueden expander o contraer (*ver imagen siguiente*).



En esta presentación gráfica de las definiciones, cada arco de definición se representa con una flecha curva con dos extremos (un extremo de *origen* y un extremo de *destino*). El tipo de relación existente entre los dos elementos en ambos extremos se muestra en la columna *Función de arco* del elemento de *destino*. Por ejemplo, en una relación `hypercube-dimension` (hipercubo-dimensión), la relación o función de arco aparece en el elemento `dimension` de la pareja de elementos. Los URI de la función de arco también se pueden introducir/editar en el ayudante de entrada Detalles.

Para más información sobre relaciones de definición, consulte la sección Crear relaciones: Parte 1 853.

Pestaña Presentación

Los arcos de presentación tienen los mismos atributos que los arcos de definición y funcionan de la misma manera (consulte el subapartado anterior). Los URI de la función de arco se introducen/editan en el ayudante de entrada Detalles. Un atributo importante de presentación es el atributo `order`, que determina el orden en que se presenta el elemento secundario dentro de su elemento primario. En la pestaña **Presentación** estos elementos secundarios se presentan en orden ascendente. El valor del atributo `order` se puede cambiar en cualquier momento con sólo arrastrar un elemento secundario a otra posición de la lista. El valor del atributo `order` también se puede cambiar en el ayudante de entrada Detalles (*ver imagen siguiente*).

Para más información sobre las relaciones de presentación, consulte la sección Crear relaciones: Parte 2 855.

Pestaña Cálculo

Los arcos de cálculo tienen los mismos atributos que los arcos de definición y funcionan de manera parecida a los arcos de definición (consulte el subapartado anterior sobre definiciones). Los URI de la función de arco se introducen/editan en el ayudante de entrada Detalles. Hay dos tipos de funciones de arco:

- las funciones de arco que suman los valores de un elemento a los de otro elemento
- las funciones de arco que no representan una relación de suma, sino de equivalencia

© 2016-2022 Altova GmbH

En el caso de las funciones de arco de suma, el atributo `weight` determina qué cantidad del valor del elemento se suma al elemento de agregación. Si el atributo `weight` tiene el valor `1.0`, se suma el 100% del valor. Si el atributo `weight` tiene un valor negativo, el valor se resta del elemento de agregación. El valor del atributo `weight` se puede editar en la pestaña **Cálculo** o en el ayudante de entrada Detalles.

Pestaña Fórmula

En la pestaña **Fórmula** puede definir y gestionar fórmulas XBRL. Esta pestaña se usa en combinación con los ayudantes de entrada Vista general y Detalles para crear y editar fórmulas. Las definiciones y las relaciones entre componentes de las fórmulas se editan y crean en el diagrama de la ventana principal. Para más información consulte la sección Editor de fórmulas XBRL 864.

Pestaña Tabla

Las tablas permiten definir de otra forma las vistas de conceptos definidos en taxonomías XBRL. Estas definiciones están dentro de bases de enlaces de tabla, que se pueden crear y editar en la pestaña Tabla. Para aprender a editar bases de enlaces de tabla consulte la sección Editor de definiciones de tabla XBRL 889.

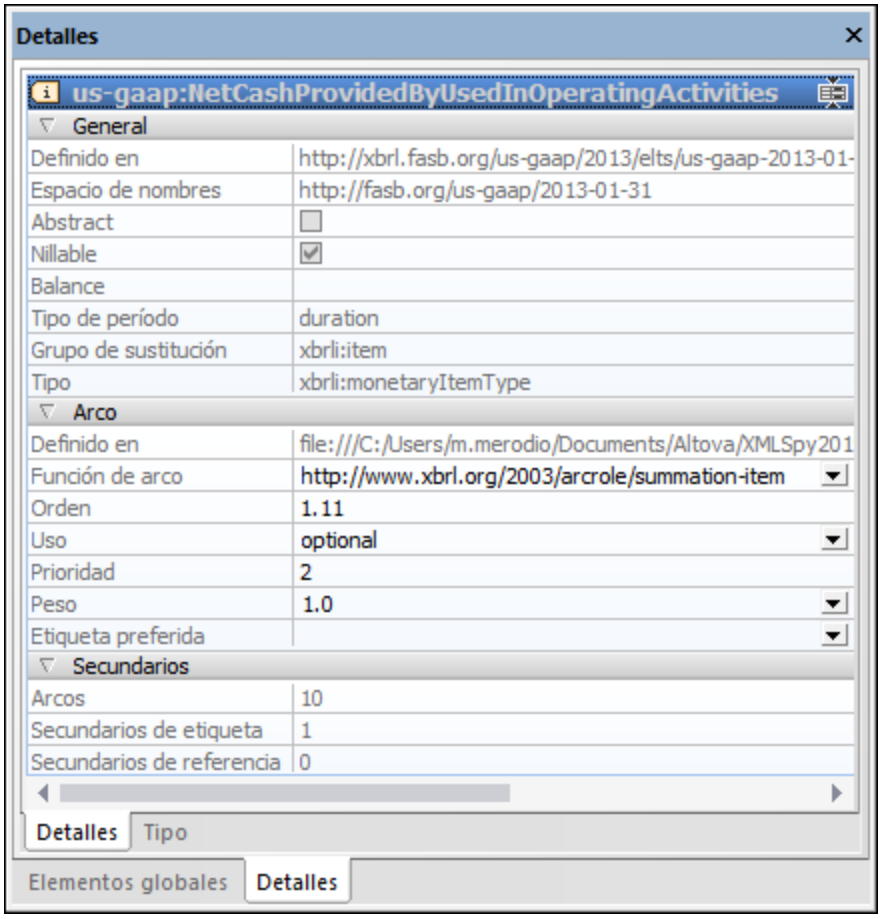
Edición en la vista gráfica

Estas son las funciones de edición disponibles:

- Puede arrastrar elementos desde el ayudante de entrada Elementos globales hasta una relación de destino de un elemento del árbol de la ventana principal.
- Puede crear una función de arco o editar una función de arco ya existente (selecciónela en la columna *Función de arco*). La relación definida en la función de arco expresa una relación de un origen a un destino, cuyo destino es el elemento seleccionado.
- Puede arrastrar los elementos a otra posición del árbol. Esta es la manera más rápida de cambiar el valor del atributo `order`.
- Puede editar las propiedades de un elemento con solo hacer clic en los símbolos de sus propiedades y editarlos. También puede expandir su recuadro de propiedades y editarlas dentro del recuadro.

Edición en el ayudante de entrada Detalles

Cuando está seleccionado un elemento en la ventana principal, las propiedades de su arco de definición se presentan en el ayudante de entrada Detalles, en el panel *Arco* (*ver imagen siguiente*). Los valores de estas propiedades se pueden editar con solo hacer doble clic en el campo del valor y escribir el valor deseado. Algunas propiedades también tienen un cuadro combinado con una lista desplegable de valores permitidos.



Asimismo, las propiedades del arco del elemento seleccionado se presentan en el panel *Arco* del ayudante de entrada Detalles. El arco será un arco de definición, presentación o cálculo dependiendo de la pestaña que esté activa en cada momento. La función de arco se puede introducir en el campo *Función de arco*.

Las relaciones de etiqueta y referencia se presentan en el panel *Secundarios* del ayudante de entrada Detalles (*imagen anterior*). Estas relaciones se pueden editar en la pestaña **Elementos**.

Para más información consulte la sección XBRL 817.

4.6.3 Ayudantes de entrada de la vista XBRL

La vista **XBRL** ofrece estos tres ayudantes de entrada:

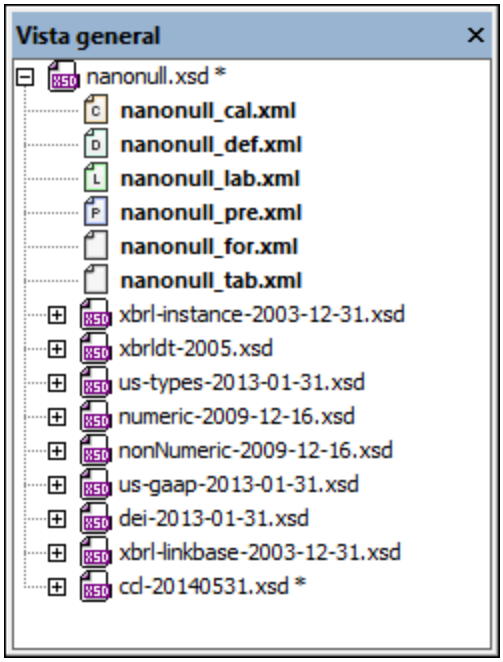
- El ayudante de entrada Vista general 273.
- El ayudante de entrada Elementos globales 274.
- El ayudante de entrada Detalles 276.

También ofrece una ventana de mensajes 125 donde se ofrece información sobre el proceso de validación. Para más información consulte la sección XBRL 817.

© 2016-2022 Altova GmbH

Ayudante de entrada Vista general

El ayudante de entrada Vista general presenta los archivos de la taxonomía[835] en una estructura en forma de árbol (*ver imagen siguiente*).



El árbol se organiza de la siguiente manera:

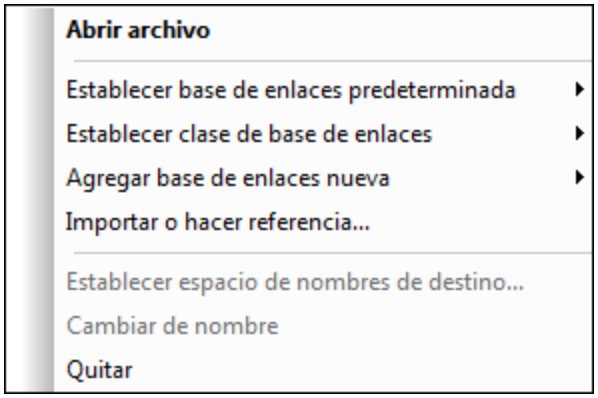
- El principal archivo de conceptos (un documento de esquema XML) aparece como elemento raíz del árbol y representa el archivo activo.
- Los archivos de bases de enlaces de relación (archivos XML) llevan un icono de archivo cuyo color e inicial depende del tipo de relación:

  indica una base de enlaces de definición (*Definition*);
  indica una base de enlaces de etiqueta (*Label*);
  indica una base de enlaces de cálculo (*Calculation*);
  indica una base de enlaces de presentación (*Presentation*);
  indica una base de enlaces de referencia (*Reference*) y
  indica una base de enlaces de fórmula (*Formula*).

- Los esquemas importados aparecen al final de la jerarquía. Los archivos de esquema XML (`.xsd`) llevan un icono con las letras XSD.

La información del ayudante de entrada Vista general se obtiene del elemento `/schema/annotation/appinfo` del archivo de conceptos principal. Para más información sobre este elemento, consulte la sección sobre archivos de taxonomía[835].

Al hacer clic con el botón derecho en un archivo del ayudante de entrada Vista general, aparece un menú contextual (*imagen siguiente*) con estas opciones:

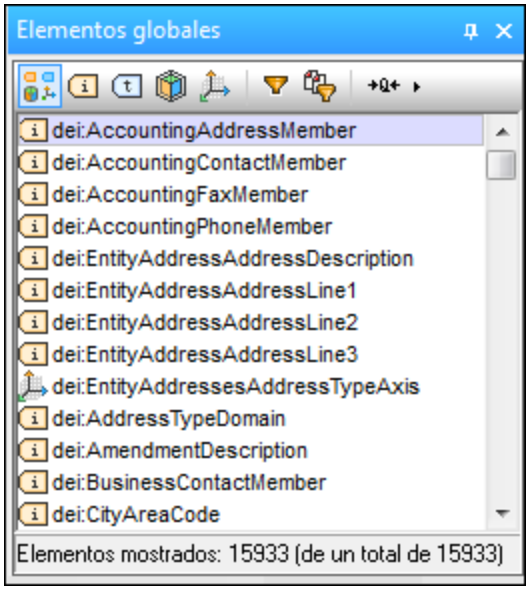


- *Abrir archivo:* abre el archivo XML o de esquema XML seleccionado en XMLSpy.
- *Establecer base de enlaces predeterminada:* si hay varios archivos de una clase de relación (por ejemplo, relaciones de presentación), puede establecer uno de esos archivos como base de enlaces predeterminada. A partir de ese momento, las relaciones nuevas se guardarán en la base de enlaces predeterminada de esta clase de relación. La base de enlaces predeterminada de cada clase de relación aparece en negrita.
- *Establecer clase de base de enlaces:* especifica la clase de relación de la base de enlaces seleccionada. En el submenú seleccione la clase de relación deseada. El comando **Todas** del submenú indica que la base de enlaces seleccionada se debe usar para todas las clases de relaciones.
- *Agregar base de enlaces nueva:* crea un archivo de base de enlaces nuevo que contiene una de las cinco clases de relación (definición, presentación, cálculo, etiqueta, referencia). Puede cambiar el nombre del archivo recién creado. Para ello, haga clic en él con el botón derecho y seleccione el comando **Cambiar de nombre**. Se añade un elemento `linkbaseRef` nuevo al archivo de conceptos. Este elemento hace referencia a la base de enlaces recién creada.
- *Importar o hacer referencia*: importa una taxonomía estándar o crea una referencia a un esquema XML o base de enlaces ya existente. Si selecciona la opción "Importar taxonomía estándar", aparece una lista con las taxonomías estándar. Seleccione la taxonomía que desea importar. Para más información, consulte la sección Importar una taxonomía 839. Si crea una referencia a un esquema XML o a una base de enlaces, se añade al archivo de conceptos un elemento `linkbaseRef` nuevo que contiene la referencia. Si elige la opción "Esquema de referencia" o la opción "Base de enlaces de referencia", puede examinar el sistema de archivos y encontrar el archivo deseado.
- *Establecer espacio de nombres de destino:* establece el espacio de nombres de destino y declara este espacio de nombres en el ámbito del elemento `xs:schema` (es decir, para toda la taxonomía). Para más información, consulte la descripción del comando **XBRL | Establecer espacio de nombres de destino** 1519.
- *Cambiar de nombre:* permite cambiar el nombre del archivo seleccionado.
- *Quitar:* quita el archivo seleccionado de la vista general y elimina el elemento `linkbaseRef` del archivo de conceptos.

Ayudante de entrada Elementos globales

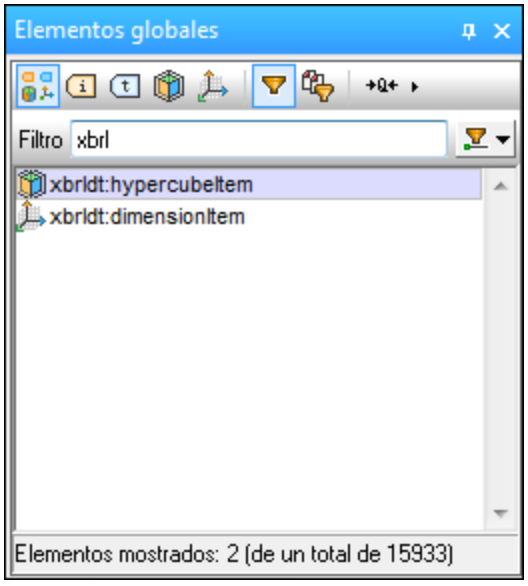
El ayudante de entrada Elementos globales (*imagen siguiente*) muestra todos los ítems, tuplas, hipercubos y dimensiones presentes en un documento de taxonomía. Los elementos se pueden arrastrar desde el ayudante de entrada Elementos globales hasta la ventana principal.

© 2016-2022 Altova GmbH



Las funciones disponibles en el ayudante de entrada Elementos globales son:

- *Filtrar según el tipo:* puede filtrar el tipo de elemento que muestra el ayudante de entrada (ítems, tuplas, hipercubos, dimensiones) si pulsa el icono correspondiente en la barra de herramientas del ayudante de entrada Elementos globales. Si hace clic en el icono *Mostrar todos los elementos* , el ayudante de entrada muestra todos los elementos. El filtro seleccionado aparece siempre resaltado (*Mostrar todos los elementos* está resaltado en la imagen siguiente).
- *Filtrar según el texto*: si pulsa el icono *Filtro de texto*  aparece la barra del filtro de texto (*ver imagen siguiente*).



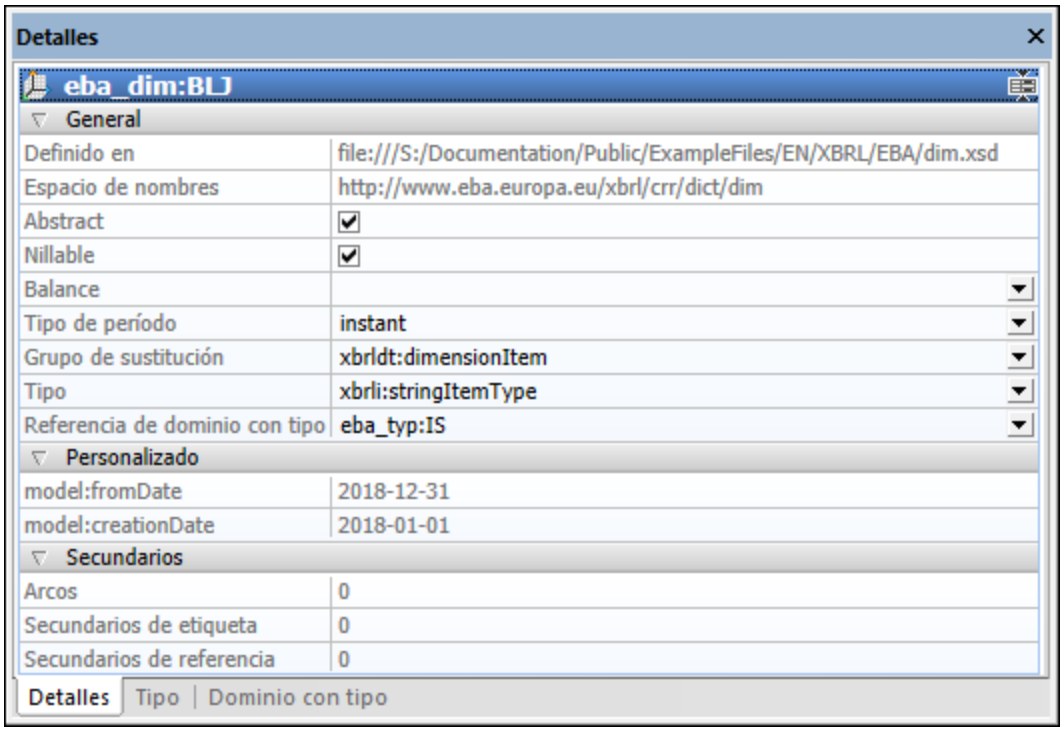
Después de seleccionar el filtro de texto, en el cuadro combinado situado a la derecha de la barra del filtro de texto (*con un icono en forma de embudo*) seleccione una condición con la que se deben filtrar los elementos. Las opciones disponibles son *Contiene*, *No contiene* y *Empieza con*. En la imagen

anterior, se seleccionó la condición *Empieza con.* A continuación, escriba el texto para el filtro en el cuadro de texto del filtro. Los elementos se filtrarán de acuerdo con estas condiciones. En la imagen anterior, se filtran los elementos que empiezan con el texto `xbrl`. Recuerde que (i) los filtros funcionan como cadenas de texto y se aplican al nombre de los elementos y (ii) los nombres de los elementos se pueden cambiar con ayuda del icono *Formato* (*ver imagen siguiente*).

- *Filtrar según la fuente:* si pulsa el icono *Fuentes*, aparece una pequeña ventana emergente donde puede seleccionar las fuentes que le interesan. El ayudante de entrada mostrará sólo los elementos de las fuentes seleccionadas.
- *Formato:* hay tres opciones de formato para el modo de presentación del nombre de los elementos. Las opciones son: *Nombre completo abreviado*, *Nombre completo expandido* y *Etiquetas*. El nombre completo abreviado utiliza los prefijos asignados a los respectivos espacios de nombres. El nombre completo expandido (cuyo icono es ) utiliza el espacio de nombres completo. La etiqueta usa la etiqueta asociada al elemento. Tenga en cuenta que la opción de formato seleccionada afectará a la opción de *filtro de texto*. Es decir, el filtro de texto se aplicará a los elementos presentes en la lista después de filtrarlos según la opción de formato elegida.
- *Operaciones de arrastrar y colocar:* puede arrastrar y colocar elementos desde el ayudante de entrada Elementos globales hasta las relaciones de las vistas de relaciones de la ventana principal (ya sea en la pestaña Definiciones, Presentación o Cálculo).

Ayudante de entrada Detalles

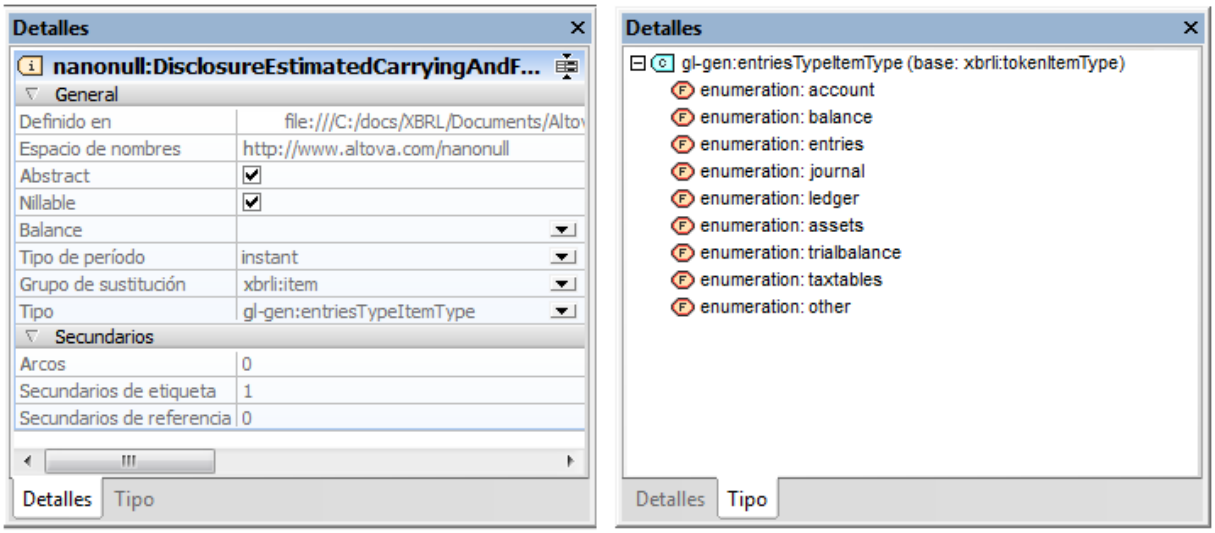
Cuando se selecciona un elemento en la ventana principal, el ayudante de entrada Detalles (*imagen siguiente*) muestra sus propiedades. Si el elemento tiene atributos personalizados, entonces el ayudante de entrada muestra una sección llamada *Personalizados* en la que aparecen esos atributos.



Las propiedades de algunos elementos se pueden editar en el ayudante de entrada Detalles directamente: por ejemplo, *Abstract*, *Nillable*, *Balance*, *periodType*, *substitutionGroup* y *Type*.

© 2016-2022 Altova GmbH

El ayudante de entrada Detalles también incluye una pestaña llamada *Tipo* que muestra el tipo del elemento seleccionado como nodo raíz de una vista en forma de árbol (*imagen siguiente*). Para lo conceptos de tipo `enum:enumerationItemType` pueden especificarse extensiones extensibles con etiquetas multilingües tal y como se define en la recomendación Extensible Enumerations del 29 de octubre de 2014. Para los elementos de este tipo se ofrecen entradas adicionales para los atributos `enum:domain`, `enum:linkrole` y `enum headUsable`.



Si el tipo no es `xbrli-item-type` ni un tipo XSD integrado, el ayudante de entrada muestra información sobre el tipo base entre paréntesis. La vista en forma de árbol también ofrece un menú contextual con opciones para ver o modificar la definición de tipo del concepto en la vista **Esquema**.

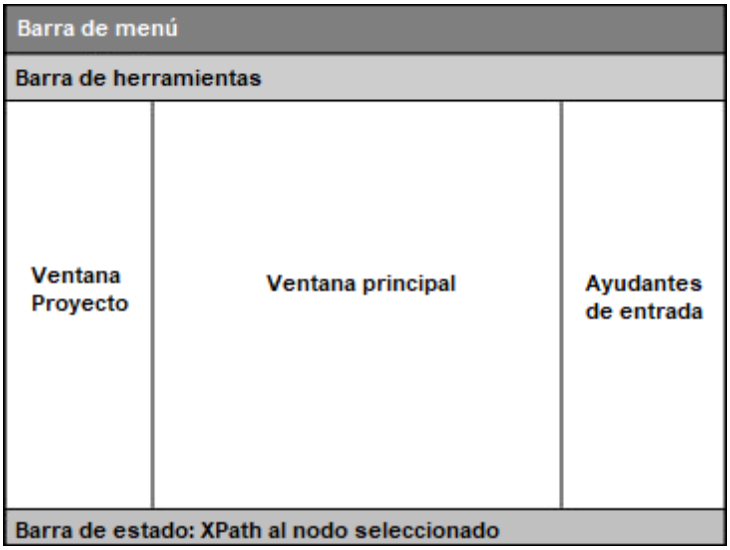
4.6.4 Configuración de la vista XBRL

En la vista XBRL se pueden configurar dos tipos de opciones:

- Las fuentes y los colores de los componentes de la vista XBRL. Puede acceder a estas opciones desde la sección *Fuentes y colores* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**[1612]).
- El formato y el diseño de la vista, así como los valores predeterminados de las etiquetas. Estas opciones se encuentran en el cuadro de diálogo "Configurar la vista XBRL"[1527], al que se accede con el comando **XBRL | Configurar la vista XBRL**.

4.7 Vista Authentic

La vista Authentic tiene una barra de herramientas y una barra de menús en la parte superior de la ventana. El resto de la interfaz se divide en tres grandes secciones: la ventana Proyecto, la ventana principal y la ventana de los ayudantes de entrada (*imagen siguiente*).



La interfaz de la vista Authentic[613] se describe con detalle en el apartado **Authentic**[597] | **Interfaz de la vista Authentic**[613].

© 2016-2022 Altova GmbH

4.8 Vista Explorador

La vista **Explorador** se suele usar para ver:

- Archivos XML que tienen un archivo XSLT asociado. Al cambiar a la vista **Explorador**, el archivo XML se transforma instantáneamente usando la hoja de estilos XSLT asociada y el resultado se presenta directamente en la vista **Explorador**.
- Archivos HTML que se crearon directamente como HTML o a partir de un archivo XML transformado con una hoja de estilos XSLT.

Para ver archivos XML y HTML en la vista **Explorador**, basta con hacer clic en la pestaña **Explorador**.

Motores de navegación en la vista Explorador

Por defecto, la vista Explorador utiliza actualmente Internet Explorer de Microsoft como motor del explorador. Si desea usar el nuevo motor Edge WebView2 de Microsoft para la vista Explorador, puede seleccionar esta opción en la sección Vista[1597] del cuadro de diálogo Opciones[1583].

Nota: Microsoft Edge WebView2 utiliza el proyecto de software Chromium, en el que se basa el explorador Chrome de Google, por lo que si usa WebView2 para la vista Explorador también podrá acceder a una buena vista previa del aspecto en Chrome de una página web.

Nota sobre Microsoft Internet Explorer y XSLT

Para poder usar la vista Explorador necesita tener instalados Internet Explorer de Microsoft, versión 5.0 o superior, o bien Microsoft Edge WebView2 (*véase el punto anterior*).

Tenga en cuenta estos puntos sobre Internet Explorer en la vista Explorador:

- Si quiere usar la vista Explorador para visualizar archivos XML transformados por una hoja de estilos XSLT, recomendamos que use Internet Explorer 6.0 o superior, que usa MSXML 3.0, un analizador XML completamente compatible con el estándar XSLT 1.0. Puede que también quiera instalar MSXML 4.0.
- La compatibilidad de Internet Explorer 5 no se ajusta al 100% a la recomendación XSLT oficial, por lo que si está teniendo problemas en la vista Explorador con IE5, deberá actualizarlo a la versión 6 o superior.
- En general es recomendable que compruebe siempre la compatibilidad con XSLT de su versión de Internet Explorer.
- Si tiene problemas con la visualización de HTML en Internet Explorer, incluya esta etiqueta **`meta`** en el elemento **`head`** de su documento HTML:

```
<head>
... <meta http-equiv="X-UA-Compatible" content="ie=edge">...
</head>
```

Herramientas de desarrollo en la vista Explorador

Puede usar las herramientas de desarrollo del explorador Edge para inspeccionar, depurar y hacer pruebas con su código HTML. Para abrir esas herramientas haga clic con el botón derecho en el panel vista Explorador y seleccione **Abrir herramientas de desarrollo**.

Funciones de la vista Explorador

Estas son las funciones disponibles en la vista **Explorador**. Para tener acceso a estas funciones, utilice los menús **Explorador**, **Archivo** y **Edición**.

- **Abrir en una ventana separada**: cuando la vista **Explorador** se abre en una ventana separada, se puede colocar en paralelo a una de las vistas de edición del mismo documento. Este comando está en el menú **Explorador | Ventana separada**. Con él puede usar la vista Explorador en dos ventanas: (i) una ventana separada y (ii) una pestaña en la ventana principal. También puede acceder a este comando desde el menú desplegable del botón **Vista Explorador** (en la parte inferior de la ventana principal). En la sección 1597 *Vista* 1597 del cuadro de diálogo "Opciones", puede especificar si la vista **Explorador** se debe abrir por defecto en una ventana separada o no.
- **Adelante** y **Atrás**: los comandos típicos de un explorador para navegar por las páginas que se cargaron en la vista **Explorador**. Estos comandos están en el menú **Explorador**.
- Tamaño de la fuente: se puede ajustar con el comando **Tamaño de fuente** del menú **Explorador**.
- **Detener**, **Actualizar** e **Imprimir**: más comandos típicos de un explorador. Estos están en los menús **Explorador** y **Archivo**.
- **Buscar**: permite buscar cadenas de texto. Este comando está en el menú **Edición**.

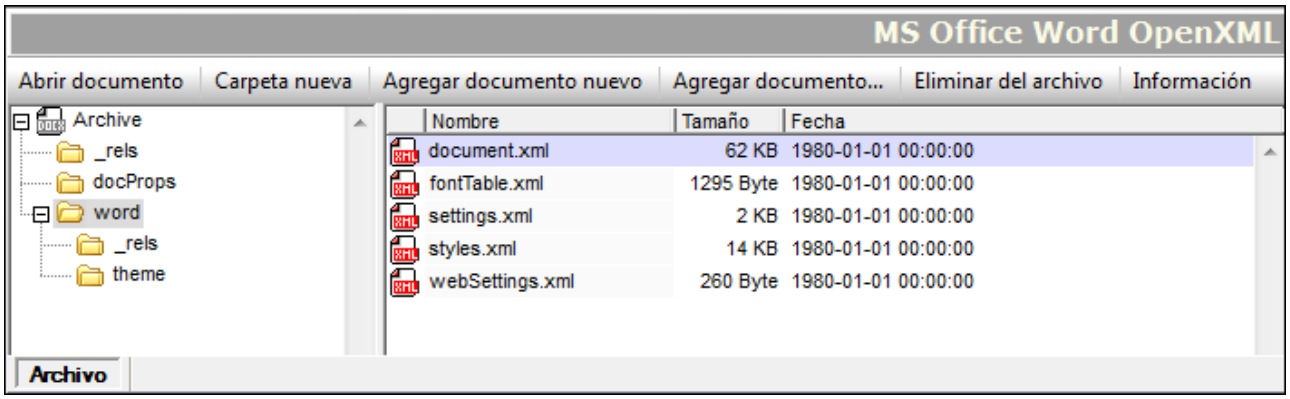
© 2016-2022 Altova GmbH

4.9 Vista Archivo

En la vista Archivo puede ver y editar archivos de Office Open XML (OOXML)[938], archivos ZIP[943] (WinZip o WinRAR, por ejemplo) o archivos EPUB[945]. En la vista Archivo no sólo puede modificar la estructura de los archivos OOXML, ZIP y EPUB, sino que también puede abrir los distintos ficheros del archivo desde la vista Archivo, editarlos en una de las vistas de edición de XMLSpy y guardarlos directamente en el archivo.

Archivos de almacenamiento en la vista Archivo

Cuando un archivo de almacenamiento (un archivo OOXML, ZIP o EPUB) se crea o se abre en XMLSpy[940], el archivo se abre en la vista Archivo (*imagen siguiente*). Puede abrir varios archivos de forma simultánea y cada archivo se abre en una ventana distinta de la vista Archivo. El tipo de archivo de almacenamiento aparece en la parte superior derecha de la vista Archivo. En la imagen siguiente, el tipo de archivo de almacenamiento es MS Office Word Open XML.



Vista de carpetas

La vista de carpetas está en el lado izquierdo de la ventana de la vista Archivo y muestra la estructura de carpetas del archivo comprimido. En cada nivel, las carpetas aparecen en orden alfabético. Para ver las subcarpetas de una carpeta, haga clic en el signo + a la izquierda de la carpeta. Si una carpeta no tiene un signo + es porque no tiene subcarpetas. Para ver los ficheros de documento (en adelante, documentos) que contiene una carpeta, seleccione la carpeta y los documentos aparecerán en la ventana principal. En la imagen anterior, los documentos que aparecen en la ventana principal están en la carpeta `word`, que tiene dos subcarpetas: `_rels` y `theme`.

Ventana principal

La ventana principal muestra los documentos de la carpeta que está seleccionada en la vista de carpetas. Los documentos aparecen en orden alfabético y por cada documento se indica el tamaño del documento descomprimido y la fecha y hora de la última modificación. Para abrir un documento desde la vista Archivo, haga doble clic en el documento. El documento se abre en una ventana distinta de XMLSpy.

Botones de comando

Los botones de comando están en la parte superior de la ventana de la vista Archivo.

- ***Abrir documento:*** se habilita cuando se selecciona un documento en la ventana principal. Al hacer clic en el botón se abre el documento seleccionado. También puede abrir un documento haciendo doble clic en el documento en la ventana principal.
- ***Carpeta nueva:*** añade una carpeta nueva a la carpeta que está seleccionada en la vista de carpetas. Debe nombrar la carpeta inmediatamente después de crearla en la vista de carpetas porque una vez creada no se puede cambiar el nombre de la carpeta. La carpeta nueva se guarda en el archivo de almacenamiento cuando se guarda el archivo propiamente dicho.
- ***Agregar documento nuevo:*** añade un documento nuevo a la carpeta que está seleccionada en la vista de carpetas. Al hacer clic en este botón se abre el cuadro de diálogo "Crear documento nuevo" de XMLSpy. El documento nuevo se abre en una ventana distinta de XMLSpy. Debe nombrar el documento inmediatamente después de crearlo porque una vez creado no se puede cambiar el nombre del documento. El documento se guarda en el archivo de almacenamiento sólo si guarda el documento en su ventana de edición o si guarda el archivo de almacenamiento.
- ***Agregar documento...:*** abre el cuadro de diálogo Examinar para que pueda buscar el documento que desea agregar. El documento se añade a la carpeta que esté seleccionada, aparece en la lista de documentos de la ventana principal y se abre automáticamente en una ventana distinta de XMLSpy. Para guardar el documento en el archivo de almacenamiento, guarde el documento en su vista de edición o guarde el archivo de almacenamiento.
- ***Eliminar del archivo:*** elimina el documento seleccionado (en la ventana principal) o la carpeta seleccionada (en la vista de carpetas) del archivo de almacenamiento. Para que surta efecto la eliminación es necesario guardar el archivo de almacenamiento.
- ***Información:*** activa o desactiva la ventana de información (ver el subapartado siguiente).

Ventana de información

La ventana de información se activa y desactiva con solo hacer clic en el botón de comando Información. La ventana de información ofrece información general sobre el archivo de almacenamiento: el número de ficheros que contiene, su tamaño comprimido y descomprimido y el grado de compresión.

© 2016-2022 Altova GmbH

4.10 Teclas de acceso rápido

La tabla siguiente presenta las teclas de acceso rápido predeterminadas de los comandos de edición usados con más frecuencia. Puede personalizarlas en la pestaña Teclado del cuadro de diálogo Personalizar[1569].

Teclas de función (también para validación y transformación)

F1	Menú Ayuda
F1 + Alt	Abrir último archivo
F3	Buscar siguiente
F4 + CTRL	Cerrar ventana activa
F4 + Alt	Cerrar XMLSpy
F5	Actualizar
F6 + CTRL	Recorrer ventanas abiertas
F7	Comprobar formato XML
F8	Validar
F10	Transformación XSL
F10 + CTRL	Transformación XSL:FO

Comandos de archivo y de aplicación

Alt + F1	Abrir último archivo
CTRL + O	Abrir archivo
CTRL + N	Archivo nuevo
CTRL + P	Imprimir archivo
CTRL + S	Guardar archivo
CTRL + F4	Cerrar ventana activa
CTRL + F6	Recorrer ventanas abiertas
CTRL + TAB	Cambiar de documento abierto
Alt + F4	Cerrar XMLSpy

Otras teclas

Teclas de flecha arriba/abajo	Mover el cursor o la barra de selección
Esc	Abandonar cambios o cerrar cuadro de diálogo
Entrar	Confirmar selección
Supr	Eliminar carácter o elemento seleccionado
Mayús + Supr	Cortar

Comandos de edición

CTRL + A	Seleccionar todo

CTRL + F	Buscar
CTRL + G	Ir a la línea o carácter
CTRL + H	Reemplazar
CTRL + V	Pegar
CTRL + X	Cortar
CTRL + Y	Rehacer
CTRL + Z	Deshacer

© 2016-2022 Altova GmbH

5 XML

Esta sección describe cómo se trabaja con documentos XML en XMLSpy. Abarca estos aspectos:

- Cómo crear, abrir y guardar documentos XML 286. En esta sección se explican también importantes opciones de configuración de XMLSpy relacionadas con la creación de archivos.
- Los documentos XML se pueden editar en las vistas Texto 290, Cuadrícula 293 y Authentic 331. Puede seleccionar la vista que más le convenga en cada momento y cambiar de una vista a otra mientras edita el documento. Cada vista tiene sus ventajas.
- Puede añadir fragmentos XML 339 rápidamente a un documento XML desde fuentes externas.
- Cómo validar XML 335 de XMLSpy.
- Los ayudantes de entrada 333 para documentos XML tienen funciones específicas, descritas en detalle en esta sección.
- Cómo procesar documentos XML con XSLT y XQuery 341. Se explican también varias funciones de XMLSpy relacionadas con el procesamiento de documentos XML. Esta sección también incluye un apartado sobre fuentes PDF 343 que explica cómo se procesan las fuentes al generar documentos de salida en PDF.
- También se describen otras funciones 423 varias relacionadas con los documentos XML.

modelar con UModel Editor XML

5.1 Crear, abrir y guardar documentos XML

Al crear, abrir y guardar documentos XML, es necesario hacerse estas preguntas:

- ¿En qué vista se abrirá el documento XML: en la vista Texto, Cuadrícula o Authentic?
- Cuando se crea un documento XML nuevo, ¿se le asignará un esquema (esquema XML o DTD) automáticamente, manualmente o no se asignará?
- Si se asigna un esquema al documento XML, ¿se validará el documento automáticamente cada vez que se abra o se guarde?

Vista predeterminada

Existen opciones generales de configuración para especificar en qué vista se deben abrir los documentos XML (tanto documentos recién creados como documentos ya existentes). Estas opciones de configuración están en el cuadro de diálogo "Opciones" (**Herramientas | Opciones**).

En la sección *Tipos de archivo* del diálogo "Opciones", seleccione el tipo de archivo `.xml` y en el panel "Vista predeterminada" marque la vista de edición que desea utilizar (Texto o Cuadrícula) para este tipo de archivo. Recuerde que: (i) la vista Esquema y la vista WSDL se puede utilizar sólo para archivos de esquema XML y documentos WSDL respectivamente y que (ii) la vista Explorador es una vista de visualización, no de edición.

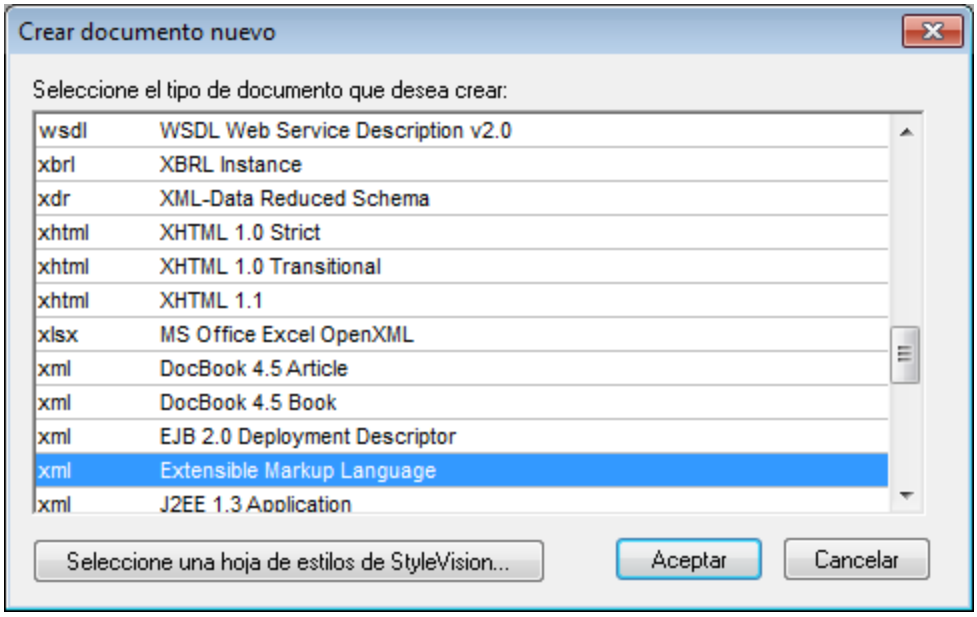
En la sección *Tipos de archivo* también puede definir XMLSpy como editor predeterminado para el tipo de archivo seleccionado.

Los documentos XML se pueden editar en la vista Authentic si al documento se le asignó una hoja de estilos StyleVision Power Stylesheet (SPS). Puede especificar que el archivo XML se abra directamente en la vista Authentic siempre que el archivo XML tenga asociada una SPS. Para ello, en la sección *Vista* del cuadro de diálogo "Opciones" marque la casilla *Cuando se haya asignado una hoja de estilos de StyleVision, abrir los archivos en la vista Authentic*. Si no se marca esta opción, el archivo se abrirá en la vista predeterminada que se especificara para los archivos `.xml` en la sección *Tipos de archivo* (ver párrafos anteriores).

Asignar esquemas

Cuando necesite crear un archivo XML nuevo, seleccione el comando de menú **Archivo | Nuevo**. A continuación se abre el cuadro de diálogo "Crear documento nuevo" (*imagen siguiente*).

© 2016-2022 Altova GmbH



Observe que hay varios tipos de documento XML disponibles en el cuadro de diálogo. La opción *Extensible Markup Language* crea un documento XML genérico. El resto de opciones están asociadas a un esquema, por ejemplo a la DTD DocBook. Si selecciona una de estas opciones, se crea un documento XML que (i) tiene asociado automáticamente el esquema correspondiente y (ii) tiene una estructura base válida de acuerdo con el esquema asignado. Recuerde que puede crear su propio documento XML base. Si lo guarda en la carpeta `Template` de la carpeta de la aplicación, el documento base estará disponible en el cuadro de diálogo "Crear documento nuevo".

Si selecciona el tipo de documento XML genérico (*Extensible Markup Language*), se le pedirá que asigne un esquema (DTD o esquema XML) al documento. Puede elegir ahora el esquema o continuar y crear un documento XML que no tenga asignado ningún esquema.

El esquema se puede asignar más tarde al documento (menú **DTD/Esquema**).

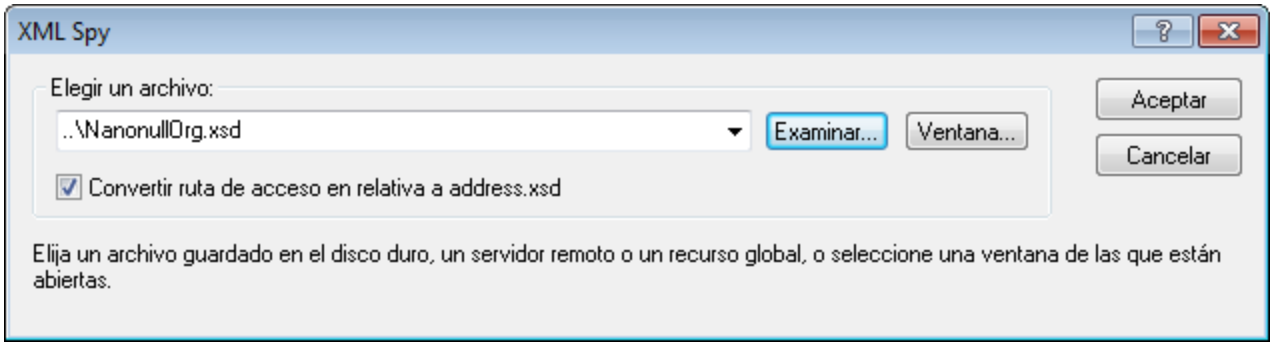
Validación automática

Si un documento XML tiene asignado un esquema, se puede validar automáticamente cada vez que se abra o cierre el documento. Esta opción de configuración está en la sección *Archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**). Las opciones de configuración de validación automática de la sección *Archivo* se pueden combinar con una opción de la sección *Tipos de archivo*, lo cual permite deshabilitar la validación automática para ciertos tipos de archivo. Gracias a las opciones de ambas pestañas puede configurar la validación automática para diferentes tipos de archivo.

5.2 Asignación de esquemas y validación

Sitio web de Altova: Validador XML, Validación XML

Cuando se crea un documento XML por primera vez 286 se le puede asignar un esquema (DTD o esquema XML). También se puede asignar el esquema o cambiar el esquema asignado con los comandos **Asignar DTD** o **Asignar esquema** del menú **DTD/Esquema**.



Para convertir en relativa la ruta de acceso al archivo de esquema que se inserta en el documento XML, marque la casilla correspondiente en el cuadro de diálogo.

Recursos globales para esquemas

Un recurso global es un alias para un archivo o una carpeta. El archivo o la carpeta de destino se puede cambiar dentro de la IGU con solo cambiar la configuración activa del recurso global (comando de menú **Herramientas | Configuración activa**). Por tanto, los recursos globales permiten cambiar el esquema asignado por otros esquemas, lo cual puede ser muy práctico a la hora de realizar pruebas. Para más información sobre cómo usar los recursos globales, consulte la sección Recursos globales de Altova 1042.

Esquema XML más DTD

Una característica muy práctica de DTD que no tiene XML Schema es el uso de entidades. Sin embargo, si desea usar entidades en un documento XML que se valida con un esquema XML, puede añadir una declaración `DOCTYPE` al documento XML e incluir en ella las declaraciones de entidad.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrgChart [
  <!ENTITY name-int "value">
  <!ENTITY name-ext SYSTEM "extfile.xml">
]>
<OrgChart xmlns="http://www.xs.com/org"
          xsi:schemaLocation="http://www.xs.com/org OrgChart.xsd">
  ...
</OrgChart>
```

Una vez declaradas las entidades, podrá usarlas en el documento XML. El documento tendrá un formato XML correcto y será válido. Recuerde, no obstante, que las entidades externas analizadas no son compatibles con la vista Authentic.

© 2016-2022 Altova GmbH

Ir a definiciones de esquema

Mientras está abierto el documento XML, puede abrir la DTD o el esquema XML asignados al documento con solo hacer clic en los comandos **Ir a la DTD** o **Ir al esquema** del menú **DTD/Esquema**. Además, puede situar el cursor dentro de un nodo del documento XML y saltar a la definición de esquema de dicho nodo con solo hacer clic en el comando **Ir a la definición** del menú **DTD/Esquema**.

Validar y comprobar el formato XML

Para validar y comprobar el formato XML del documento, utilice los comandos **Validar XML (F8)** y **Comprobar formato XML (F7)** del menú XML o sus respectivos iconos de la barra de herramientas. Los errores de validación y de formato XML se comunican en la ventana Mensajes. Si un documento XML resulta no ser válido, el validador ofrece opciones de corrección rápida e inteligente para reparar el error[1337] basadas en la información disponible en el esquema.

También puede usar un servidor RaptorXML Server[1068] para validar documentos XML.

5.3 Datos XML en la vista Texto

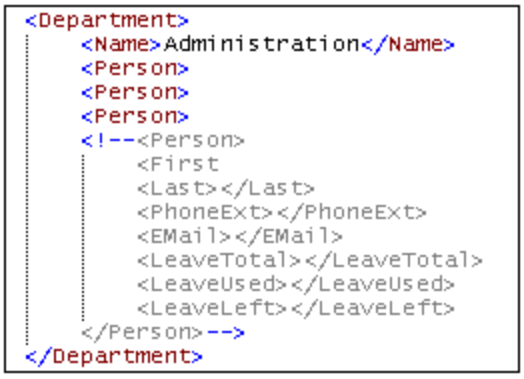
XMLSpy ofrece algunas funciones de edición de texto XML especializadas, además de las funciones de edición que suelen estar disponibles en la vista Texto (consulte el apartado Vista Texto 145 de la sección Vistas de edición).

- Insertar o quitar comentarios 290
- Escapar caracteres XML y anular escape de caracteres XML 290
- Insertar rutas de acceso de archivo 291
- Insertar fragmentos XML mediante XInclude 291
- Copiar expresiones XPath y XPointer en el portapapeles 291
- Guardar una cadena cifrada en base 64 como imagen 291

Insertar o quitar comentarios

El texto del documento XML se puede convertir en comentario si se usan los delimitadores de inicio y fin de comentario XML (`<!--` y `-->` respectivamente). En XMLSpy estos delimitadores de comentario se pueden insertar con el comando de menú **Edición | Insertar o quitar comentario**.

Puede acceder a este comando desde el menú **Edición** o desde el menú contextual que aparece al hacer clic con el botón derecho en un fragmento de texto. El texto del comentario aparece en color gris (*ver imagen siguiente*).



Para eliminar el comentario y recuperar el bloque de texto que formaba parte del documento, seleccione el bloque de texto **excepto** los delimitadores de comentario y seleccione el comando **Insertar o quitar comentario** del menú **Edición** o del menú contextual que aparece al hacer clic con el botón derecho en el texto. Los delimitadores de comentario se eliminan y el texto deja de estar en gris.

Nota sobre líneas vacías

En los documento XML, las líneas vacías se desechan al cambiar de vista o al guardar el documento. Si desea conservar las líneas vacías, inclúyalas entre delimitadores de comentario.

Escapar caracteres XML y anular el escape de caracteres XML

A los cinco caracteres XML especiales (*ver la lista que aparece a continuación*) se les puede añadir caracteres de escape (*ver lista*). Para ello, resalte el bloque de texto al que quiere aplicar caracteres de escape y

© 2016-2022 Altova GmbH

seleccione el comando de menú **Escapar caracteres XML** o **Anular escape de caracteres XML**. Los caracteres XML especiales del bloque de texto seleccionado se modificarán según el comando seleccionado.

<	<
>	>
&	&
'	'
"	"

Por ejemplo:

Si seleccionamos el texto `<a></a>` y hacemos clic en el comando **Escapar caracteres XML**, el texto seleccionado se sustituye por `<a></a>`
Si seleccionamos el texto `<a>` y `</a>` y hacemos clic en el comando **Anular escape de caracteres XML**, el texto seleccionado se sustituye por `<a></a>`

Insertar rutas de acceso de archivos

El comando **Edición | Insertar ruta de acceso del archivo**[1284] permite buscar el archivo en cuestión e insertar su ruta de acceso en la posición seleccionada del documento XML que se está editando. Este comando permite escribir rutas de acceso de forma rápida y precisa. Para más información, consulte la descripción del comando[1284].

Insertar fragmentos XML mediante XInclude

El comando **Edición | Insertar XInclude**[1284] permite insertar mediante XInclude el contenido de un documento XML entero o parte de un documento XML en el documento XML que se está editando. Con este comando puede introducir con rapidez y precisión documentos XML enteros (mediante el mecanismo XInclude) o fragmentos de documentos XML (mediante una extensión XPointer del mecanismo XInclude). Para más información, consulte la descripción del comando[1284].

Copiar expresiones XPath y XPointer en el portapapeles

Las expresiones XPath y XPointer del nodo seleccionado (que expresan la posición del documento XML donde se encuentra el nodo) se pueden copiar en el portapapeles con los comandos **Edición | Copiar XPath**[1283] y **Edición | Copiar XPointer**[1283] respectivamente. Así podrá obtener las expresiones XPath y XPointer correctas que apuntan al nodo seleccionado.

Imaginemos, por ejemplo, que el nodo seleccionado en la vista Texto o Cuadrícula es el tercer elemento `Office` de un elemento de documento llamado `Offices`. En este caso, la expresión XPath que se copiaría sería `/Offices/Ofice[3]`. Si los elementos `Office` no tienen más elementos con nombre del mismo nivel antes del tercer elemento `Office`, la expresión XPointer que se copiaría en este caso sería `element(/1/3)`.

Después de copiarlas, puede insertar las expresiones donde desee. Por ejemplo, puede insertar la expresión XPath en una hoja de estilos XSLT y la expresión XPointer en el atributo `href` de un elemento `xinclude`.

Para consultar una descripción más detallada de estos comandos, le remitimos a la *Comandos de menú*.

Guardar una cadena cifrada en base 64 como imagen

Para guardar una cadena cifrada en base 64 en formato imagen haga clic con el botón derecho del ratón en el

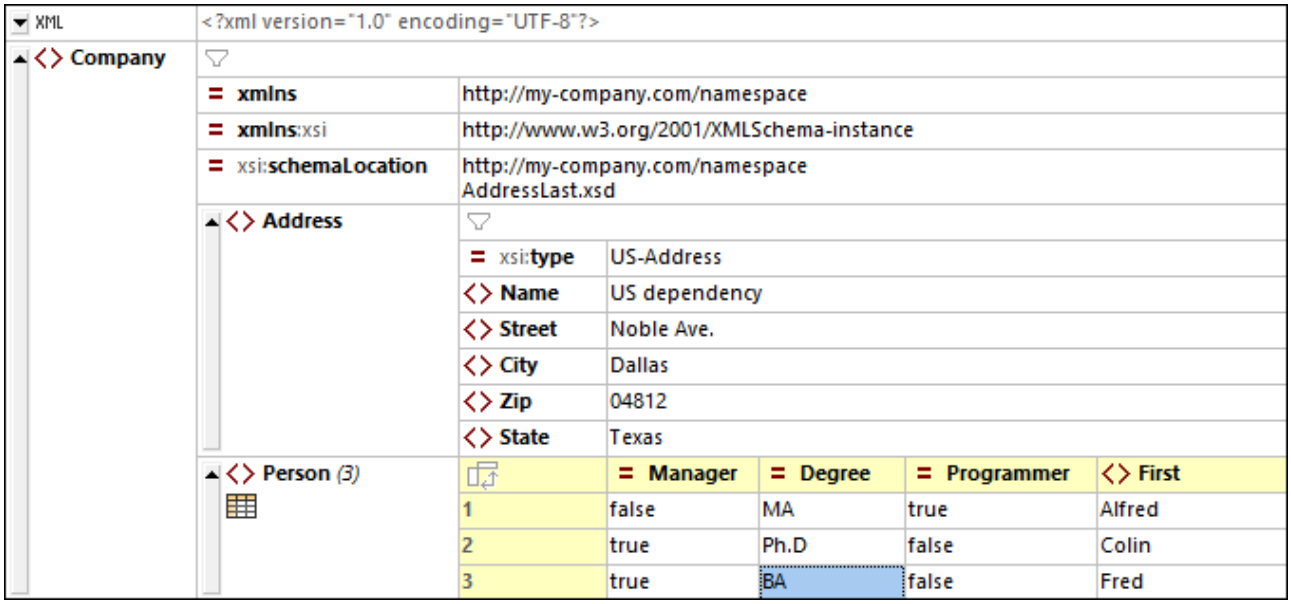
tecto cifrado y seleccione el comando **Guardar como imagen**. En el cuadro de diálogo que aparece, seleccione la ubicación donde quiere guardar la imagen y asigne un nombre al archivo de imagen. La extensión del archivo (`.png`, `.gif`, `.svg`, etc.) se detecta automáticamente en el cifrado en base 64 y aparece en el cuadro de diálogo **Guardar**. Haga clic en Guardar para terminar.

También puede ejecutar esta acción con el comando de menú **Edición | Guardar como imagen**.

© 2016-2022 Altova GmbH

5.4 Datos XML en la vista Cuadrícula

La vista Cuadrícula[169] muestra la estructura jerárquica de los **documentos XML** mediante un conjunto de contenedores anidados que se pueden expandir o contraer. De este modo se presenta una imagen clara de la estructura del documento. En la vista Cuadrícula se puede modificar la estructura del documento y editar su contenido fácilmente.



En la imagen anterior se puede observar que el documento se presenta en forma de cuadrícula y de manera jerárquica. Si un nodo contiene datos puede ser directamente contenido (como en el caso de los nodos de *Texto*) o puede estar dividido en dos campos: nombre del nodo y contenido del nodo (como en el caso de los nodos *Elemento*). Los nombres de los nodos aparecen en negrita y el contenido en la fuente normal.

Barra de herramientas de la vista Cuadrícula

Puede acceder a los comandos relacionados con la edición en la vista Cuadrícula desde la barra de herramientas de la vista Cuadrícula (*imagen siguiente*), que se encuentra en el área de las barras de herramientas, en la parte superior de la ventana de la aplicación. Pase el cursor del ratón sobre los iconos para ver su nombre y acceso rápido.



Nota: estos comandos también están disponibles en el menú XML[1332].

Resumen de las características de la vista Cuadrícula

Para ver las características de la vista Cuadrícula consulte estos apartados:

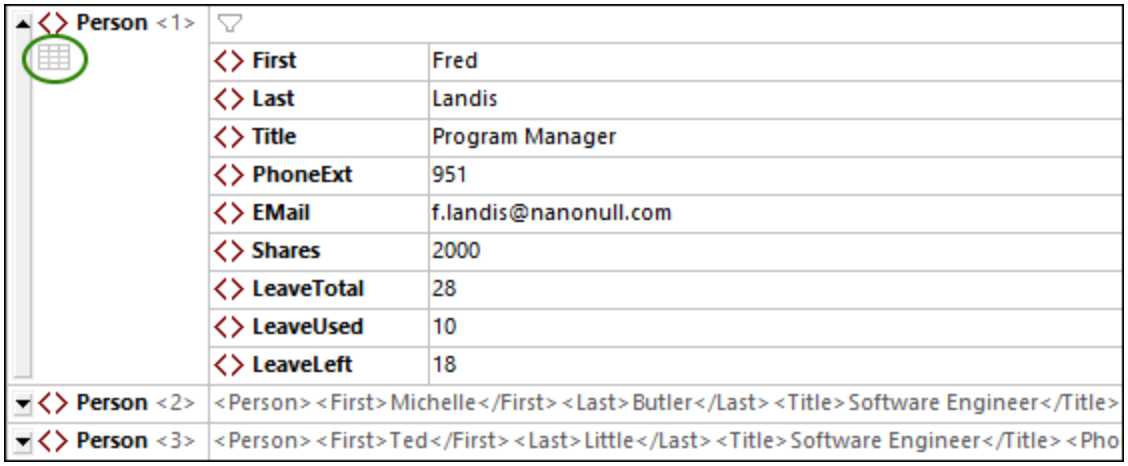
- Visualización de documentos[295], donde se explican las características de visualización de la vista Cuadrícula

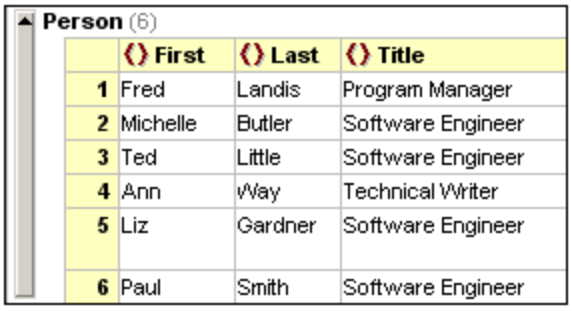
- Estructura de un documento[299], donde se explica cómo aparece la estructura de un documento en la vista Cuadrícula y qué características puede usar para modificar esa estructura
- Contenido de un documento[302], donde se explica cómo editar y validar contenido en la vista Cuadrícula
- Configurar la vista Cuadrícula[326], donde se explican las opciones que permiten configurar la vista Cuadrícula

Mostrar como tabla

Los elementos extensibles aparece en una Vista Cuadrícula *estándar* uno tras otro, progresando verticalmente hacia abajo en el orden de los documentos (*imagen siguiente*). Sin embargo, al mostrar los elementos extensibles como *filas de una tabla* se habilitan más opciones de edición. En la imagen siguiente, se repite el elemento **Person**. La imagen izquierda muestra la vista Cuadrícula estándar, donde el primer elemento **Person** aparece ampliado, mientras que las instancias siguientes aparecen contraídas. En la imagen de la derecha aparecen los elementos extensibles **Person** como filas de una tabla.

Para cambiar a la Vista Tabla haga clic en el icono **Modo tabla** (*dentro de un círculo verde en la imagen izquierda*). Al cambiar a la vista de tabla el icono aparece en color (*imagen siguiente derecha*).





La vista Tabla tiene una ventaja a la hora de editar los documentos: puede manipular columnas y filas enteras en relación con otras columnas y filas de la tabla. Esto permite operaciones como ordenar las filas de tabla según los valores de una columna. Por ejemplo, en la imagen anterior derecha los seis elementos **Person** se pueden ordenar en base a sus elementos secundarios **Last** con un simple comando en la barra de herramientas de la vista Cuadrícula[293]. Esta operación es más sencilla que ejecutar una transformación XSLT, que sería la forma habitual de ordenar un conjunto de nodos XML.

Para más información consulte Visualizar tablas[309].

© 2016-2022 Altova GmbH

Filtros y fórmulas

La vista Cuadrícula ofrece dos características únicas y muy útiles:

- Los Filtros 314 permiten filtrar los datos que aparecen en la vista Cuadrícula. Un filtro se puede aplicar a uno o más elementos del documento.
- Las fórmulas 316 permiten calcular un resultado o generar un conjunto de nodos que se puede guardar en el documento.

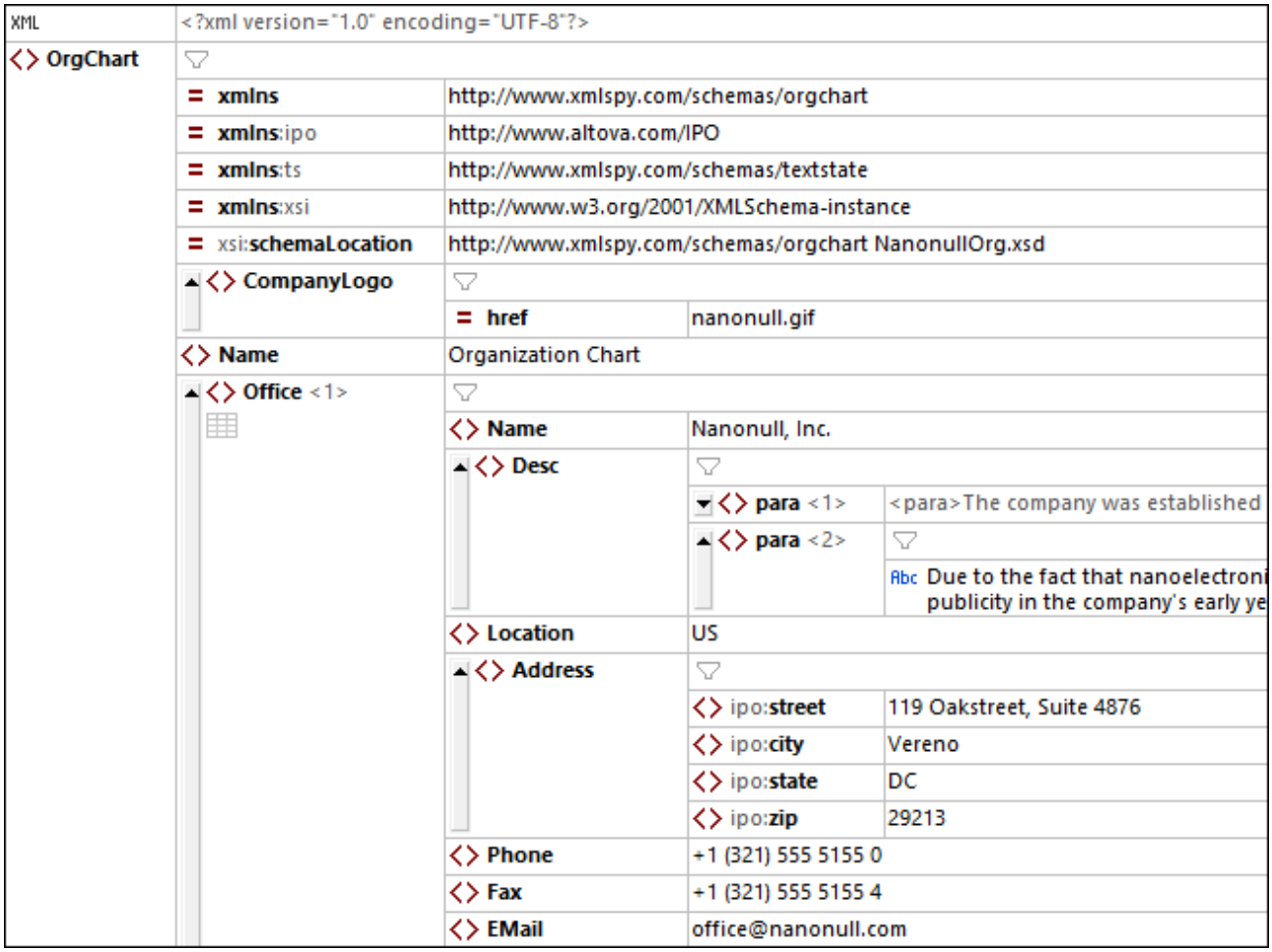
Para más información consulte los apartados correspondientes.

Imágenes y gráficos en la vista Cuadrícula

Las imágenes y los gráficos se pueden ver como gráficos en la vista Cuadrícula. Para más información consulte Ver imágenes en la vista Cuadrícula 319 y Gráficos en la vista Cuadrícula 320

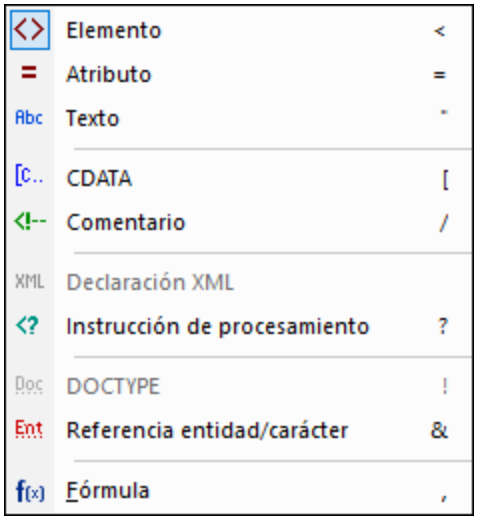
5.4.1 Vista documentos

La vista Cuadrícula presenta los documentos XML de forma jerárquica dentro de una cuadrícula (*imagen siguiente*).



Cada línea de la cuadrícula contiene un componente de la estructura (al que se conoce como nodo), como un elemento, atributo, comentario o texto. En la imagen siguiente puede ver los distintos tipos de nodos que existen en los documentos XML y sus correspondientes iconos. En la vista Cuadrícula el tipo de nodo viene indicado por el icono a la izquierda de la celda de cuadrícula en cuestión. Recuerde que los nodos de tipo *elemento*, *atributo* e *instrucción de procesamiento* tienen una clave y un valor, mientras que los nodos de tipo *texto*, *CDATA* y *comentario* solo tienen un valor. Por ejemplo, un nodo de elemento tendrá un nombre (su clave) y un contenido (su valor), mientras que un nodo de texto solo tendrá un valor.

© 2016-2022 Altova GmbH



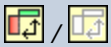
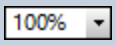
Características de la vista Cuadrícula

Estas son algunas de las características de esta vista:

- Puede acercar y alejar el contenido con la tecla **Ctrl y la rueda del ratón** o con **Ctrl+[más]** y **Ctrl+[menos]**.
- Los elementos del mismo nivel que se llamen igual se numeran empezando por el 1. En la imagen anterior, por ejemplo, el elemento `Office` con el número 1 es el primero de una secuencia de elementos del mismo nombre. Los demás elementos de la imagen no están numerados porque no hay elementos del mismo nivel con el mismo nombre.
- Los nodos se pueden expandir o contraer; para ello haga clic en la cabeza de flecha que verá a la izquierda del símbolo del nodo (*imagen anterior*). Cuando se contrae un nodo su contenido aparece en una sola línea.
- Si selecciona varios elementos en un mismo nivel puede expandirlos o contraerlos todos a la vez pulsando la tecla **Mayús** y haciendo clic en las cabezas de flecha correspondientes.
- Si el ajuste automático de línea está activado en la barra de herramientas de la vista Cuadrícula[293], todas las celdas que contengan un texto que sobrepase el ancho de la celda aplicarán el ajuste automático de línea. Para desactivas el ajuste automático de línea vuelva a hacer clic en el icono correspondiente de la barra de herramientas de la vista Cuadrícula[293].
- Las Fórmulas[696] son específicas de XMLSpy.

Botones de comandos de la barra de herramientas y dentro de las celdas

En la vista Cuadrícula XML los botones que hay dentro de las celdas o en la barra de herramientas de la vista Cuadrícula[293] permiten realizar acciones en el contexto de la selección actual (celda o columna). Es más rápido usar estos botones de comando para las acciones que se usan más a menudo. En la tabla siguiente describimos los comandos de estos botones. Los comandos relacionados con la visualización tienen fondo azul. Los comandos para editar el documento (estructura y contenido) tienen fondo beige. Si necesita otros comandos en el contexto de alguna de las celdas puede usar el menú contextual[706] de la celda. Nota: si pasa el cursor del ratón por encima de un botón aparece información rápida con el nombre del comando del botón en cuestión.

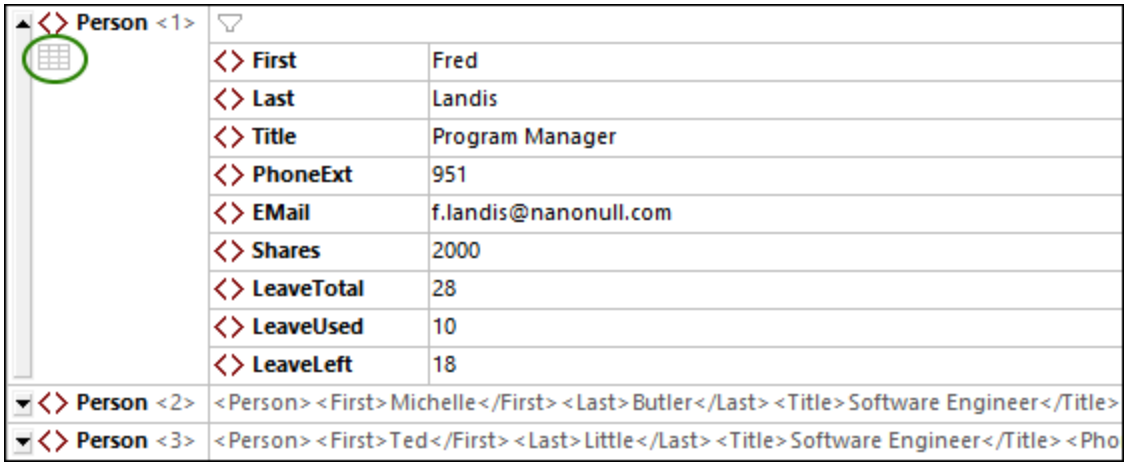
	Optimiza el ancho de las columnas para que se ajuste a su contenido.
	Si aparece en color, la Vista Diseño de tabla 692 está activada, si está desactivada aparece en gris. Haga clic en el icono para cambiar la vista
	Este comando cambia las filas por las columnas y viceversa
	Si está naranja significa que hay un filtro 696 para la tabla, el objeto o la matriz en cuestión. Si el icono está blanco, el filtro está desactivado. Para activar/desactivar el filtro, haga clic en el icono. Para editar la expresión, haga doble clic en el icono.
	Ajuste automático de línea que afecta a todas las celdas
	Nivel de acercamiento de la vista Cuadrícula
	Se habilita cuando se selecciona un encabezado de columna en la vista Tabla. Los botones ordenan las filas de la tabla, respectivamente, en orden alfabético ascendente o descendente.

Nota: para ver una descripción de otros comandos de la barra de herramientas consulte el apartado Menú contextual de la vista Cuadrícula 323.

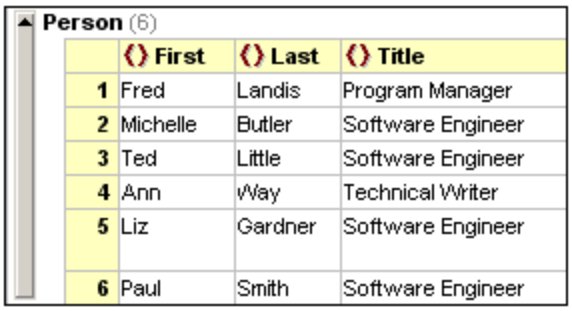
Mostrar como tabla

Los elementos extensibles aparece en una Vista Cuadrícula *estándar* uno tras otro, progresando verticalmente hacia abajo en el orden de los documentos (*imagen siguiente*). Sin embargo, al mostrar los elementos extensibles como *filas de una tabla* se habilitan más opciones de edición. En la imagen siguiente, se repite el elemento **`Person`**. La imagen izquierda muestra la vista Cuadrícula estándar, donde el primer elemento **`Person`** aparece ampliado, mientras que las instancias siguientes aparecen contraídas. En la imagen de la derecha aparecen los elementos extensibles **`Person`** como filas de una tabla.

Para cambiar a la Vista Tabla haga clic en el icono **Modo tabla** (*dentro de un círculo verde en la imagen izquierda*). Al cambiar a la vista de tabla el icono aparece en color (*imagen siguiente derecha*).



 © 2016-2022 Altova GmbH



Person (6)

	First	Last	Title
1	Fred	Landis	Program Manager
2	Michelle	Butler	Software Engineer
3	Ted	Little	Software Engineer
4	Ann	Way	Technical Writer
5	Liz	Gardner	Software Engineer
6	Paul	Smith	Software Engineer

La vista Tabla tiene una ventaja a la hora de editar los documentos: puede manipular columnas y filas enteras en relación con otras columnas y filas de la tabla. Esto permite operaciones como ordenar las filas de tabla según los valores de una columna. Por ejemplo, en la imagen anterior derecha los seis elementos **`Person`** se pueden ordenar en base a sus elementos secundarios **`Last`** con un simple comando en la barra de herramientas de la vista Cuadrícula[293]. Esta operación es más sencilla que ejecutar una transformación XSLT, que sería la forma habitual de ordenar un conjunto de nodos XML.

Para más información consulte Visualizar tablas[309].

5.4.2 Estructura del documento

En la vista Cuadrícula puede editar la estructura del documento XML de manera gráfica. Por ejemplo, puede contraer y expandir segmentos diferentes de la estructura del documento, insertar, anexar o eliminar nodos, arrastrar nodos hasta otra posición y convertir un tipo de nodo en otro tipo de nodo.

Agregar nodos nuevos

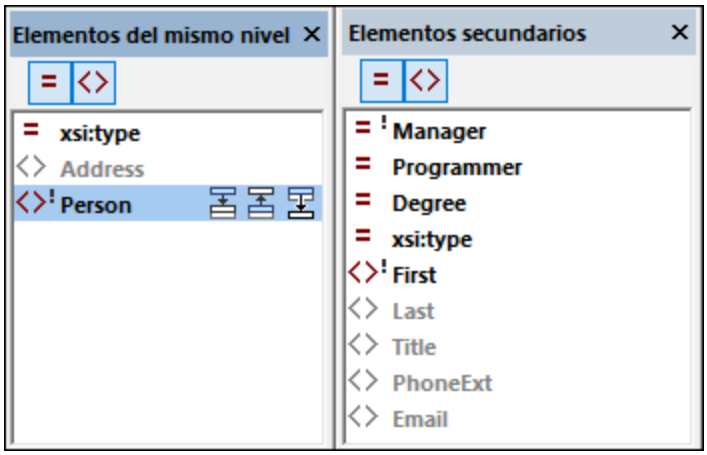
Existen dos formas de agregar nodos nuevos a un documento:

- Los ayudantes de entrada Elementos del mismo nivel y Secundarios introducen nodos en el lugar seleccionado. Para ello debe asociar un esquema (DTD o XML) al documento XML. Por ejemplo, cuando selecciona un nodo en la ventana principal, los elementos del mismo nivel y secundarios que se permiten en ese punto (según el esquema) aparecen en los ayudantes de entrada correspondientes. Puede usar los ayudantes de entrada seleccionar el nodo que quiere añadir.
- Puede añadir un nodo nuevo que no esté definido en un esquema como elemento del mismo nivel, secundario o de nivel superior. Los comandos para ello se encuentran en los menús contextuales de las celdas de la vista Cuadrícula, así como en el menú XML[1332] y en la barra de herramientas de la vista Cuadrícula[293].

Ayudantes de entrada Elementos del mismo nivel y Secundarios

Si un documento XML está asociado a un esquema (DTD o XML), entonces los elementos del mismo nivel y secundarios del nodo seleccionado aparecen en los ayudantes de entrada Elementos del mismo nivel y Secundarios. En cada uno de estos ayudantes de entrada puede activar o desactivar la visibilidad de elementos y atributos haciendo clic en los correspondientes iconos de la barra de herramientas. Los nodos obligatorios están marcados con un signo de exclamación. Los nodos que aparecen en gris no se pueden añadir. Esto puede deberse a dos motivos: (i) El nodo ya se ha añadido y no se admiten más instancias de este en el esquema, como en el caso del secundario `Address` de la imagen siguiente; (ii) tiene que añadir otro

nodo antes de poder añadir el nodo gris, como en el caso del secundario `Last` de la imagen siguiente, que solo se puede añadir una vez se haya añadido `First`.



Para añadir nodos:

- *Elementos del mismo nivel:* En el ayudante de entrada seleccione el nodo que quiere añadir como elemento del mismo nivel. Después haga clic en el icono correspondiente (*imagen anterior izquierda*) según si quiere añadir el elemento del mismo nivel antes o después del nodo seleccionado en la cuadrícula, o si lo quiere añadir como último elemento del mismo nivel del nodo seleccionado.
- *Elementos secundarios:* Haga doble clic en el nodo que quiere añadir como secundario.

Insertar nodos nuevos

Al seleccionar un nodo en el documento (es decir, una celda de la cuadrícula) puede añadir un nodo nuevo vacío como elemento del mismo nivel, secundario o principal. Estos son los comandos para llevar a cabo estas operaciones: (i) el menú contextual de la celda; (ii) el menú XML 1332 y (iii) la barra de herramientas de la vista Cuadrícula 293.

Comando	Acceso rápido
Insertar (secundario) después	Ctrl+Entrar
Insertar (secundario) antes	Ctrl+Mayús+Entrar
Anexar (secundario)	Ctr+Alt+A
Agregar secundario	Ctrl+Alt+Entrar
Envolver en elemento	Ctr+Alt+W

Tenga en cuenta que:

- El nodo nuevo se crea en un elemento vacío por defecto. Puede cambiar el tipo de nodo 302 más adelante.

© 2016-2022 Altova GmbH

- El comando **Envolver en elemento** crea un elemento de nodo alrededor del nodo. Este elemento se convierte en el nuevo elemento principal del nodo en cuestión.

Modificar una estructura con mecanismos estándar de Windows

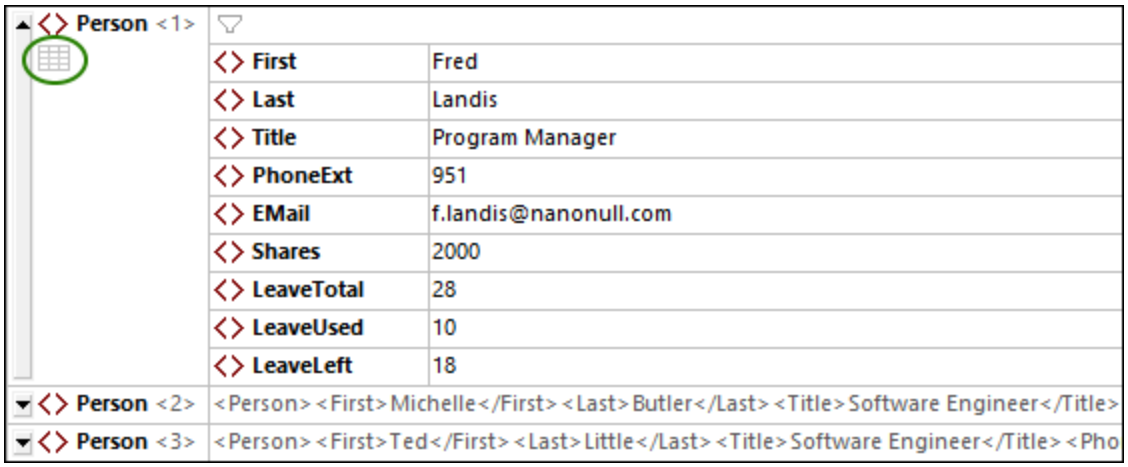
En la vista Cuadrícula también puede modificar la estructura de un documento con estos mecanismos de Windows:

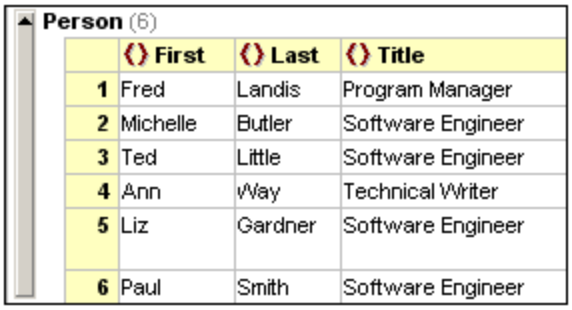
- *Eliminar:* Seleccione un componente y bórrelo con la tecla **Supr**.
- *Mover:* Seleccione un componente y arrástrelo hasta su nueva ubicación.

Mostrar como tabla

Los elementos extensibles aparece en una Vista Cuadrícula *estándar* uno tras otro, progresando verticalmente hacia abajo en el orden de los documentos (*imagen siguiente*). Sin embargo, al mostrar los elementos extensibles como *filas de una tabla* se habilitan más opciones de edición. En la imagen siguiente, se repite el elemento `Person`. La imagen izquierda muestra la vista Cuadrícula estándar, donde el primer elemento `Person` aparece ampliado, mientras que las instancias siguientes aparecen contraídas. En la imagen de la derecha aparecen los elementos extensibles `Person` como filas de una tabla.

Para cambiar a la Vista Tabla haga clic en el icono **Modo tabla** (*dentro de un círculo verde en la imagen izquierda*). Al cambiar a la vista de tabla el icono aparece en color (*imagen siguiente derecha*).





	First	Last	Title
1	Fred	Landis	Program Manager
2	Michelle	Butler	Software Engineer
3	Ted	Little	Software Engineer
4	Ann	Way	Technical Writer
5	Liz	Gardner	Software Engineer
6	Paul	Smith	Software Engineer

La vista Tabla tiene una ventaja a la hora de editar los documentos: puede manipular columnas y filas enteras en relación con otras columnas y filas de la tabla. Esto permite operaciones como ordenar las filas de tabla según los valores de una columna. Por ejemplo, en la imagen anterior derecha los seis elementos `Person` se pueden ordenar en base a sus elementos secundarios `Last` con un simple comando en la barra de

herramientas de la vista Cuadrícula[293]. Esta operación es más sencilla que ejecutar una transformación XSLT, que sería la forma habitual de ordenar un conjunto de nodos XML.

Para más información consulte Visualizar tablas[309].

5.4.3 Contenido del documento

Editar contenido XML en la vista Cuadrícula es fácil: Haga doble clic dentro del campo y edite el contenido. Puede cambiar el tipo de nodo haciendo clic en el icono Tipo del nodo y seleccionando otro tipo del menú que aparece. Para insertar entidades utilice el ayudante de entrada Entidades. La vista Cuadrícula también ofrece funcionalidades de validación y de búsqueda.

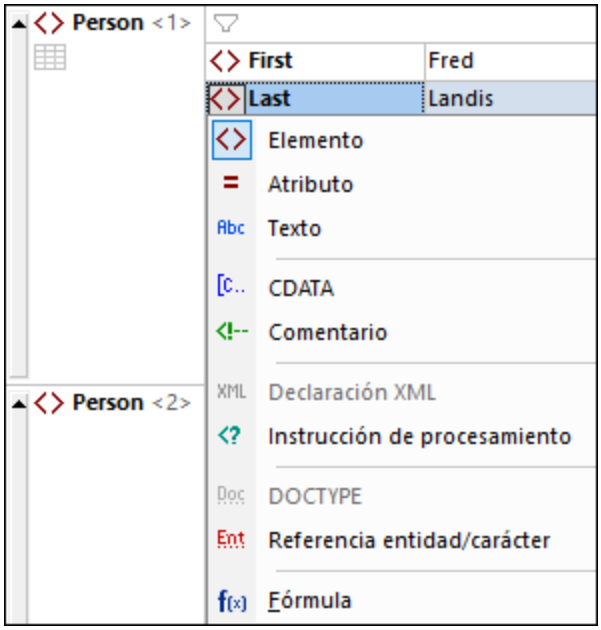
Seleccionar y convertir el tipo de nodo

Las celdas de la vista Cuadrícula contiene nodos del documento XML. Hemos agrupado los tipos para una mejor comprensión:

- *Tipos Nombre-Valor:* Elemento, atributo, instrucciones de procesamiento (PI). Los nodos de este tipo tienen nombres y valores.
- *Tipo de valor:* Texto, CDATA, comentario, referencias entidad/carácter. Los nodos de este tipo solo toman valores.
- *Tipos de definición:* Declaración XML, DOCTYPE (DTD interna o externa). Estos nodos definen las propiedades del documento XML.
- *Tipos de contenedor especiales específicos de XMLSpy:* Fórmula.

El tipo de un valor se indica con un símbolo delante del valor (*imagen siguiente*). Para cambiar un tipo haga clic en su símbolo y seleccione el tipo que quiera del menú que aparece. También puede hacer clic en una celda, en el menú contextual que aparece, seleccionar un tipo del submenú **Tipo**. En la imagen siguiente puede ver los símbolos y accesos rápidos de los tipos:

© 2016-2022 Altova GmbH



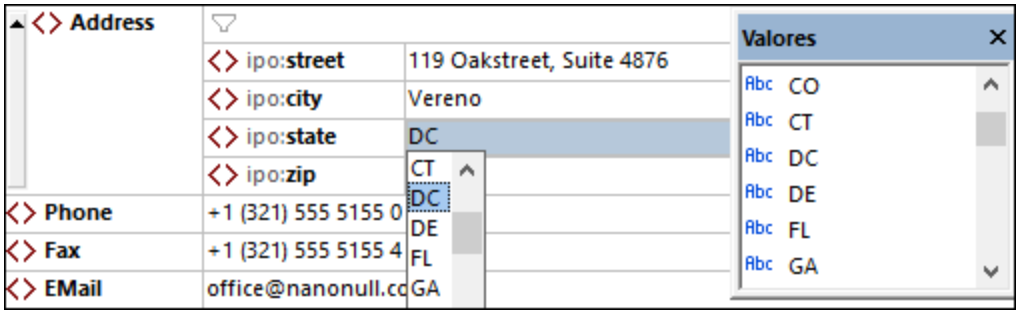
Tenga en cuenta que:

- Los nodos de tipo *elemento*, *atributo* y *PI* toman un nombre y un valor, mientras que los nodos de tipo *texto*, *CDATA* y *comentario* solo toman un valor. Por ejemplo, un nodo de elemento tendrá un nombre y un contenido, mientras que un nodo de texto solo tendrá un valor.
- En las conversiones se intentan conservar la clave y el valor originales. Por ejemplo, si convierte un elemento en un atributo, el atributo tiene (i) el mismo nombre que el elemento y (ii) un valor que sea, dentro de lo posible, el mismo que el que contiene el elemento.
- En la secuencia de nodos secundarios de un elemento los atributos aparecen siempre los primeros. Por tanto, las conversiones de tipos pueden resultar en una reorganización de los nodos.
- En el caso de contenido mixto (datos de caracteres con secundarios de elementos, como un elemento de párrafo que contiene texto y también elementos en negrita y cursiva) no se admiten nodos de texto consecutivos, por lo que puede que se anexen automáticamente.
- Puede definir si, al seleccionar varios nodos para convertirlos en un solo tipo, el cambio se debe aplicar directamente o si el programa debe avisarle antes de llevarlo a cabo.
- Para editar texto sin formato en una celda o fila seleccione la celda o fila principal y haga clic en el icono **Editar como texto sin formato** de la barra de herramientas. Al editar texto sin formato, las entidades y el marcado de esa celda o fila no se resuelven, respectivamente, en glifos y componentes de la vista Cuadrícula (por lo que se pueden editar).

Finalización automática

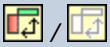
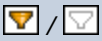
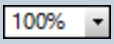
La finalización automática se habilita si el documento XML que se edita está asociado con un esquema.

La finalización automática proporciona opciones de entrada en la ubicación del cursor (*imagen siguiente*). Estas opciones, que se basan en las definiciones del esquema, (i) aparecen en recuadros emergentes en la ventana principal o (ii) las proporcionan los ayudantes de entrada. Tanto los recuadros emergentes como los ayudantes de entrada incluyen una lista con todas las opciones de entrada que son válidas en la posición donde está el cursor. Para navegar por la lista de opciones basta con usar las teclas de dirección. Seleccione una entrada del recuadro emergente o haga doble clic en una entrada del ayudante de entrada para insertarlo.



Botones de comandos de la barra de herramientas y dentro de las celdas

En la vista Cuadrícula XML los botones que hay dentro de las celdas o en la barra de herramientas de la vista Cuadrícula[293] permiten realizar acciones en el contexto de la selección actual (celda o columna). Es más rápido usar estos botones de comando para las acciones que se usan más a menudo. En la tabla siguiente describimos los comandos de estos botones. Los comandos relacionados con la visualización tienen fondo azul. Los comandos para editar el documento (estructura y contenido) tienen fondo beige. Si necesita otros comandos en el contexto de alguna de las celdas puede usar el menú contextual[706] de la celda. Nota: si pasa el cursor del ratón por encima de un botón aparece información rápida con el nombre del comando del botón en cuestión.

	Optimiza el ancho de las columnas para que se ajuste a su contenido.
	Si aparece en color, la Vista Diseño de tabla[692] está activada, si está desactivada aparece en gris. Haga clic en el icono para cambiar la vista
	Este comando cambia las filas por las columnas y viceversa
	Si está naranja significa que hay un filtro[696] para la tabla, el objeto o la matriz en cuestión. Si el icono está blanco, el filtro está desactivado. Para activar/desactivar el filtro, haga clic en el icono. Para editar la expresión, haga doble clic en el icono.
	Ajuste automático de línea que afecta a todas las celdas
	Nivel de acercamiento de la vista Cuadrícula
	Se habilita cuando se selecciona un encabezado de columna en la vista Tabla. Los botones ordenan las filas de la tabla, respectivamente, en orden alfabético ascendente o descendente.

Nota: para ver una descripción de otros comandos de la barra de herramientas consulte el apartado Menú contextual de la vista Cuadrícula[323].

Validar al modificar

El modo *Validar al editar* está activado por defecto. Cuando esta opción está activada, se van comprobando el formato y la validación conforme edita el documento en la vista Cuadrícula. Para poder llevar a cabo la validación (además de comprobar el formato), debe asignar un esquema DTD o XML al documento. Los errores se indican marcando el texto erróneo en rojo y marcando el lugar con un signo de exclamación rojo. Si la aplicación dispone de una solución para un error, aparece un icono en forma de bombilla en la línea que genera

© 2016-2022 Altova GmbH

el error. Si pasa el cursor por encima de este icono, aparecen todas las soluciones para el error en un menú contextual. Seleccione una solución para corregir el error inmediatamente. Para más información consulte Validar un documento XML[335].

El modo *Validar al editar* se puede activar y desactivar (i) con el comando de menú **XML | Validar al editar**[1344], (ii) el botón **Validar al editar** de la barra de herramientas o (iii) con la opción *Al editarlos* de las las opciones de validación del cuadro de diálogo "Opciones"[1584].

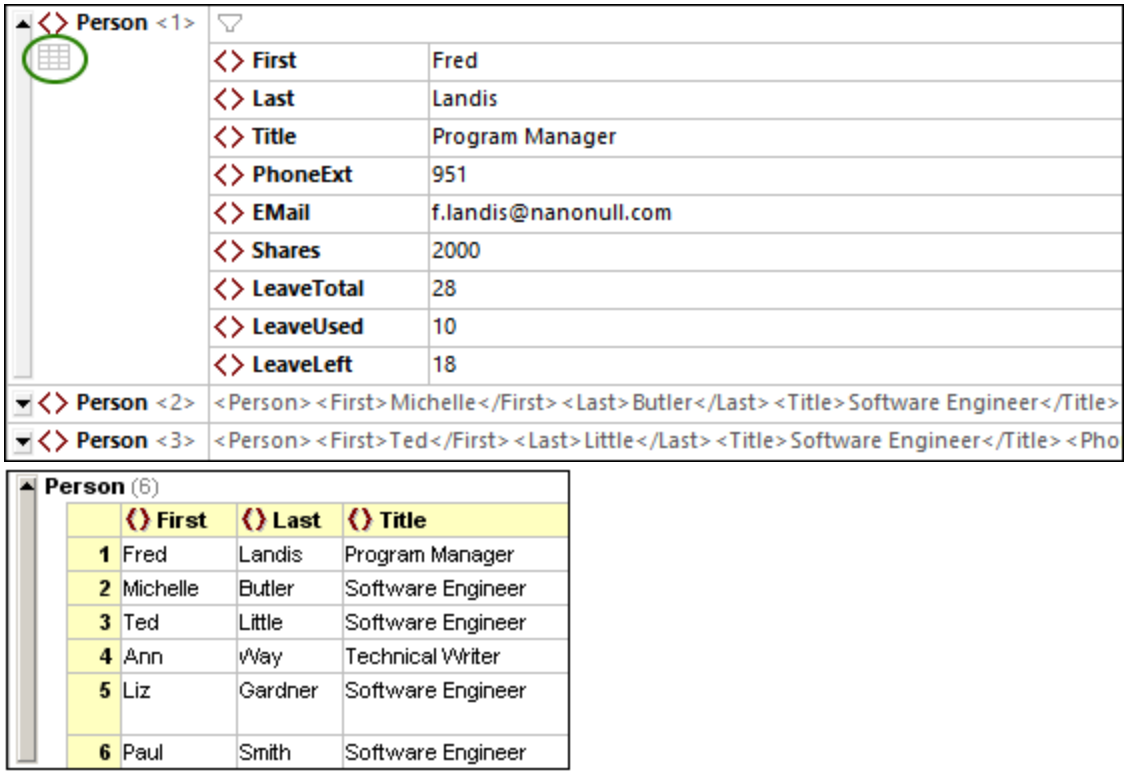
Buscar y reemplazar

Puede usar los comandos **Buscar (Ctrl+F)** y **Reemplazar (Ctrl+H)** del menú **Edición** para buscar y reemplazar texto en la vista Cuadrícula. Los resultados están resaltados en naranja y las celdas relevantes también están resaltadas en naranja.

Mostrar como tabla

Los elementos extensibles aparece en una Vista Cuadrícula *estándar* uno tras otro, progresando verticalmente hacia abajo en el orden de los documentos (*imagen siguiente*). Sin embargo, al mostrar los elementos extensibles como *filas de una tabla* se habilitan más opciones de edición. En la imagen siguiente, se repite el elemento `Person`. La imagen izquierda muestra la vista Cuadrícula estándar, donde el primer elemento `Person` aparece ampliado, mientras que las instancias siguientes aparecen contraídas. En la imagen de la derecha aparecen los elementos extensibles `Person` como filas de una tabla.

Para cambiar a la Vista Tabla haga clic en el icono **Modo tabla** (*dentro de un círculo verde en la imagen izquierda*). Al cambiar a la vista de tabla el icono aparece en color (*imagen siguiente derecha*).



La vista Tabla tiene una ventaja a la hora de editar los documentos: puede manipular columnas y filas enteras

en relación con otras columnas y filas de la tabla. Esto permite operaciones como ordenar las filas de tabla según los valores de una columna. Por ejemplo, en la imagen anterior derecha los seis elementos **Person** se pueden ordenar en base a sus elementos secundarios **Last** con un simple comando en la barra de herramientas de la vista Cuadrícula 293. Esta operación es más sencilla que ejecutar una transformación XSLT, que sería la forma habitual de ordenar un conjunto de nodos XML.

Para más información consulte Visualizar tablas 309.

5.4.4 Arrastrar y colocar en la vista Cuadrícula

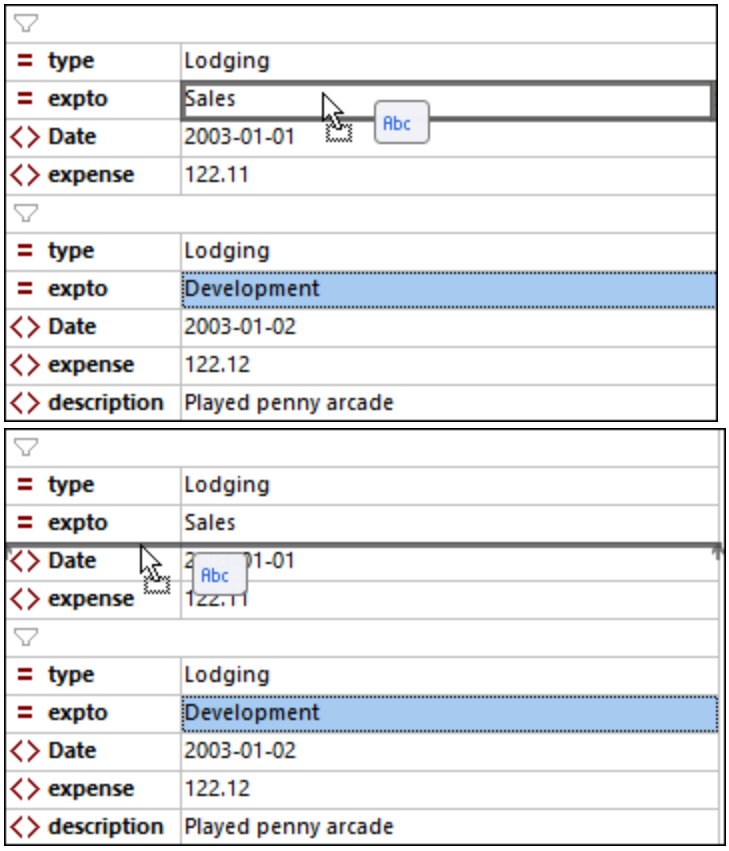
La vista Cuadrícula permite arrastrar fragmentos de documentos XML a la vista Cuadrícula desde un documento que esté abierto en XMLSpy, una aplicación externa o incluso un sitio web.

Al colocar el cursor sobre el nodo de destino aparece una capa que no solo ofrece información sobre lo que se está arrastrando (tipo y cantidad de elementos), sino también sobre cómo se creará ese elemento cuando se coloque. Por ejemplo, en la imagen siguiente se está arrastrando (moviendo) el valor **Development**. La ventana emergente **Abc** indica que se trata de un valor de cadena.

- Si pasa el cursor sobre un campo de valor (como en la imagen, a la izquierda) se marca todo el campo, lo que indica que el valor de ese campo será reemplazado por el valor del campo que se está arrastrando.
- Si pasa el cursor del ratón sobre un nombre de nodo (como en la imagen, a la derecha, con **Date**), entonces aparece una línea de inserción que indica que la cadena de valor se colocará allí para crear un nodo de texto en la estructura.

Nota: puede deshacer cualquier acción con **Ctrl+Z** o el comando de menú **Edición | Deshacer**.

© 2016-2022 Altova GmbH

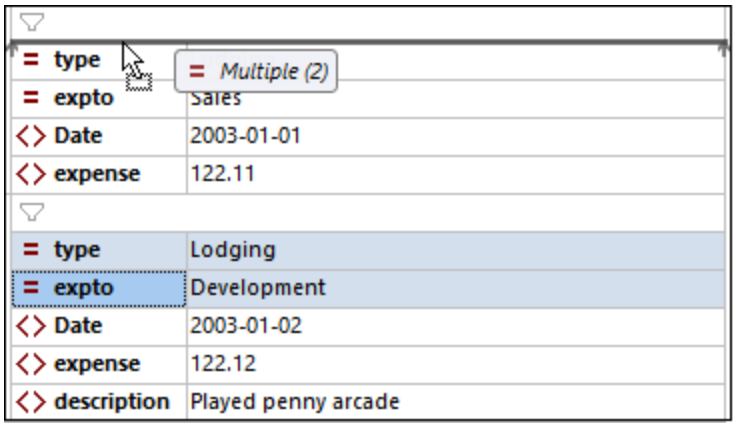


Información de la capa de arrastre

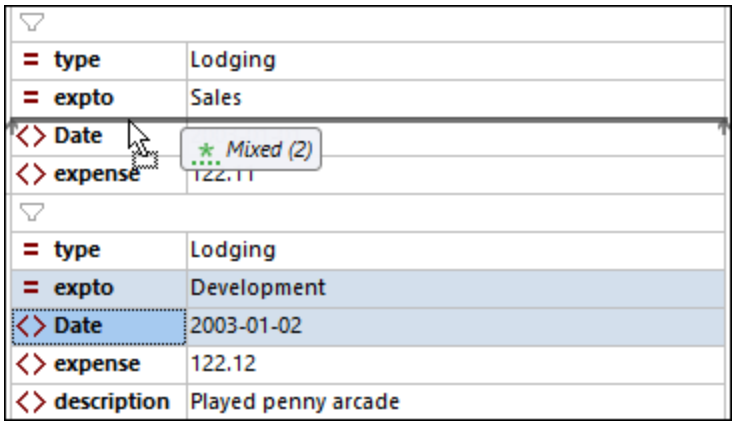
Estos son los tipos de información de la capa de arrastre.

Vista Cuadrícula normal

- El valor que se coloca en un campo de valor mueve el valor de origen para sobrescribir el valor del campo de destino (*imagen anterior a la izquierda*).
- El valor que se coloca en la estructura como nodo crea un nodo de texto (*imagen anterior a la derecha*).
- Si hay varios nodos de un mismo tipo se muestran el tipo de nodo que se va a colocar y el número de nodos. Tenga en cuenta que para seleccionar un nodo (y no su valor) debe hacer clic en el nombre del nodo. En la imagen siguiente la información de la capa de arrastre indica que se van a colocar dos atributos.

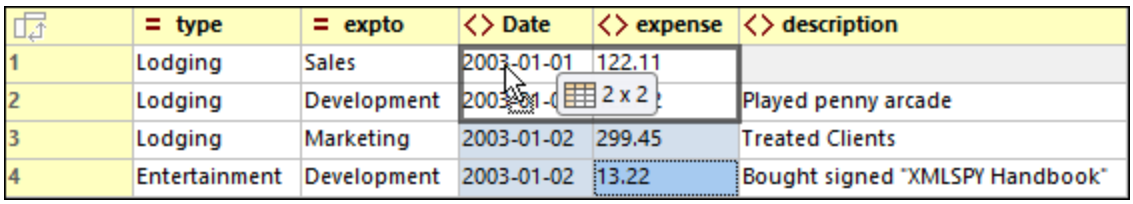


- Si hay varios nodos de distintos tipos se muestran el texto *Mixed* y la cantidad de nodos que hay en el portapapeles (*imagen siguiente*). Tenga en cuenta que para seleccionar un nodo (y no su valor) debe hacer clic en el nombre del nodo.



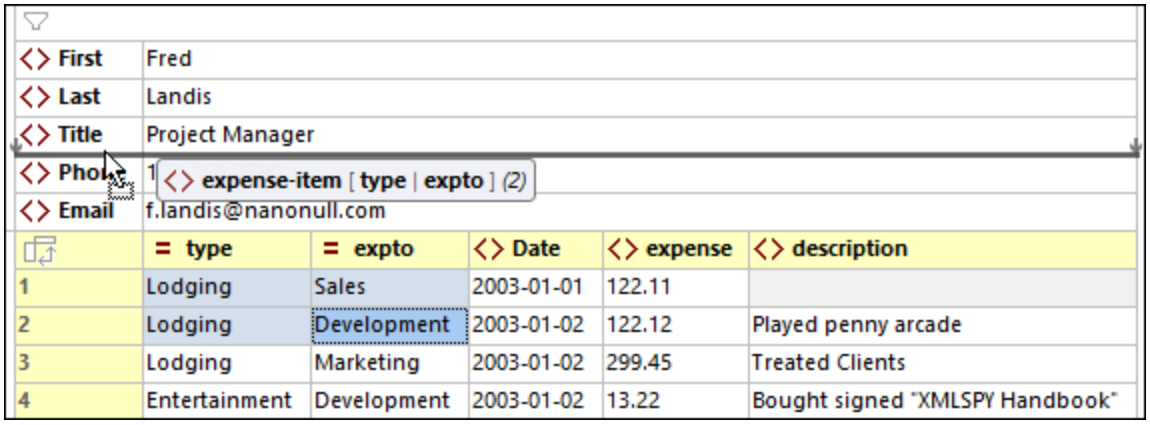
Vista Diseño de tabla

- Al arrastrar valores, el número de celdas seleccionadas viene indicado por `columna x filas` (*imagen siguiente*). La matriz de celdas arrastradas reemplaza a la matriz correspondiente. La celda de destino (en la que coloca la matriz) recibe el valor de la celda superior izquierda de la matriz. El resto de celdas rellena las celdas de destino hacia la derecha y hacia abajo. Por ejemplo, en la imagen siguiente las celdas azules se arrastran hasta colocarlas en la primera celda `Date`. Las celdas reemplazan las celdas que se habían marcado. Si la matriz de celdas que se arrastran supera los límites de la tabla se añaden las filas o columnas necesarias para que quepan todas las celdas. En este caso los límites que se han tenido que ampliar aparecen como líneas discontinuas.



 © 2016-2022 Altova GmbH

- Si el destino es una estructura de nodos y no una celda, el nodo que se añade se indica junto con (i) las columnas que se van a añadir (entre corchetes) y (ii) el número de filas del elemento nuevo (*imagen siguiente*).

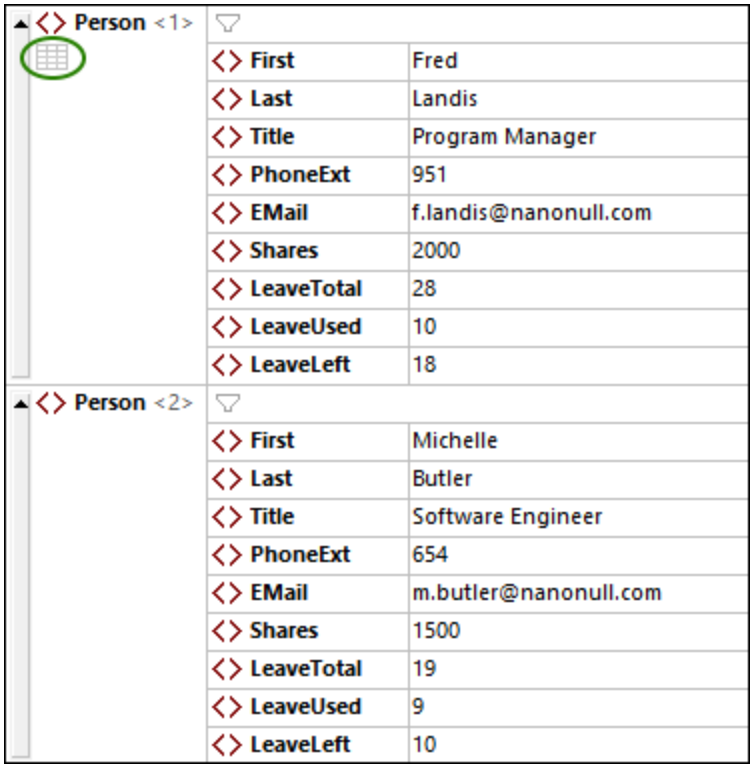


- Puede mover columnas de tabla seleccionándolas (para ello haga clic en su encabezado) y arrastrándolas hasta el encabezado de columna adyacente al que quiere mover. Una línea de inserción le indicará a qué lado de la columna de destino se colocará la columna que arrastra.

5.4.5 Vista de tablas

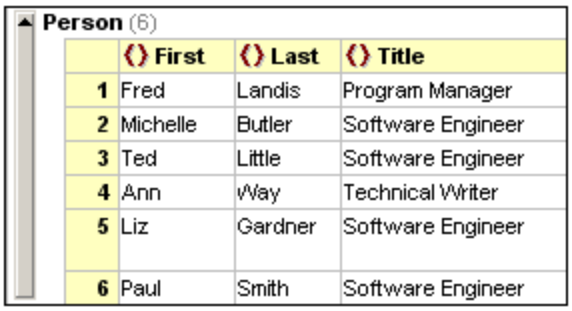
Acerca de la vista Diseño de tabla

En la vista Cuadrícula estándar, si un elemento se repite, las instancias de ese elemento aparecen una después de la otra en el orden del documento. En la imagen siguiente, por ejemplo, se repite el elemento **`Person`**.



Person <1>	
First	Fred
Last	Landis
Title	Program Manager
PhoneExt	951
EMail	f.landis@nanonull.com
Shares	2000
LeaveTotal	28
LeaveUsed	10
LeaveLeft	18
Person <2>	
First	Michelle
Last	Butler
Title	Software Engineer
PhoneExt	654
EMail	m.butler@nanonull.com
Shares	1500
LeaveTotal	19
LeaveUsed	9
LeaveLeft	10

Un elemento que se repite, como **`Person`**, también se puede mostrar como tabla (*imagen siguiente*). En una representación de tabla, los nodos secundarios del elemento que se repite forman las columnas de la tabla, mientras que las instancias de ese elemento forman las filas. Para cambiar a la vista Texto haga clic en el icono **Modo tabla** que aparece en la primera instancia del elemento que se repite (*resaltado en verde en la imagen anterior*). Al cambiar a la vista de tabla el icono **Modo tabla** aparece en color (*imagen siguiente*).



Person (6)

	First	Last	Title
1	Fred	Landis	Program Manager
2	Michelle	Butler	Software Engineer
3	Ted	Little	Software Engineer
4	Ann	Way	Technical Writer
5	Liz	Gardner	Software Engineer
6	Paul	Smith	Software Engineer

La vista Tabla tiene una ventaja a la hora de editar los documentos: puede manipular columnas y filas enteras en relación con otras columnas y filas de la tabla. Esto permite operaciones como ordenar las filas según los valores de una columna. Por ejemplo, en la imagen anterior los seis elementos `Person` se pueden ordenar en base a sus elementos secundarios **`Last`** con una simple operación de la IGU. Esta operación (*detalles más abajo*) es más sencilla que ejecutar una transformación XSLT, que es la forma habitual de ordenar un conjunto de nodos XML.

Nota

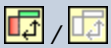
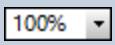
- La vista de diseño de tabla se puede aplicar a una secuencia de elementos con el mismo nombre.

© 2016-2022 Altova GmbH

- También está disponible para un único elemento. El icono **Modo tabla** se vuelve visible cuando se hace clic en el elemento en cuestión.
- Puede definir los colores de la Vista Diseño de Tabla en el cuadro de diálogo Opciones: Fuentes y colores | Vista Cuadrícula | Colores de celda[1606].

Iconos para ver y editar en la vista Diseño de tabla

Estos iconos están disponibles en la vista Diseño de tabla y representan funciones de visualización y edición. Los puede encontrar también en las celdas de la tabla o en la barra de herramientas de la vista Cuadrícula.

	Optimiza el ancho de las columnas para que se ajuste a su contenido.
	Si aparece en color, la Vista Diseño de tabla[692] está activada, si está desactivada aparece en gris. Haga clic en el icono para cambiar la vista
	Este comando cambia las filas por las columnas y viceversa
	Si está naranja significa que hay un filtro[696] para la tabla, el objeto o la matriz en cuestión. Si el icono está blanco, el filtro está desactivado. Para activar/desactivar el filtro, haga clic en el icono. Para editar la expresión, haga doble clic en el icono.
	Ajuste automático de línea que afecta a todas las celdas
	Nivel de acercamiento de la vista Cuadrícula
	Se habilita cuando se selecciona un encabezado de columna en la vista Tabla. Los botones ordenan las filas de la tabla, respectivamente, en orden alfabético ascendente o descendente.

Editar en la vista Diseño de tabla

A continuación explicamos las acciones que puede realizar en la vista Diseño de tabla.

Agregar una fila de tabla (una instancia nueva del elemento de la tabla que se repite)
Para agregar una tabla nueva, es decir, otra instancia del elemento de la tabla que se repite:

1. Haga clic con el botón derecho en la *celda que contiene el número de la fila* de encima o de debajo de donde quiere añadir la línea nueva.
2. Seleccione el comando **Insertar después (Ctrl+Entrar)** para agregar una fila bajo la fila seleccionada o **Insertar antes** para añadirla antes de la fila seleccionada. Estos comandos también están disponibles en el menú XML[1332] y la barra de herramientas de la vista Cuadrícula[293].

La fila nueva se crea como un nodo de elemento. Puede cambiar el tipo del nodo si quiere (*véase más abajo*).

Agregue elementos del mismo nivel o secundarios a una celda de tabla
Si una celda de tabla representa un elemento secundario de una fila de tabla puede asignar a este elemento secundario otro nodo del mismo nivel o secundario. Haga clic con el botón derecho en la celda de tabla y seleccione el comando **Anexar (Ctrl+Alt+A)** o **Agregar secundario (Ctrl+Alt+Entrar)**. Estos comandos también están disponibles en el menú XML[1332] y la barra de herramientas de la vista Cuadrícula[293]. La fila nueva se crea como un nodo de elemento. Puede cambiar el tipo del nodo si quiere (*véase más abajo*).

Envolver celda en un elemento
Puede crear un elemento que envuelva una celda de tabla El elemento nuevo se crea entre la celda y su celda de nivel superior. Para ello haga clic con el botón derecho y seleccione el comando **Envolver en elemento (Ctrl+Alt+W)**. menú XML[1332] y la barra de herramientas de la vista Cuadrícula[293].

Agregar una columna de tabla (nodo secundario nuevo de todas las instancias de un elemento de tabla que se repite)
Para agregar una columna nueva, es decir, un nodo secundario nuevo de todas las instancias del elemento de tabla que se repite:

1. Haga clic con el botón derecho en un encabezado de columna o en una celda de columna que no esté vacía.
2. Seleccione el comando **Insertar después (Ctrl+Entrar)** para añadir una columna a la derecha de la columna seleccionada, o **Insertar antes (Ctrl+Mayús+Entrar)** para añadir una columna a la izquierda de la columna seleccionada. Estos comandos también están disponibles en el menú XML[1332] y la barra de herramientas de la vista Cuadrícula[293].

La columna nueva se crea como un nodo de elemento. Puede cambiar el tipo del nodo si quiere (*véase más abajo*).

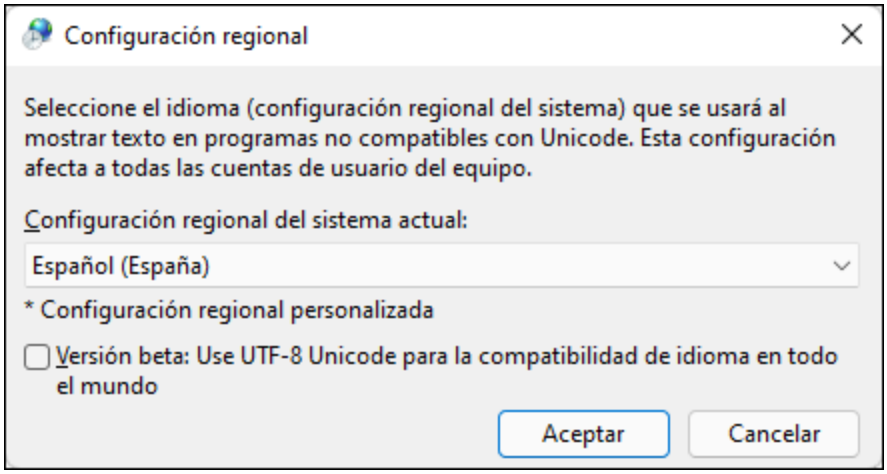
Cambiar los tipos de nodo y los nombres de las columnas
Para cambiar el tipo de nodo en una columna haga clic en el icono del tipo de nodo que hay en la columna y, en el menú que aparece, seleccione el tipo nuevo. El tipo de este nodo cambia para todas las instancias del elemento que se repite.

Para cambiar el nombre del encabezado de una columna, haga doble clic en su nombre y edítelo. El nombre de este nodo cambia en todas las instancias del elemento que se repite.

Ordenar las filas de tablas según los valores de una columna seleccionada
Puede ordenar las filas de una tabla en función de los valores de uno de sus nodos secundarios (una columna). Por ejemplo, puede ordenar filas en función de la columna *Lastname* para colocar los elementos que se repiten en orden alfabético. Para ordenar una columna seleccione el encabezado de la columna correspondiente, haga clic en los comandos **Orden ascendente** u **Orden descendente** de la barra de herramientas de la vista Cuadrícula[293]. Nota:estos comandos también están disponibles en el menú XML[1332].

Sort order in some languages, especially those with non-Latin alphabets, may benefit from enabling the beta Unicode UTF-8 support in the Language Region Settings dialog of Windows 10 (or later). Do this as follows: Go to your Windows *Settings* dialog and search for *Language Settings*. Under *Related Settings*, click *Administrative language settings*. In the Region dialog that appears, go to the *Administrative* tab, and, under *Language for non-Unicode programs*, click *Change system locale*. In the Region Settings dialog that appears (*screenshot below*), select the option *Beta: Use Unicode UTF-8 for worldwide language support* and click **OK**.

© 2016-2022 Altova GmbH



La Vista Diseño de tabla y aplicaciones externas

Puede aprovechar la estructura de la tabla para intercambiar datos entre la vista Diseño de tabla y una aplicación de hojas de cálculo (como MS Excel). Para mover datos desde la vista Diseño de tabla:

1. En la vista Diseño de tabla seleccione los nodos que quiere copiar. Para ello haga clic en celdas, encabezados de columna, encabezados de fila o en la tabla entera. Si hace clic en la tabla entera o en encabezados de columna, también se copiarán los encabezados de columna. (En la imagen siguiente se han seleccionado las filas 2 y 3.)

item (6)

	= partNum	{} productName	{} quantity	{} price	{} shipDate	{} *ipo*:comment
1	833-AA	Lapis necklace	2	99.95	1999-12-05	Need this for the holidays!
2	748-OT	Diamond heart	1	248.90	2000-02-14	Valentine's day packaging.
3	783-KL	Uncut diamond	7	79.90	2000-01-07	
4	238-KK	Amber ring	3	89.90	2000-01-07	With no inclusions, please.
5	229-OB	Pearl necklace	1	4879.00	1999-12-05	
6	128-UL	Jade earring	5	179.90	2000-02-14	

2. Seleccione el comando del menú contextual Copiar | Copiar como texto separado por comas[1282].
3. Pegue los datos directamente en el programa de hojas de cálculo.

	A	B	C	D	E	F
1	238-KK	Amber ring	3	89.9	01/07/2000	With no inclusions, please.
2	229-OB	Pearl necklace	1	4879	12/05/1999	
3						
4						

El intercambio de datos funciona también en la dirección inversa. Es decir, que también puede copiar datos de la hoja de cálculo y pegarlos en la vista Diseño de tabla. Para ello siga estas instrucciones:

1. Seleccione un rango en la aplicación externa y cópielo en el portapapeles (con **Ctrl+C**).

2. Seleccione una sola celda en la vista Diseño de tabla.
3. Pegue los datos con **Ctrl+V**.

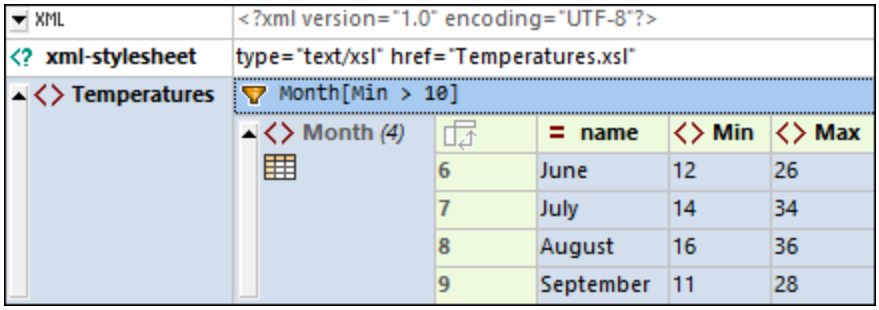
Los datos se pegarán en la tabla con una estructura equivalente a la estructura original y empezando en la celda seleccionada en la vista Tabla. Cabe destacar que:

- si ya existen datos en estas celdas de la vista Tabla, los datos nuevos sobrescribirán los datos originales.
- si no hay suficientes filas y/o columnas para alojar los datos nuevos, se crearán las filas que hagan falta.
- los datos se convierten en el contenido de los elementos representados por las celdas correspondientes.

Para otras tareas de intercambio complejo de datos use las funciones import/export[1452] de XMLSpy.

5.4.6 Filtros

En la vista Cuadrícula XML se puede aplicar un filtro a un nodo de elemento para filtrar sus descendientes. En los sitios donde se pueden aplicar filtros aparece un icono de filtro en gris. Una vez se ha definido un filtro, el icono aparece en color (*imagen siguiente*). Los filtros se definen con expresiones XQuery 3.1. Por ejemplo, en la imagen siguiente se ha aplicado un filtro al nodo `Temperatures` para mostrar solamente los elementos `Month` que tienen un elemento secundario `Min` con un valor mayor que `10`. (Tenga en cuenta que en la imagen el número de índice de la instancia del elemento correspondiente aparece en la lista.)



Para configurar un filtro haga clic con el botón derecho en el elemento que quiere filtrar, seleccione **Filtro** en el menú contextual que aparece, introduzca la expresión XQuery y haga clic en **Entrar**. La celda *Filtro* viene indicada por el icono (*imagen siguiente*). Puede activar/desactivar el filtro haciendo clic en el icono *Filtro*.

Lista de documentos XML usados en la imagen anterior

```
<?xml version="1.0" encoding="UTF-8"?>
<Temperatures>
   <Month name="January">
      <Min>-5</Min>
      <Max>3</Max>
   </Month>
   <Month name="February">
      <Min>-16</Min>
      <Max>1</Max>
```


© 2016-2022 Altova GmbH

```
    </Month>
    <Month name="March">
       <Min>-9</Min>
       <Max>7</Max>
    </Month>
    <Month name="April">
       <Min>2</Min>
       <Max>16</Max>
    </Month>
    <Month name="May">
       <Min>8</Min>
       <Max>21</Max>
    </Month>
    <Month name="June">
       <Min>12</Min>
       <Max>26</Max>
    </Month>
    <Month name="July">
       <Min>14</Min>
       <Max>34</Max>
    </Month>
    <Month name="August">
       <Min>16</Min>
       <Max>36</Max>
    </Month>
    <Month name="September">
       <Min>11</Min>
       <Max>28</Max>
    </Month>
    <Month name="October">
       <Min>10</Min>
       <Max>26</Max>
    </Month>
    <Month name="November">
       <Min>-1</Min>
       <Max>14</Max>
    </Month>
    <Month name="December">
       <Min>-3</Min>
       <Max>9</Max>
    </Month>
</Temperatures>
```

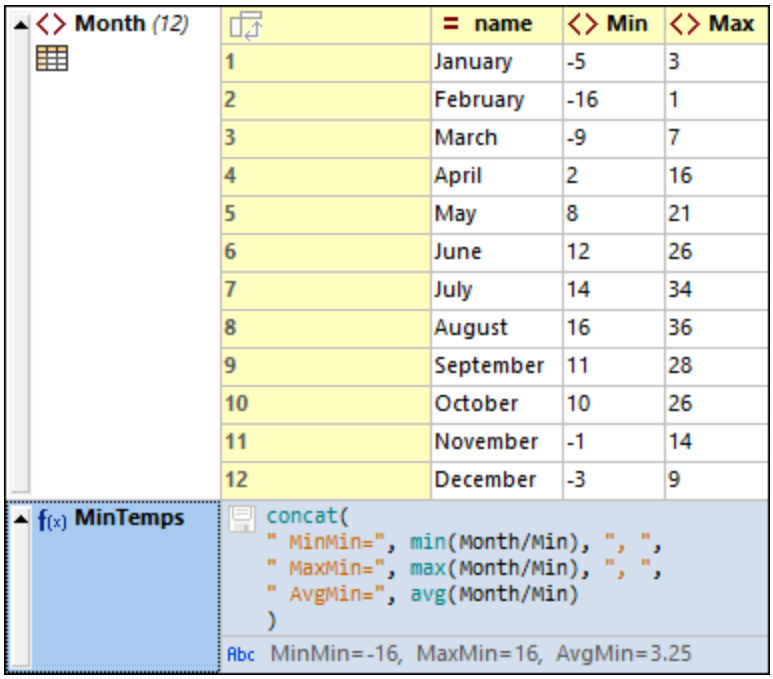
Tenga en cuenta estos aspectos sobre los filtros:

- Los filtros solo se pueden aplicar a nodos de elementos.
- El nodo de contexto de la expresión XQuery del filtro es el nodo actual. En la imagen anterior, por ejemplo, el nodo de contexto es **`Temperatures`**.
- Los filtros se pueden anidar. Se puede aplicar un filtro anidado al contenido filtrado del filtro principal.
- Para añadir una línea nueva en una expresión pulse **Ctrl+Entrar**. Esto es útil si quiere ver una expresión en varias líneas para leerla mejor.

- El filtro solo filtra el contenido visualmente. El contenido en sí permanece inalterado.
- Cada filtro funciona de forma independiente y no se ve afectado por el resto de filtros o fórmulas 316.
- Los filtros **no se almacenan** en el documento XML, pero se pueden conservar en un archivo de metadatos para su uso posterior. Si se guardan los filtros, estos se aplican automáticamente cuando el documento se vuelve a abrir en la vista Cuadrícula. Para indicar si quiere guardar los filtros siempre, solo en ocasiones o en absoluto vaya a **Herramientas | Opciones | Vista | Configurar la vista Cuadrícula** 1597.

5.4.7 Fórmulas

En la vista Cuadrícula XML las fórmulas usan expresiones XQuery 3.1 para calcular un resultado o generar un conjunto de nodos que se puede guardar en el documento. Las fórmulas se definen con expresiones XQuery 3.1. Por ejemplo, en la imagen siguiente se ha creado una fórmula llamada `MinTemps` que genera la temperatura mínima, máxima y media de un conjunto de temperaturas mínimas.



Month (12)

	name	Min	Max
1	January	-5	3
2	February	-16	1
3	March	-9	7
4	April	2	16
5	May	8	21
6	June	12	26
7	July	14	34
8	August	16	36
9	September	11	28
10	October	10	26
11	November	-1	14
12	December	-3	9

f(x) MinTemps

```
concat(
  " MinMin=", min(Month/Min), ", ",
  " MaxMin=", max(Month/Min), ", ",
  " AvgMin=", avg(Month/Min)
  )
```

Abc MinMin=-16, MaxMin=16, AvgMin=3.25

Siga estos pasos para crear una fórmula:

1. Añada un nodo nuevo donde quiera que aparezca la fórmula.
2. El nodo se crea por defecto como elemento. Cambiar el tipo de nodo 302 a Fórmula.
3. Haga doble clic en la celda que contiene el icono `f(x)` e introduzca el nombre de la fórmula (*imagen anterior*).
4. Puede hacer clic en este icono para guardar los resultados de la fórmula en el documento.
5. Haga doble clic en la celda de la expresión e introduzca la expresión, después pulse **Entrar**.

Lista de documentos XML usados en la imagen anterior

```
<?xml version="1.0" encoding="UTF-8"?>
```


© 2016-2022 Altova GmbH

```
<Temperatures>
   <Month name="January">
      <Min>-5</Min>
      <Max>3</Max>
   </Month>
   <Month name="February">
      <Min>-16</Min>
      <Max>1</Max>
   </Month>
   <Month name="March">
      <Min>-9</Min>
      <Max>7</Max>
   </Month>
   <Month name="April">
      <Min>2</Min>
      <Max>16</Max>
   </Month>
   <Month name="May">
      <Min>8</Min>
      <Max>21</Max>
   </Month>
   <Month name="June">
      <Min>12</Min>
      <Max>26</Max>
   </Month>
   <Month name="July">
      <Min>14</Min>
      <Max>34</Max>
   </Month>
   <Month name="August">
      <Min>16</Min>
      <Max>36</Max>
   </Month>
   <Month name="September">
      <Min>11</Min>
      <Max>28</Max>
   </Month>
   <Month name="October">
      <Min>10</Min>
      <Max>26</Max>
   </Month>
   <Month name="November">
      <Min>-1</Min>
      <Max>14</Max>
   </Month>
   <Month name="December">
      <Min>-3</Min>
      <Max>9</Max>
   </Month>
</Temperatures>
```

Lista de expresiones de fórmula usadas en la imagen anterior

```
concat(
" MinMin=", min(//Month/Min), ", ",
" MaxMin=", max(//Month/Min), ", ",
" AvgMin=", avg(//Month/Min)
)
```

Tenga en cuenta que:

- El nodo de contexto de la expresión XQuery de la fórmula es el nodo principal del nodo de la fórmula.
- Para añadir una línea nueva en una expresión pulse **Ctrl+Entrar**. Esto es útil si quiere ver una expresión en varias líneas para leerla mejor.
- Las fórmulas (es decir, las expresiones que definen las fórmulas) se almacenan en los metadatos del documento (que está en un archivo de metadatos que sirve para toda la aplicación). Las fórmulas también se guardan como instrucciones de procesamiento en el archivo XML.
- El resultado generado por una fórmula aparece en la celda bajo la expresión XQuery de la fórmula y se almacena en los metadatos del documento. El resultado también se puede guardar en ese documento. Para ello haga clic en el icono de disco que hay junto a la expresión XQuery. Haga o no clic en el icono, el resultado de la fórmula se calcula y se almacena en los metadatos del documento.
- Si el resultado de la fórmula es un cálculo, este se almacena como contenido de un elemento que tiene el nombre que haya asignado a la fórmula. Por ejemplo, en la fórmula `MinTemps` de la imagen anterior el resultado se almacena en un elemento llamado `MinTemps`.
- Vaya a **Herramientas | Opciones | Vista | Configurar la vista Cuadrícula**[1597] para: (i) indicar si quiere almacenar las fórmulas en los metadatos siempre, solo en ocasiones o nunca, (ii) indicar si quiere almacenar las fórmulas en el documento.

Fórmulas en tablas

Si todas las celdas de una columna de tabla (en Vista Diseño de tabla[692]) contienen la misma fórmula, entonces la fórmula aparece solamente una vez, en el encabezado de la columna (*imagen siguiente*). Los *resultados* del cálculo de la fórmula aparecen en las celdas correspondientes. La fórmula del encabezado de la columna es una representación de la vista Cuadrícula. En el documento XML (véase Vista Texto), la fórmula se repite por cada elemento fila de tabla.

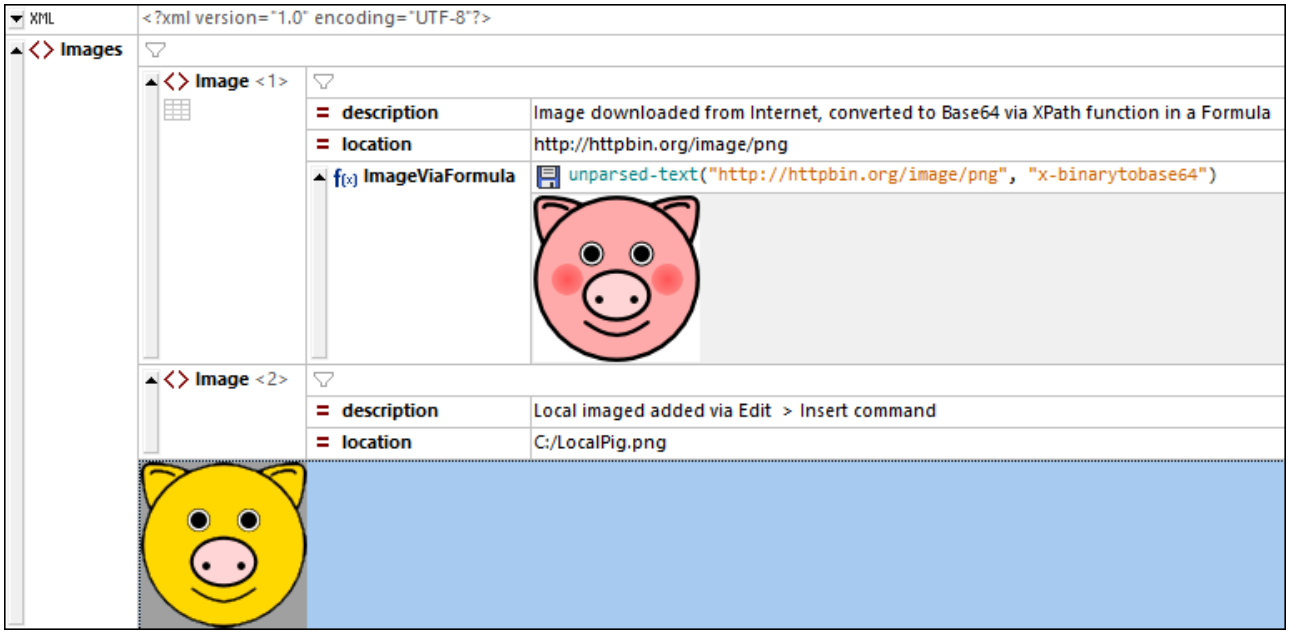
Month (12)		name	Min	Max	MidRange (Min + Max) div 2
	1	January	-5	3	# -1
	2	February	-16	1	# -7.5
	3	March	-9	7	# -1
	4	April	2	16	# 9
	5	May	8	21	# 14.5
	6	June	12	26	# 19
	7	July	14	34	# 24
	8	August	16	36	# 26
	9	September	11	28	# 19.5
	10	October	10	26	# 18
	11	November	-1	14	# 6.5
	12	December	-3	9	# 3

© 2016-2022 Altova GmbH

Si alguna de las fórmulas es diferente, cada fórmula aparece en su respectiva celda. Su todas las fórmulas de una columna de tabla son iguales que la que aparece en el encabezado y quiere crear una distinta para una sola celda, cambie a la vista Diseño de tabla y edite la fórmula de esa celda. Al cambiar a la vista Diseño de tabla las fórmulas aparecen cada una en su correspondiente celda.

5.4.8 Imágenes

Las imágenes se pueden visualizar directamente en la vista Cuadrícula (*imagen siguiente*). Para ello la imagen debe insertarse cifrada en Base64.



Hay dos formas de insertar imágenes cifradas en Base64 en una celda de la vista Cuadrícula:

- Crear un nodo de tipo *Fórmula*. En la celda de la expresión de la fórmula (*imagen anterior*) introduzca esta expresión XPath para cifrar una imagen en Base64: `unparsed-text("<Image-URL>", "x-binarytobase64")`. La función XPath `unparsed-text` cifra la imagen en Base64. La imagen aparece bajo la celda que contiene la expresión de fórmula.
- Ponga el cursor en la celda donde desea añadir la imagen. Por ejemplo, en el ejemplo anterior la imagen se añade como contenido del segundo elemento `Image` seleccionando para ello el elemento `Image`. Después seleccione **Editar | Insertar | Insertar archivo externo codificado**. En el cuadro de diálogo que aparece: (i) introduzca la ruta de acceso a la imagen que quiera insertar (puede ser una ruta local o de Internet). (Ii) seleccione *Base 64*, (iii) seleccione *Crear texto*. La imagen se cifra en Base64 y el texto cifrado se introduce en la celda. Sin embargo, lo que se ve dentro de la celda no es el texto cifrado sino la imagen (*imagen anterior*).

Las URLs de imágenes se pueden indicar de varias formas, también como rutas relativas:

- `http://httpbin.org/image/png`
- `file:///c:/LocalPig.png`
- `C:/LocalPig.png`

- `LocalPig.png`

Se admiten la mayoría de formatos comunes de imagen. Entre ellos se encuentran PNG, JPEG, BMP y GIFs animados. SVG se lee como un documento XML; en la vista Cuadrícula la imagen aparece como el último secundario del elemento .

Guardar una cadena cifrada en base 64 como imagen

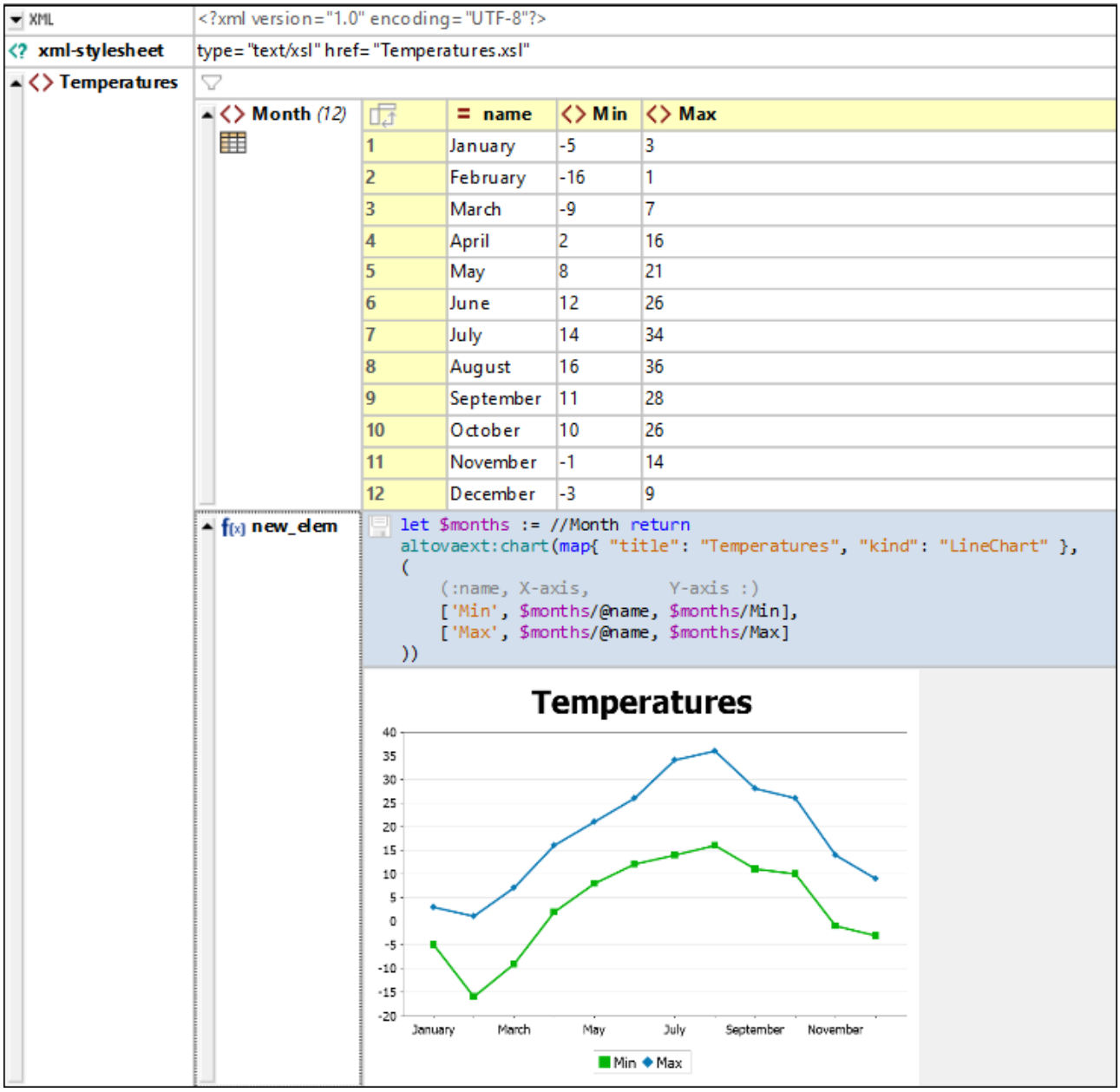
Para guardar una cadena cifrada en base 64 en formato imagen haga clic con el botón derecho del ratón en la imagen o su celda y seleccione el comando **Guardar como imagen**. (Tenga en cuenta que la imagen está cifrada como una cadena en base 64 en el archivo activo pero aparece en la vista Cuadrícula como una imagen.) En el cuadro de diálogo que aparece, seleccione la ubicación donde quiere guardar la imagen y asigne un nombre al archivo de imagen. La extensión del archivo (`.png`, `.gif`, `.svg`, etc.) se detecta automáticamente en el cifrado en base 64 y aparece en el cuadro de diálogo **Guardar**. Haga clic en Guardar para terminar.

También puede ejecutar esta acción con el comando de menú **Edición | Guardar como imagen**.

5.4.9 Gráficos

Puede crear gráficos en la vista Cuadrícula XML usando la función de extensión XPath/XQuery de Altova **`altovaext:chart`** (*imagen siguiente*). Más abajo explicamos esta función de extensión. También puede encontrar la explicación junto con las de otras funciones de extensión en el apartado Funciones de extensión para gráficos 2200. La extensión **`altovaext:chart`** de la imagen siguiente se usa en una expresión XQuery `Let` que está definida dentro de una fórmula de la vista Cuadrícula 316. El gráfico aparece como imagen bajo la fórmula. (Puede usar el extracto de código XML que incluimos a continuación y la función de expresión XQuery para probar la función para gráficos.)

© 2016-2022 Altova GmbH



Lista de documentos XML usados en la imagen anterior

```
<?xml version="1.0" encoding="UTF-8"?>
<Temperatures>
   <Month name="January">
      <Min>-5</Min>
      <Max>3</Max>
   </Month>
   <Month name="February">
      <Min>-16</Min>
```

```
      <Max>1</Max>
   </Month>
   <Month name="March">
      <Min>-9</Min>
      <Max>7</Max>
   </Month>
   <Month name="April">
      <Min>2</Min>
      <Max>16</Max>
   </Month>
   <Month name="May">
      <Min>8</Min>
      <Max>21</Max>
   </Month>
   <Month name="June">
      <Min>12</Min>
      <Max>26</Max>
   </Month>
   <Month name="July">
      <Min>14</Min>
      <Max>34</Max>
   </Month>
   <Month name="August">
      <Min>16</Min>
      <Max>36</Max>
   </Month>
   <Month name="September">
      <Min>11</Min>
      <Max>28</Max>
   </Month>
   <Month name="October">
      <Min>10</Min>
      <Max>26</Max>
   </Month>
   <Month name="November">
      <Min>-1</Min>
      <Max>14</Max>
   </Month>
   <Month name="December">
      <Min>-3</Min>
      <Max>9</Max>
   </Month>
</Temperatures>
```

Lista de expresiones XQuery para generar el gráfico que aparece en la imagen anterior

```
let $months := //Month return
altovaext:chart(map{ "title":"Temperatures", "kind":"LineChart" },
(
    (:name, X-axis,       Y-axis   :)
    ['Min', $months/@name, $months/Min],
```


© 2016-2022 Altova GmbH

```
        ['Max', $months/@name, $months/Max]
    ))
```

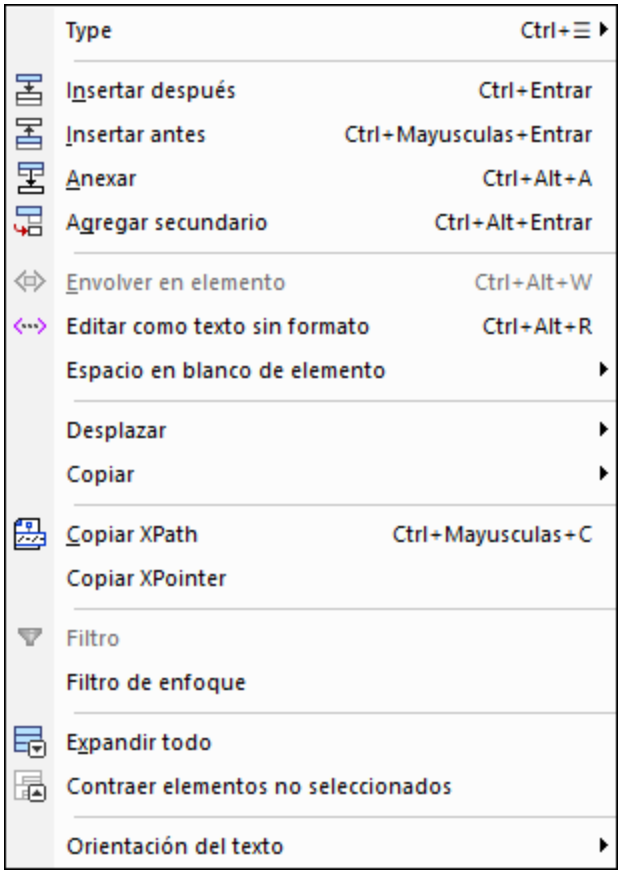
Usar las funciones de extensión para gráficos de Altova

- La función de extensión para gráficos `altovaext:chart` debe usar el prefijo del espacio de nombres `altovaext:`.
- La función `altovaext:chart` toma dos argumentos: (I) información de configuración del gráfico y (ii) información de series de datos de gráficos.
- La información de configuración del gráfico es el primer argumento de `altovaext:chart`. Se trata de una serie sin ordenar de cuadro pares clave-valor. Estos pares se aplican (i) al título del gráfico (la clave es `title`), (ii) al tipo de gráfico (la clave es `kind`, consulte Funciones de extensión[2200] para gráficos para ver qué tipos existen), (iii) ancho del gráfico en píxeles (solo números enteros, la clave es `width`) y (iv) altura del gráfico en píxeles (solo números enteros, la clave es `height`). Si no se indican el ancho o la altura, los valores que faltan se calculan automáticamente en función de los datos.
- La serie de datos del gráfico es el segundo argumento de `altovaext:chart`. Cada serie de datos es una matriz de tamaño 3 (i) el nombre de la serie de datos, (ii) los valores del eje X y (iii) los valores del eje Y. Si quiere crear varias series (por ejemplo, para que cada serie represente una línea, como en el ejemplo anterior) debe crear una secuencia de varias matrices. El ejemplo anterior tiene tres series de datos, que corresponden a las temperaturas mínima, máxima y media. Los datos de los ejes X e Y son referenciados desde la secuencia de todos los elementos `Month`.

Para más información consulte el apartado Funciones de extensión para gráficos[2200].

5.4.10 Menú contextual en la vista Cuadrícula

Al hacer clic con el botón derecho en una celda en la vista Cuadrícula aparece un menú contextual (*imagen siguiente*) con comandos que puede usar para editar el contenido relacionado con la celda y modificar la presentación de los datos. También puede acceder al menú contextual pulsando la tecla **Menú** del teclado. Estos comandos se describen a continuación.



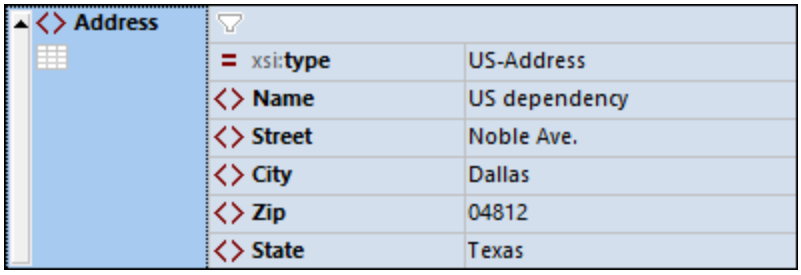
Tipo
Si pasa el cursor del ratón sobre el comando **Tipo** aparece un submenú en el que puede seleccionar el tipo de componente[302]. También puede acceder al submenú **Tipo** con el atajo de teclado **Ctrl+Menú**.

Insertar antes/después, anexar, agregar secundario
Los comandos **Insertar** y **Anexar** agregan un elemento del mismo nivel. Ese elemento nuevo se añade antes o después del elemento seleccionado según si usa **Insertar después** o **Insertar antes**, y en el caso de **Anexar** el elemento se añade como último secundario del elemento seleccionado. El comando **Agregar secundario** anexa un elemento nuevo como secundario. Haga doble clic en la celda del nombre del elemento recién añadido para editar su nombre. Para cambiar su tipo haga clic en el icono del elemento (a la izquierda de su nombre) y seleccione el tipo de nodo que prefiera.

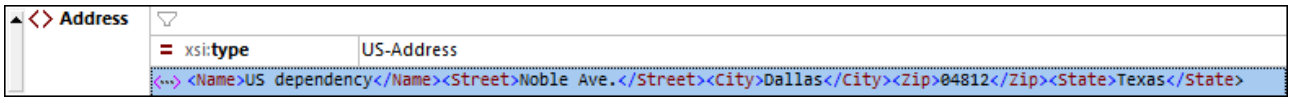
Envolver en elemento
Con este comando se añade al elemento seleccionado un elemento de nivel superior con un elemento predeterminado cuyo nombre puede cambiar haciendo doble clic en él.

Editar como texto sin formato
Permite editar texto del elemento seleccionado como texto sin formato. Este comando es útil si está editando contenido complejo, como es el código HTML. Por ejemplo, en la imagen siguiente se ve el elemento `Address` en la vista Cuadrícula.

© 2016-2022 Altova GmbH



Si cambia la vista del elemento `Address` a texto sin formato (*imagen siguiente*) la estructura de la cuadrícula se convierte en un solo elemento de texto sin formato que puede editar en la vista Texto.



Espacio en blanco de elemento
Este comando está disponible en los nodos de elementos y añade el atributo **`xml:space`** al marcado del elemento y asigna al atributo el valor que seleccione en el submenú del comando.

- Sin embargo, si selecciona el valor *Omitir*, entonces el atributo **`xml:space`** no se añade, lo que normaliza el espacio en blanco.
- El valor *Conservar* conserva los espacios en blanco significativos.
- El valor *Predeterminado* toma el valor indicado para ese elemento en el esquema, lo que puede ser útil para sobrescribir un valor heredado.

Subir/Bajar/Desplazar a la izquierda/Desplazar a la derecha
Si es posible mover un elemento, entonces se habilitan los comandos correspondientes para subirlo, bajarlo o desplazarlo a la izquierda o a la derecha. Seleccione el comando correspondiente para mover el elemento como necesite.

Copiar
Estos comandos se pueden usar para copiar la selección actual. Solo se habilitan las opciones que se pueden aplicar al componente en cuestión.

Comando	Descripción
Copiar como texto XML	La selección actual se serializa como marcado XML
Copiar como texto separado por comas	La selección de tabla actual se serializa como TSV (valores separados por tabulador)
Copiar como imagen	La celda de imagen actual se copia como imagen

Tenga en cuenta que:

- *Copiar como imagen* copia la cadena cifrada en Base64 de la imagen seleccionada. Si la cadena cifrada en Base64 se pega en un documento en el que se puede mostrar como imagen (por ejemplo, en otra celda de la vista en forma de tabla), aparecerá como tal. De lo contrario al pegarla seguirá siendo solo una cadena.
- Para insertar texto en una celda copie el texto y péguelo en ella.

- Para importar desde un archivo, por ejemplo una imagen, use el comando **Editar | Insertar | Archivo externo codificado**[1284]. Este comando inserta una imagen como cadena cifrada en Base64 y la muestra como imagen en la vista Cuadrícula JSON. Estos son los formatos de imagen que se pueden importar: `PNG`, `JPEG`, `BMP`, `GIF`, `TIFF`.

Copiar XPath
Este comando copia en el portapapeles la expresión XPath 3.1 de ruta, empezando por la raíz del documento, que ubica el nodo seleccionado.

Copiar XPointer
Este comando copia en el portapapeles una expresión XPointer que ubica el nodo seleccionado. Para más detalles consulte **Copiar XPointer/Pointer JSON**[1283].

Filtro, Filtro de enfoque
El comando **Filtro** se puede activar y desactivar. Con él puede añadir un filtro al elemento que seleccione o desactivar un filtro que ya existe. Una vez que añada un filtro con este comando, recuerde que tiene que introducir una expresión de filtro o el filtro desaparecerá automáticamente. Una vez que se haya añadido un filtro puede desactivarlo volviendo a seleccionar este comando o haciendo clic en el filtro. Para eliminar el filtro haga clic en el comando *Eliminar todos los filtros* en la barra de herramientas.

El comando **Filtro de enfoque** (i) genera un filtro en el elemento raíz que contiene una expresión XPath para ubicar el nodo seleccionado y (ii) filtra los datos de la vista Cuadrícula para mostrar el nodo seleccionado (y sus descendientes) directamente bajo el elemento raíz. Esta opción es útil si quiere ver solamente el nodo seleccionado en la cuadrícula. Para quitar el filtro haga clic en el icono Filtro de la celda del elemento raíz.

Expandir todo
Este comando se habilita si se contraen el componente seleccionado o cualquiera de sus descendientes. El comando sirve para expandir el componente y todos sus descendientes.

Contraer elementos no seleccionados
Este comando contrae todos los componentes excepto el seleccionado y sus antecesores.

Orientación del texto
Este comando está disponible para los nodos que contienen texto y cambia el orden de lectura para que empece por la izquierda o por la derecha de la celda. Este comando es útil para idiomas como el árabe o el hebreo.

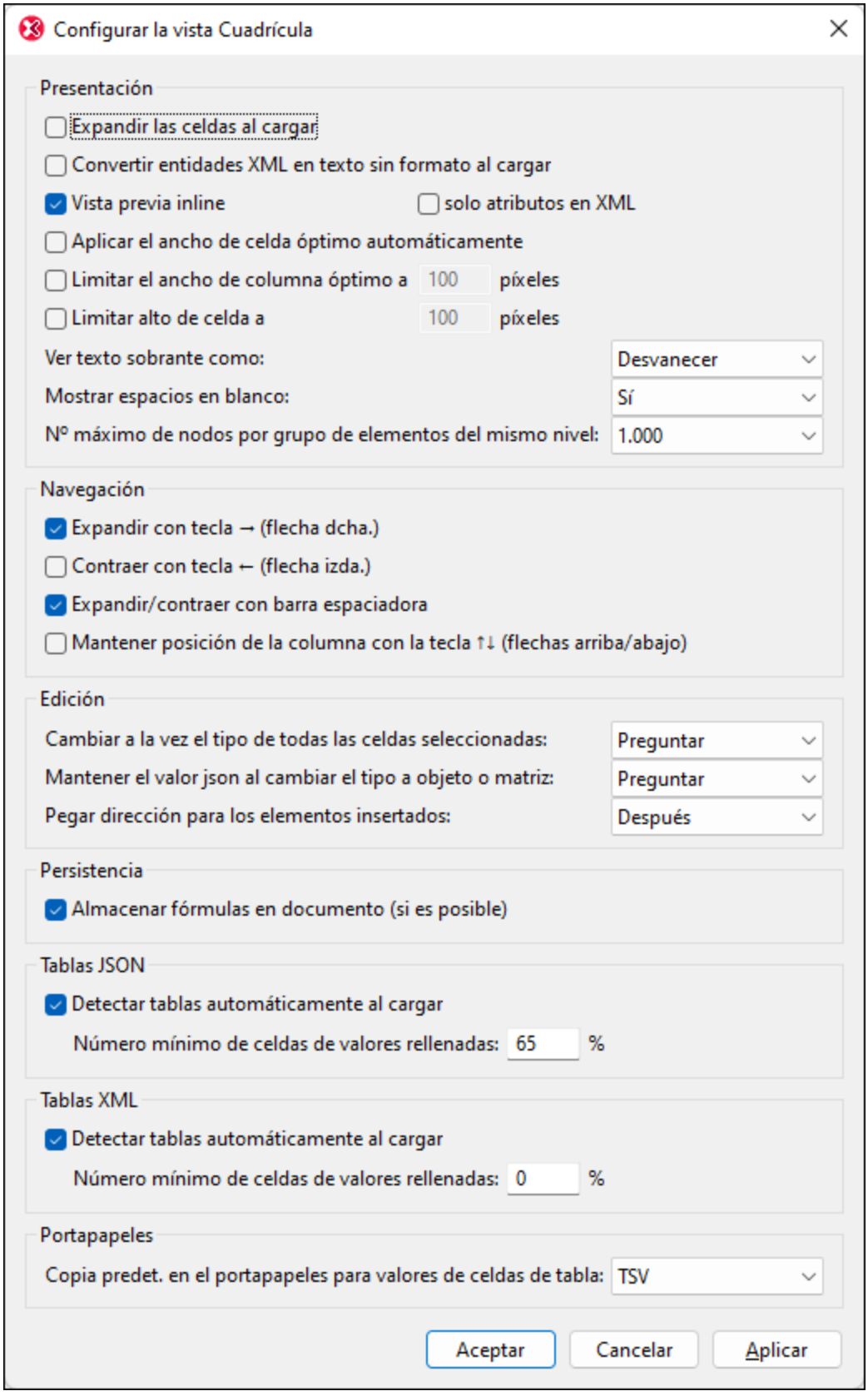
5.4.11 Configuración de la vista Cuadrícula

Puede configurar la vista Cuadrícula en el cuadro de diálogo Configurar la vista Cuadrícula (*imagen siguiente*), al que puede acceder con el icono Configuración de la barra de herramientas de la vista Cuadrícula[293].

	El icono de la configuración de la vista Cuadrícula de la barra de herramientas de la vista Cuadrícula[293]

A continuación explicamos las opciones de configuración de la Vista Cuadrícula. Tenga en cuenta que estas opciones afectan a la vista Cuadrícula de todos los documentos (XML, JSON, DTD).

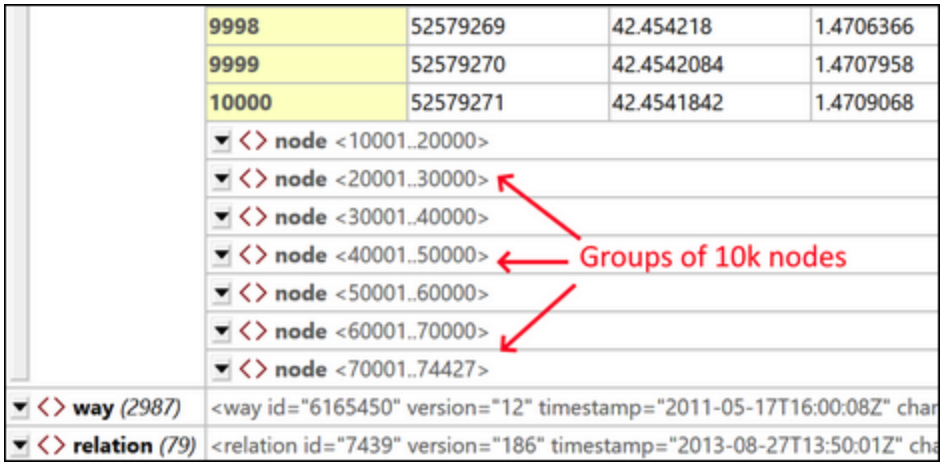
© 2016-2022 Altova GmbH


Configurar la vista Cuadrícula
Presentación
Expandir las celdas al cargar
Convertir entidades XML en texto sin formato al cargar
Vista previa inline
solo atributos en XML
Aplicar el ancho de celda óptimo automáticamente
Limitar el ancho de columna óptimo a 100 píxeles
Limitar alto de celda a 100 píxeles
Ver texto sobrante como: Desvanecer
Mostrar espacios en blanco: Sí
Nº máximo de nodos por grupo de elementos del mismo nivel: 1.000
Navegación
Expandir con tecla → (flecha dcha.)
Contraer con tecla ← (flecha izda.)
Expandir/contraer con barra espaciadora
Mantener posición de la columna con la tecla ↑↓ (flechas arriba/abajo)
Edición
Cambiar a la vez el tipo de todas las celdas seleccionadas: Preguntar
Mantener el valor json al cambiar el tipo a objeto o matriz: Preguntar
Pegar dirección para los elementos insertados: Después
Persistencia
Almacenar fórmulas en documento (si es posible)
Tablas JSON
Detectar tablas automáticamente al cargar
Número mínimo de celdas de valores rellenadas: 65 %
Tablas XML
Detectar tablas automáticamente al cargar
Número mínimo de celdas de valores rellenadas: 0 %
Portapapeles
Copia predet. en el portapapeles para valores de celdas de tabla: TSV
Aceptar
Cancelar
Aplicar

Presentación

Las opciones de la sección *Presentación* son bastante sencillas. No obstante, a continuación incluimos un par de aclaraciones.

- Si no se expanden todas las celdas al cargar, el nodo raíz y todos sus descendientes están contraídos. En este caso debe ir expandiendo los nodos conforme navega por el documento.
- Si se marca la opción *Convertir entidades XML en texto sin formato al cargar* las entidades XML se cargan en la vista Cuadrícula como texto sin formato de la entidad correspondiente; no se resuelven en sus respectivas representaciones de glifo.
- Si no se marca la opción *Vista previa inline*, en lugar de una vista previa de la celda solo aparece el número de índice del elemento de esa celda. Si se habilita esta opción, puede decidir si quiere que la vista previa contenga (i) el contenido y los atributos del elemento o (ii) solo los atributos. Si prefiere esta última opción, marque la casilla *solo atributos* en XML. Si quiere ver ambos, desmárquela. Recuerde que solamente se ve la primera parte del contenido inline de una celda; pase el cursor del ratón sobre la etiqueta de inicio para ver todo su contenido.
- Si activó el ancho de tabla óptimo podrá ver la cuadrícula entera. Para ello algunas celdas aplicarán el ajuste automático de línea.
- Si el texto es mayor que el tamaño de la celda puede elegir entre que el texto sobrante se desvanezca o sea reemplazado por puntos suspensivos.
- Puede mostrar u ocultar los espacios en blanco de las celdas. Los espacios aparecen como puntos a media altura y las tabulaciones como flechas. El final de línea se indica con un retorno de carro dentro de la celda.
- Los nodos del mismo nivel se pueden mostrar en grupos de 100, 1.000, 10.000 nodos (*imagen siguiente*). Esto es útil por dos motivos: (i) si quiere disponer de más espacio en la presentación de archivos que tienen un muchos nodos del mismo nivel; (ii) si solo quiere procesar un subconjunto de nodos del mismo nivel (porque ahorra tiempo), puede expandir los grupos de nodos del mismo nivel que quiere procesar y ejecutar la acción únicamente en esos nodos; por ejemplo si quiere usar la acción Buscar y reemplazar o si quiere convertir el tipo de un nodo (consulte *Edición*, más abajo).



Navegación

Ahora puede usar las flechas para navegar por la cuadrícula. A continuación explicamos las opciones que existen al usar estas teclas.

- *Expandir con la flecha derecha:* si un elemento de celda está contraído, la tecla *Flecha derecha* lo expande dentro de la celda. Si el elemento no está contraído, la tecla *Flecha derecha* pasa a la siguiente celda a la derecha (esto incluye elementos secundarios). Si no se activa esta opción, la tecla *Flecha derecha* se detiene en una celda contraída. La opción *Expandir con la flecha derecha* no

© 2016-2022 Altova GmbH

afecta a las celdas que están dentro de las tablas; en esas celdas la flecha derecha simplemente le llevará a la siguiente celda a la derecha.

- *Contraer con la flecha izquierda:* al moverse hacia la *izquierda* con esta tecla llega un momento en que tiene que moverse hacia arriba en la jerarquía del documento. Si se activa esta opción, lo elementos que se puedan contraer se contraen al pulsar la tecla *Flecha izquierda*; de lo contrario estos elementos no se contraen aunque la celda activa para al elemento superior. La opción *Expandir con la flecha izquierda* no afecta a las celdas que están dentro de las tablas; en esas celdas la flecha izquierda simplemente le llevará a la siguiente celda a la izquierda.
- *Expandir/contraer con la barra espaciadora:* la barra espaciadora sirve para expandir/contraer un elemento. Es decir, también puede usarla para navegar por la cuadrícula.
- *Mantener la posición de la columna con las teclas de flecha hacia arriba y abajo:* estas dos teclas suben y bajan por las celdas de la cuadrícula, incluidos los elementos secundarios y principales, que se encuentran en distintos niveles de la jerarquía y, por tanto, en distintas columnas. Si se selecciona esta opción puede saltarse los niveles que no estén en la columna activa. Esto puede ser útil, por ejemplo, si se encuentra en la columna del elemento **`subject/course/books/book/title`**. Al seleccionar esta opción puede usar las flechas arriba y abajo para navegar solamente por los títulos de los libros (sin entrar en las columnas book, books, course o subject y sin entrar en las columnas de elementos descendientes de **`Title`**.).

Edición

Las opciones de la sección *Presentación* son bastante sencillas. No obstante, a continuación incluimos un par de aclaraciones.

- Al cambiar el tipo de varias celdas seleccionadas puede escoger si llevar a cabo la acción: *Siempre*, *nunca* o *tras preguntar* (para que decida el usuario).
- Al cambiar un tipo JSON de atómico a objeto o matriz también puede escoger si la acción se lleva a cabo: (i) tras *Preguntar* (si el valor del tipo atómico se debe conservar como el valor de un par **`clave:valor`** secundario sin nombre), (ii) *Siempre* (se conserva el valor en un par **`clave:valor`** secundario sin nombre) o (iii) *Nunca* (nunca se debe conservar el valor).
- La opción *Pegar dirección* determina si la selección del portapapeles se pega encima o debajo de la celda seleccionada.

Persistencia

Los filtros y las fórmulas se pueden almacenar en los metadatos de documentos XML y JSON. Las fórmulas se almacenan como instrucciones de procesamiento en documentos XML y como comentarios en algunos tipos de documentos JSON. Estas opciones indican cómo guardar los filtros y las fórmulas. En el caso de las fórmulas, la terminología *cuando sea posible* hace referencia al hecho de que esas fórmulas no se pueden guardar como comentarios en todos los documentos JSON (solo en documentos JSON5 y JSONC).

Tablas JSON, tablas XML

Si se activa la opción que detecta tablas de la vista Cuadrícula JSON automáticamente al cargar, entonces puede seleccionar cuál es el porcentaje mínimo de celdas de tabla rellenadas para que una tabla se reconozca como tal. Si el número de celdas rellenadas no sobrepasa esta cantidad, la estructura aparece como una cuadrícula normal con los elementos extensibles uno bajo el otro en una lista.

Clipboard

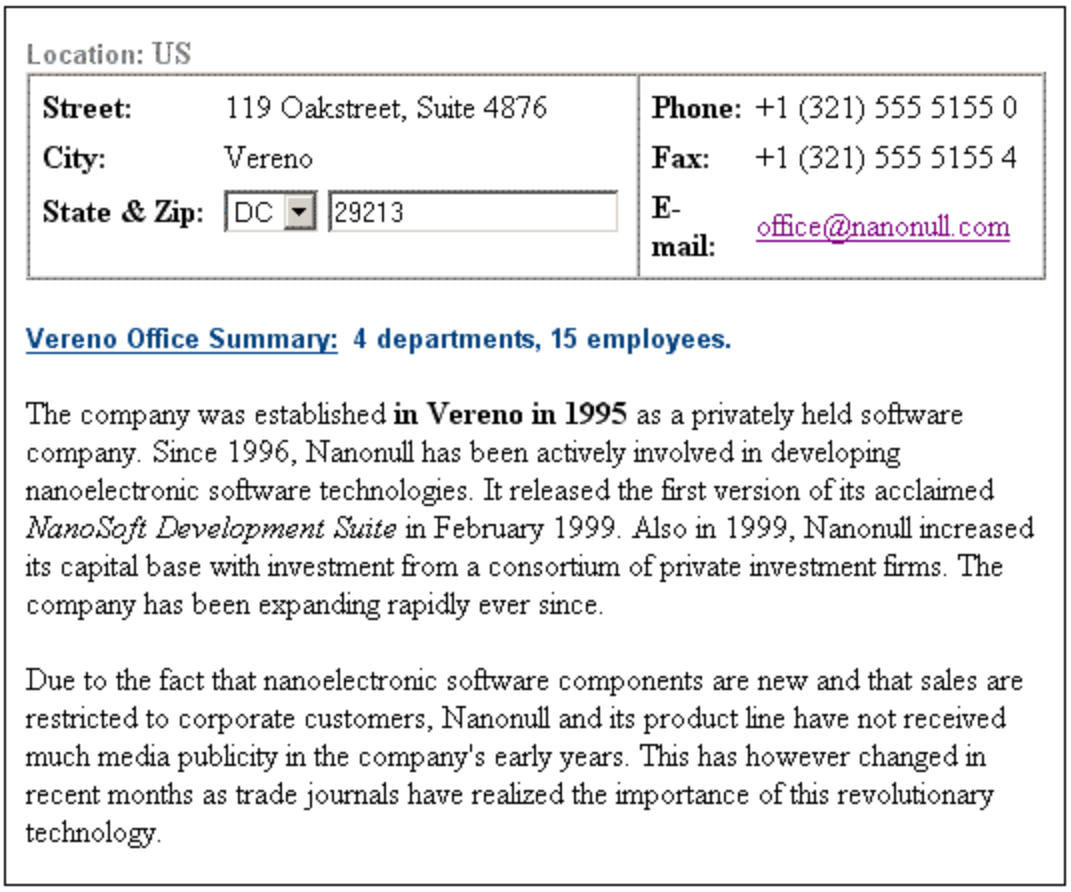
También puede escoger si el contenido del portapapeles se debe restaurar como valores separados por comas (TSV) o como XML/JSON (en función del tipo de documento). Esta es una opción muy útil: Si quiere pegar un

formulario de tabla desde el portapapeles en otro documento, esta opción permite elegir si la tabla copiada se guarda como TSV o con marcado. (Para ver la diferencia, intente pegar una tabla en un editor de texto después de copiarla en el portapapeles en los distintos formatos).

© 2016-2022 Altova GmbH

5.5 Datos XML en la vista Authentic

La vista Authentic permite al usuario editar un documento XML como si se tratase de un documento de texto (*ver imagen siguiente*). El marcado XML y el resto de texto que no sea contenido se puede ocultar. Esto puede ser práctico si quien edita el documento XML no está familiarizado con el lenguaje XML. Este tipo de usuario puede crear directamente un documento XML válido y concentrarse en el contenido del documento.



La vista Authentic de un documento se habilita si el documento XML tiene asignada una hoja de estilos SPS (StyleVision Power Stylesheet). La hoja de estilos SPS se basa en la misma fuente de esquema en la que se basa el documento XML y define la estructura del documento XML. La SPS también define el diseño y el formato del documento en la vista Authentic. Por ejemplo, en el documento de la imagen anterior se usan estas funciones de edición y formato de Authentic:

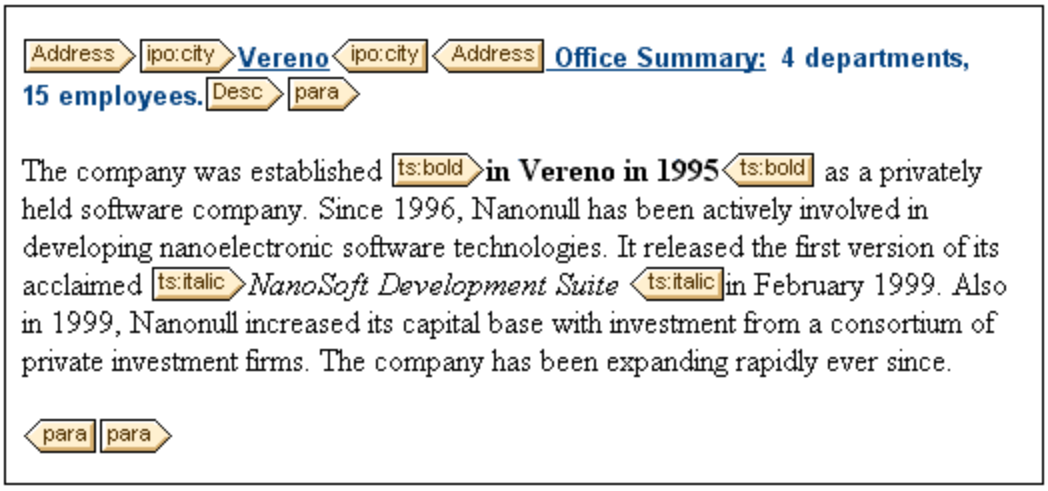
- Formato de párrafo y otros bloques
- Estructuras de tabla
- Formato de texto, como color y tipo de fuente
- Cuadros combinados (véase el campo State & Zip) que permiten al usuario seleccionar una opción de una lista de opciones. Estas opciones pueden ser, por ejemplo, enumeraciones del esquema
- Otra información calculada con los datos del documento (en el ejemplo anterior, el número de departamentos y empleados se calculó con datos presentes en el documento)

El objetivo de las hojas de estilos SPS es visualizar y editar documentos XML en la vista Authentic y generar documentos de salida estándar (en HTML, PDF, RTF o Word 2007) a partir de los documentos XML. Las SPS se crean con Altova StyleVision.

Editar la estructura del documento

Para añadir nodos válidos al documento, seleccione una posición y añada el nodo deseado desde los ayudantes de entrada (Elementos y Atributos) o con ayuda del menú contextual. Los nodos disponibles en cada posición son los nodos permitidos como secundarios o nodos del mismo nivel en dicha posición. Por ejemplo, si el cursor se sitúa dentro de un párrafo, puede anexar otro párrafo si así lo permite el esquema.

Mientras se edita la estructura de un documento XML en la vista Authentic, puede ser útil ver el marcado del documento. A tal efecto, puede activar y visualizar el marcado en forma de etiquetas (*ver imagen siguiente*). Para ello haga clic en el comando **Authentic | Mostrar marcado en tamaño grande** (o el icono correspondiente de la barra de herramientas).



Editar el contenido del documento

Para crear y editar contenido, introduzca el texto en los nodos del documento. Para añadir entidades y secciones CDATA, utilice el menú contextual (o el ayudante de entrada Entidades).

Más información sobre la vista Authentic

Para más información sobre la edición de documentos en la vista Authentic, consulte el apartado Vista Authentic 613.

© 2016-2022 Altova GmbH

5.6 Ayudantes de entrada para documentos XML

XMLSpy ofrece tres ayudantes de entrada a la hora de trabajar con documentos XML: **Elementos**, **Atributos** y **Entidades**. Cuando se añade un elemento desde el ayudante de entrada Elementos, el elemento se puede añadir con sus elementos secundarios obligatorios, con sus atributos obligatorios, con todos sus elementos secundarios o sin elementos ni atributos. Esto depende de las opciones de configuración elegidas en la sección *Edición* del cuadro de diálogo "Opciones". Cuando se añaden atributos, estos se añaden entre comillas.

Recuerde que el diseño de los ayudantes de entrada varía de una vista a otra, dependiendo de las funciones de cada vista.

Ayudante de entrada Elementos

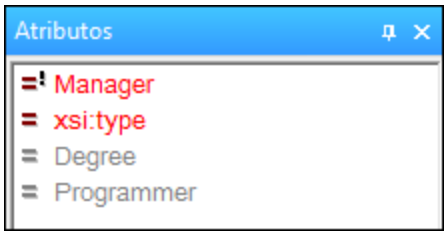
Es preciso señalar estos aspectos:

- *Vista Texto:* los elementos se insertan en el punto de inserción del cursor. Los elementos que no se utilizaron aún aparecen en rojo y los ya utilizados aparecen en gris. Los elementos obligatorios se señalan con un signo de exclamación (**!**) situado a la izquierda del nombre del elemento.
- *Vista Authentic*: los elementos se pueden insertar antes del elemento seleccionado, después o dentro del elemento. Además, el ayudante de entrada incluye una representación del documento en forma de árbol que muestra la posición del elemento seleccionado en la estructura del documento. Para más información, consulte el apartado Vista Authentic.

Ayudante de entrada Atributos

Es preciso señalar estos aspectos:

- *Vista Texto:* si el cursor se sitúa dentro de la etiqueta inicial de un elemento y después de un espacio, aparece una lista emergente con los atributos declarados para ese elemento. Los atributos que no se utilizaron aún aparecen en rojo y los ya utilizados en gris. Los atributos obligatorios se señalan con un signo de exclamación (**!**) situado a la izquierda del nombre del atributo.

  Para insertar un atributo, haga doble clic en el atributo deseado. El atributo se inserta en el punto de inserción del cursor junto con un signo = y las comillas que delimitan el valor del atributo. El cursor se sitúa automáticamente entre las comillas para que pueda escribir el valor del atributo inmediatamente.
- *Vista Authentic*: si selecciona un elemento, se muestran los atributos que se declararon para dicho elemento. Escriba el valor del atributo en el ayudante de entrada.

Ayudante de entrada Entidades

El ayudante de entrada Entidades muestra todas las entidades analizadas o sin analizar declaradas *inline* (es decir, dentro del documento XML) o en una DTD externa. Para insertar una entidad, ya sea en la vista Texto, Cuadrícula o Authentic, haga doble clic en la entidad y esta se inserta en el punto de inserción del cursor. En

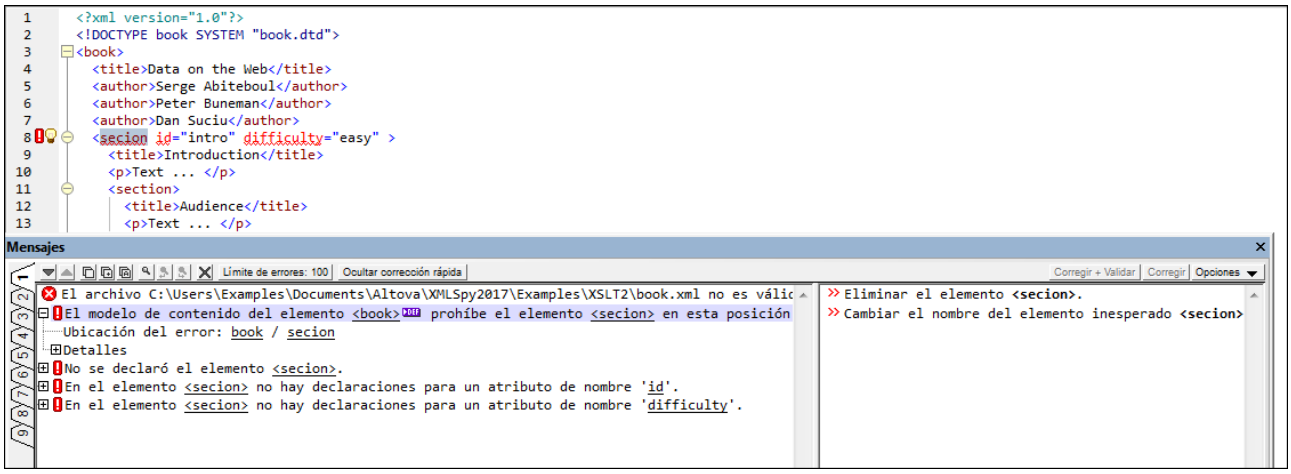
la vista Cuadrícula el ayudante de entrada Entidades tiene tres pestañas: Anexar, Insertar y Agregar secundario.

Tenga en cuenta que si añade una entidad interna, será necesario guardar, cerrar y abrir el documento de nuevo para que la entidad aparezca en el ayudante de entrada Entidades.

© 2016-2022 Altova GmbH

5.7 Validar un documento XML

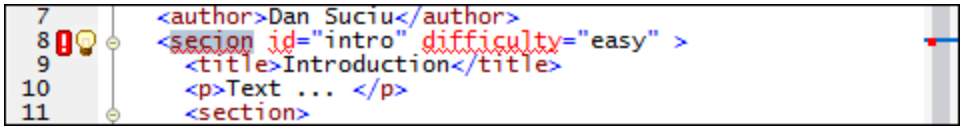
El comando **XML | Validar (F8)** valida documentos XML con el esquema asociado (DTD, XML u otro). Si un documento es válido, en la ventana Mensajes aparece un mensaje que lo indica. De lo contrario, se emiten mensajes explicando las causas del error en el panel izquierdo (*imagen siguiente*). Si selecciona un motivo de error en el panel izquierdo, en el panel derecho verá sugerencias para solucionar el problema. Las sugerencias de corrección rápida están basadas en datos disponibles en el esquema asociado. Hay dos maneras de aplicar correcciones rápidas a un documento: (i) haciendo doble clic en la sugerencia o (ii) seleccionando la sugerencia y haciendo clic en la opción **Corregir** o **Corregir + Validar** (*imagen siguiente*).



Indicadores de error y correcciones rápidas

Indicadores de error

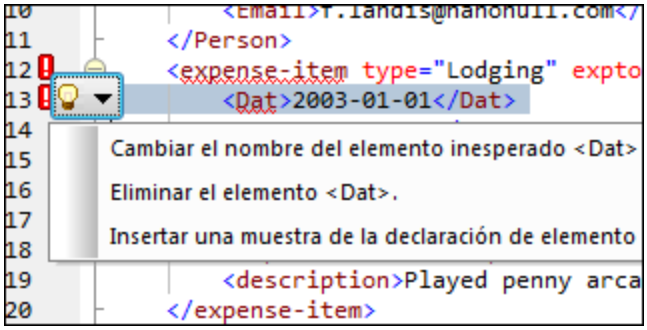
Además, en la vista Texto aparecen dos indicadores más que sirven para identificar errores de validación (*imagen siguiente*): (i) un signo de exclamación en rojo que aparece en el margen de numeración de líneas y (ii) un cuadrado rojo en la barra de desplazamiento (a la derecha de la ventana).



Observe el marcador cuadrado rojo que aparece en el borde izquierdo de la barra de desplazamiento (situada en el lado derecho de la ventana, *imagen anterior*) y recuerde que la barra de desplazamiento puede mostrar otros dos tipos de marcadores cuadrados: (i) marcadores cuadrados marrones en el borde izquierdo de la barra de desplazamiento para indicar instancias del texto resaltado (*véase* *Navegar por documentos* 154) y (ii) marcadores cuadrados marrones en el borde derecho de la barra de desplazamiento para indicar resultados de la búsqueda de texto (*véase* *Búsqueda y reemplazo* 157).

Soluciones inteligentes

Si la aplicación dispone de una solución para un error, aparece un icono en forma de bombilla en la línea que genera el error (*imagen siguiente*). Si pasa el cursor por encima de este icono, aparecen todas las soluciones para el error en un menú contextual (*imagen siguiente*). Seleccione una solución para corregir el error inmediatamente.



Tenga en cuenta que:

- XMLSpy muestra indicadores de errores de validación y sugerencias de corrección rápida para los tipos de documento compatibles con la función de validación (p. ej. documentos JSON.
- los indicadores de errores de validación y las soluciones rápidas que se describen más arriba solo se actualizan si se ejecuta el comando **XML | Validar (F8)**, pero no se actualizan en segundo plano. Por tanto, una vez que corrija un error debe ejecutar el comando **Validar (F8)** de nuevo para asegurarse de que ese error se ha corregido.

Para más información sobre la función de validación de documentos XML consulte la descripción del comando **Validar**[1337].

Validar al editar

Si activa el modo *Validar al editar* puede validar un documento y comprobar si su formato es el correcto mientras lo modifica en la vista Texto (y en la vista Cuadrícula JSON). Para validar un documento XML (además de comprobar su formato) debe asignarle un esquema DTD o XML (en el caso de los documentos JSON debe añadir un esquema JSON). Si hay errores, aparecen marcados en el margen izquierdo con indicadores de errores (imagen anterior) y en las líneas que contienen esos errores.

El modo *Validar al editar* se puede activar y desactivar (i) con el comando de menú **XML | Validar al editar**[1344], (ii) el botón **Validar al editar** de la barra de herramientas o (iii) con la opción *Al editarlos* de las las opciones de validación del cuadro de diálogo "Opciones"[1584].

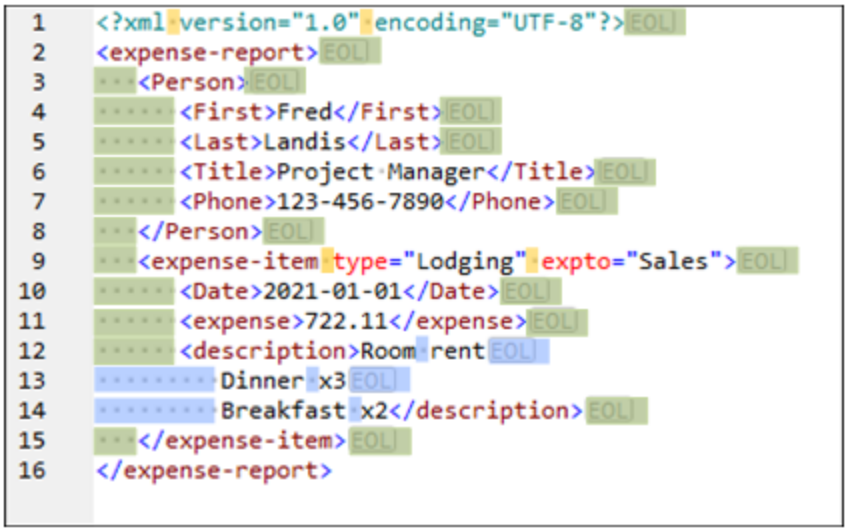
© 2016-2022 Altova GmbH

5.8 Espacios en blanco

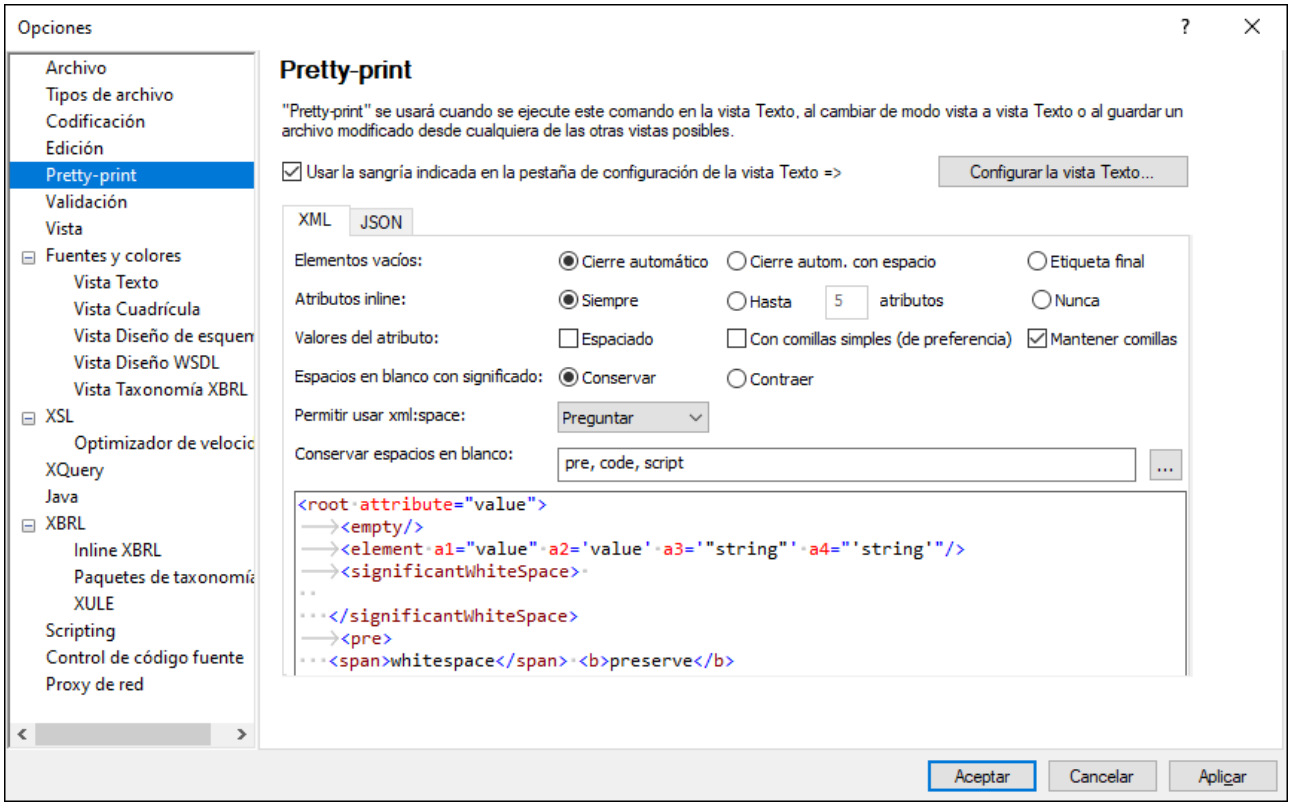
Los caracteres de espacio en blanco son el espacio, la tabulación, el retorno de carro y el salto de línea. Puede mostrar u ocultar estos caracteres en el cuadro de diálogo Configurar la vista Texto (**Vista | Configurar la vista Texto**[1489]).

En un documento XML los espacios en blanco existen por varias razones:

- Por cuestiones de sintaxis XML, por lo general para delimitar componentes XML. Este tipo de espacios en blanco está marcado en amarillo en la imagen siguiente.
- Espacios en blanco con significado, que pueden darse dentro de un elemento, atributo o instrucción de procesamiento y no se deben ignorar. Estos aparecen en azul en la imagen siguiente.
- Espacios en blanco sin significado, que aparecen entre dos elementos que no tienen nodos de texto relacionados. Estos espacios en blanco solo ocurren en elementos que no son de contenido mixto. Se suelen usar con fines de dar formato y no tienen significado. Este tipo de espacios en blanco está marcado en verde en la imagen siguiente.



En XMLSpy se añaden espacios en blanco si aplica pretty-print a un documento (**Edición | Pretty-print**[1288]). Al aplicar pretty-print se aplican espacios en blanco sin significado para dar formato al documento y mostrar así su estructura claramente. El formato pretty-print también puede contraer espacios en blanco en función de las opciones (*imagen siguiente*).



Estas son las opciones de pretty-print[1591] que afectan a los espacios en blanco:

- Los *espacios en blanco con significado* se pueden conservar o contraer. Si elige *Contraer* sigue pudiendo mostrarlos en elementos concretos añadiéndolos a la lista *Conservar espacios en blanco*.
- La opción *Conservar espacios en blanco* permite crear una lista de elementos que conservan los espacios en blanco (tengan o no significado).
- Si existen espacios en blanco con significado en un elemento vacío y ha elegido *Contraer* para *espacios en blanco con significado*, esos espacios en blanco se eliminan. La configuración de la opción *Elementos vacíos* determina cómo aparece el elemento vacío aparece una vez se han eliminado los espacios en blanco.

Nota: al cambiar de la vista Texto a la vista Cuadrícula y viceversa se conservan los cambios resultantes del formato pretty-print. En caso de que se cambie a las demás vistas, esos cambios dejan de aplicarse.

© 2016-2022 Altova GmbH

5.9 Insertar fragmentos XML

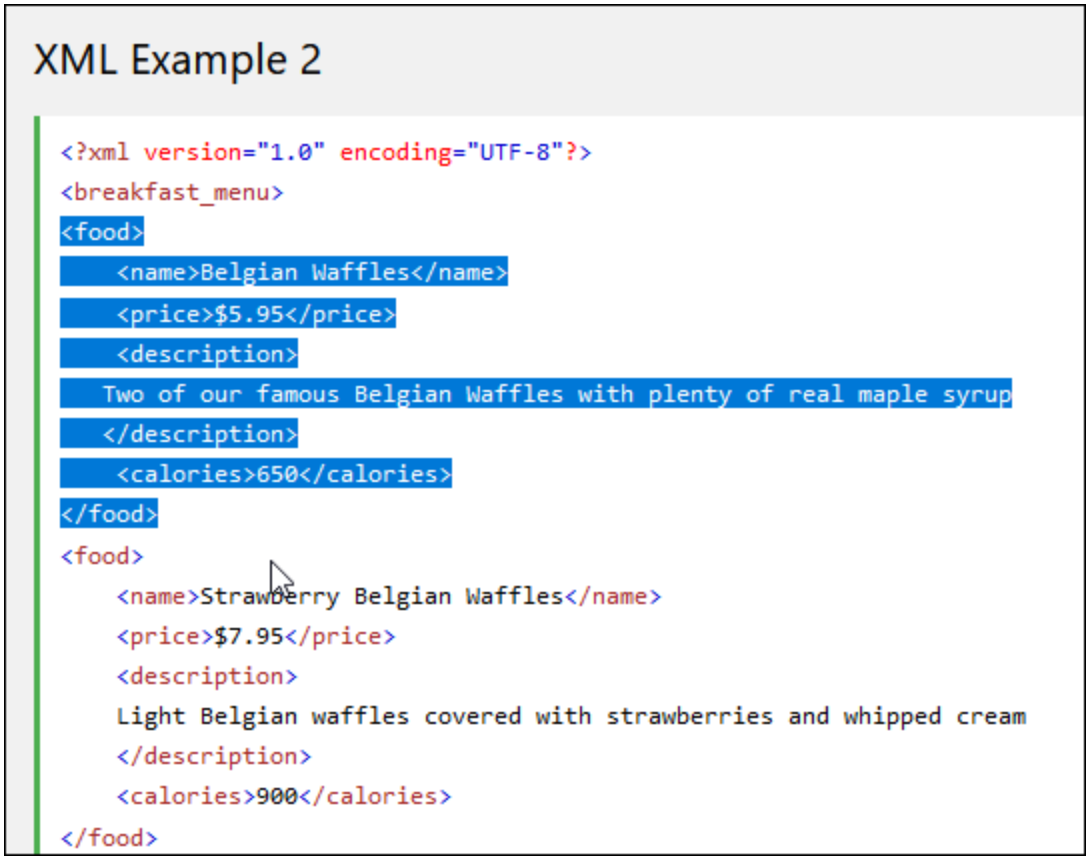
Puede insertar rápidamente fragmentos XML de otras aplicaciones y sitios web. Hay dos maneras de insertar estos fragmentos:

- Con la función de arrastrar y colocar de la vista Texto o la vista Cuadrícula. Si arrastra y coloca en la vista Cuadrícula, la información inteligente de la capa de arrastre[306] le ayudará a saber dónde colocar el fragmento.
- Con la función de copiar y pegar de la vista Texto o la vista Cuadrícula.

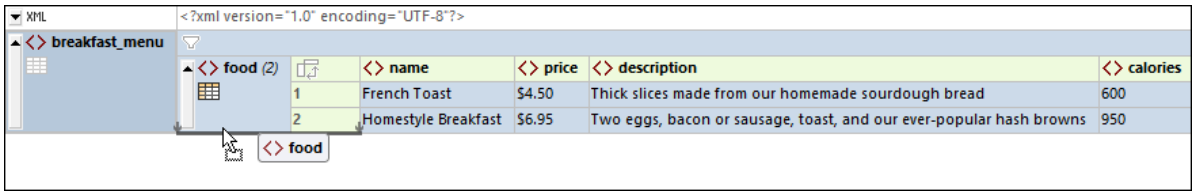
Ejemplo

En este ejemplo verá cómo añadir rápidamente un fragmento y cómo saber cuál es la ubicación correcta en un documento XML.

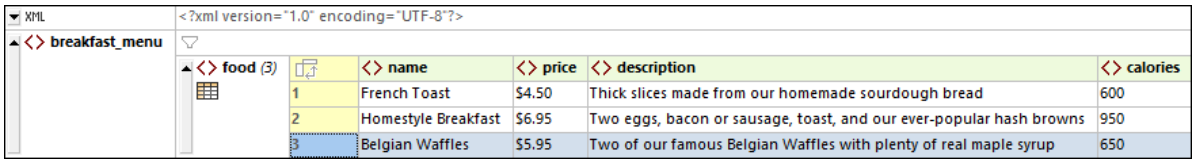
1. Seleccione el fragmento resaltado que ve a continuación (que proviene de un tutorial XML de w3schools). Es un elemento llamado **food** que contiene varios elementos secundarios.



2. En la imagen siguiente puede ver la vista Cuadrícula de un documentos XML donde aparecen dos elementos **food** como filas de tabla. Cuando se arrastra el fragmento del sitio web a la tabla **food** aparece una capa de arrastre[689] que contiene la información de que el fragmento XML se colocará como un elemento de la tabla **food** como su última fila.



3. Al soltar el botón del ratón el fragmento se coloca exactamente donde quería, como último elemento secundario tipo **`food`** de **`breakfast_menu`** (*imagen siguiente*).



© 2016-2022 Altova GmbH

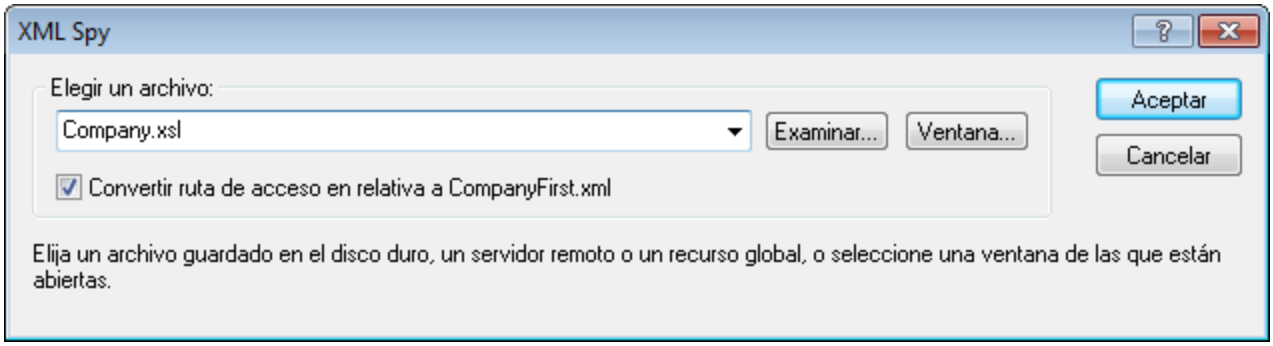
5.10 Procesamiento con XSLT y XQuery

Los documentos XML se pueden procesar con documentos XSLT o XQuery para producir documentos de salida. XMLSpy incluye procesadores XSLT 1.0, XSLT 2.0, XSLT 3.0, XQuery 1.0 y XQuery 3.0 integrados. A continuación describimos las funciones de la IGU relacionadas con el procesamiento con XSLT y XQuery:

- Asignar hojas de estilos XSLT 341
- Ir al archivo XSL 341
- Parámetros de XSL y variables de XQuery 341
- Transformaciones XSLT 342
- Ejecutar documentos XQuery 342
- Automatizar tareas XML con AltovaXML 342

Asignar hojas de estilos XSLT

Puede asignar una hoja de estilos XSLT a un documento XML con el comando **XSL/XQuery | Asignar XSL** (aparece el cuadro de diálogo de la imagen siguiente, donde puede buscar el archivo). La asignación se introduce en el documento XML en forma de instrucción de procesamiento (IP), cuyo XSLT de destino estándar viene definido por el W3C: `xml-stylesheet`. Esta asignación se utiliza al invocarse la transformación XSLT (comando **XSL/XQuery | Transformación XSL**).



Además puede asignar una hoja de estilos XSL-FO con el comando **XSL/XQuery | Asignar XSL:FO** (aparece el cuadro de diálogo de la imagen siguiente, donde puede buscar el archivo). La asignación se introduce en el documento XML en forma de instrucción de procesamiento (IP), cuyo destino viene definido por Altova: `altova_xslfo`. Esta asignación se utiliza al invocarse la transformación XSL-FO (comando **XSL/XQuery | Transformación XSL-FO**).

Si ya existe una asignación que utiliza una de estas instrucciones de procesamiento, se le preguntará si desea sobrescribir la asignación actual.

Ir al archivo XSLT

El comando **XSL/XQuery | Ir al archivo XSL** abre el archivo XSLT que se asignó al documento XML.

Parámetros de XSL y variables de XQuery

Los parámetros de XSL y las variables de XQuery se pueden definir, editar y eliminar en el cuadro de diálogo que aparece al hacer clic en el comando **XSL/XQuery | Parámetros de XSLT / Variables de XQuery**. Los

valores de los parámetros y de las variables definidos aquí se utilizan para todas las transformaciones XSLT y todas las ejecuciones de XQuery de XMLSpy. No obstante, estos valores no se pasarán a motores externos como MSXML. Para más información sobre cómo utilizar esta función, consulte la sección pertinente de la sección *Comandos de menú*.

Transformaciones XSLT

Puede utilizar dos tipos de transformaciones XSLT:

- Transformación XSLT estándar (comando **XSL/XQuery | Transformación XSL**): el resultado de la transformación se presenta en una ventana nueva o, si así lo indicase la hoja de estilos, se guarda en un archivo. El motor utilizado para la transformación se especifica en la sección *XSL* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).
- Transformación XSL:FO (comando **XSL/XQuery | Transformación XSL:FO**): el documento XML se transforma en un documento PDF en solo dos pasos. En primer lugar, el documento XML se transforma en un documento FO con el procesador XSLT especificado en la sección *XSL* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**). Tenga en cuenta que también puede seleccionar (en la parte inferior de la pestaña) el motor XSLT que viene con algunos procesadores FO, como FOP, por ejemplo. En segundo lugar, el documento FO se procesa con el procesador FO especificado en la sección *XSL* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**) para producir el documento PDF de salida.

Nota: los documentos FO (que son un tipo especial de documento XML) se pueden transformar en PDF si hace clic en el comando **Transformación XSL:FO**. Si el documento de origen es un documento FO, el segundo paso del proceso descrito se ejecuta directamente.

Ejecutar documentos XQuery

Para ejecutar un documento XQuery en el documento XML activo, haga clic en el comando **XSL/XQuery | Ejecutar XQuery**. Se le pedirá que indique el documento XQuery que se debe usar y el documento resultante aparece en una ventana nueva de la interfaz.

Automatizar tareas XML con RaptorXML

Altova RaptorXML es una aplicación que ofrece funciones de validación XML, transformaciones XSLT y ejecuciones de XQuery. Puede utilizarse desde la línea de comandos, a través de una interfaz COM, en programas Java y en aplicaciones .NET. Por tanto, esta aplicación permite automatizar tareas, como transformaciones XSLT. Por ejemplo, puede crear un archivo por lotes que llame a RaptorXML para transformar un conjunto de documentos. Para más información consulte la documentación de RaptorXML.

© 2016-2022 Altova GmbH

5.11 Fuentes en documentos PDF

Cómo utilizan las fuentes el formateador y el visor de documentos PDF

El formateador (FOP por ejemplo) crea el documento PDF y el visor de documentos PDF (Adobe PDF Reader por lo general) lee el documento.

Para poder diseñar el documento PDF, el formateador necesita tener información sobre las fuentes utilizadas en el documento, en especial sobre el ancho de los glifos utilizados. Necesita esta información para calcular la longitud de las líneas, la división de palabras al final de línea, la justificación, etc. Esta información se denomina *medidas de fuentes* y se almacena en cada fuente. Algunos formateadores pueden leer estas medidas directamente de la carpeta de fuentes del sistema. Otros (como los formateadores FOP) necesitan que las medidas estén en un formato especial que pueda comprender. Cuando las medidas de una fuente están disponibles, el formateador puede diseñar el PDF correctamente. Por ello, es necesario que estén disponibles los archivos de medidas de fuentes para todas las fuentes utilizadas en el documento.

El formateador puede hacer referencia a la fuente o incrustarla en el archivo PDF. Si se hace referencia a la fuente, el visor de documentos PDF (Adobe Reader, por ejemplo) primero buscará la fuente en su carpeta de fuentes (que contiene las fuentes Base 14) y después en la carpeta de fuentes del sistema. Si la fuente está disponible, se utilizará en el PDF. Si no está disponible, el visor utilizará otra fuente de la carpeta de fuentes o generará un error. Puesto que la fuente alternativa utilizada puede tener medidas distintas, se podrían generar errores de visualización.

Si el formateador incrusta una fuente en el archivo PDF, el visor de PDF usará la fuente incrustada. El formateador puede incrustar el juego de caracteres completo de una fuente o sólo un subjuego que contenga los glifos utilizados en el documento. Este factor repercute en el tamaño del archivo PDF y también tiene repercusiones en materia de copyright con respecto al uso de la fuente (ver la nota que aparece al final de este apartado). Si lo desea, puede configurar estas opciones del formateador.

XMLSpy y fuentes en documentos PDF

En XMLSpy los documentos PDF se generan a partir de documentos XSL:FO (en adelante, documento FO). Para ello se procesa el documento FO con un procesador FO externo como FOP. (Puede especificar la ubicación del procesador FO en el cuadro de diálogo "Opciones". De ese modo el procesamiento FO se puede iniciar desde la IGU de XMLSpy.)

El documento FO propiamente dicho se genera al procesar un documento XML con una hoja de estilos XSLT. (Para ello puede usar tanto el motor XSLT de Altova, que está integrado en XMLSpy, como un motor XSLT externo.)

El formato del documento PDF, incluidas las propiedades de fuente para todo el texto, se especifica en el documento FO. Si el formateador que utiliza puede leer las medidas de las fuentes necesarias directamente desde la fuente, entonces lo único que tiene que hacer es configurar el formateador para que pueda tener acceso a la fuente. Sin embargo, si utiliza FOP como formateador, será necesario proporcionar al formateador los archivos de medidas de fuentes para las fuentes que no sean Base 14.

Poner las fuentes a disposición del formateador

La mayoría de los formateadores (incluido FOP) ya tienen acceso a fuentes Base 14. Es importante conocer los nombres con los que el formateador reconoce estas fuentes para poder ponerlas a su disposición correctamente. No obstante, puede poner muchas más fuentes a disposición del formateador. A continuación describimos cómo poner más fuentes a disposición del formateador FOP.

Procedimiento general para configurar la compatibilidad de FOP con fuentes adicionales

Para poner más fuentes a disposición de FOP, siga estos sencillos pasos:

1. Genere un archivo de medidas de fuentes para la fuente que desee añadir a partir de los archivos de fuentes PostScript o TrueType. FOP ofrece las utilidades PFM Reader y TTF Reader para convertir fuentes PostScript y TrueType respectivamente en archivos de medidas de fuentes XML. Para más información sobre este proceso, consulte la página web FOP: Fuentes.
2. Configure el archivo de configuración de FOP para que utilice los archivos de medidas de fuentes pertinentes. Para ello, escriba la información sobre los archivos de fuentes en un archivo de configuración de FOP. Para más información, consulte la página web FOP: Fuentes.
3. En el archivo `fop.bat` cambie la última línea:

   ```
   "%JAVACMD%" […] org.apache.fop.cli.Main %FOP_CMD_LINE_ARGS%
   ```

   para que incluya la ubicación del archivo de configuración:

   ```
   "%JAVACMD%" […] org.apache.fop.cli.Main %FOP_CMD_LINE_ARGS% -c conf\fop.xconf
   ```

Después de registrar los archivos de medidas con FOP (en un archivo de configuración FOP) y después de configurar el ejecutable FOP para que lea el archivo de configuración, las fuentes adicionales estarán disponibles para la creación de documentos PDF.

Configurar el archivo de configuración de FOP

El archivo de configuración de FOP se llama `fop.xconf` y se encuentra en la carpeta `conf` de la carpeta de instalación de FOP. Este archivo, que es un documento XML, se debe editar para que FOP lea correctamente los archivos de medidas de fuentes. Por cada fuente que deba ser representada por FOP, añada un elemento `font` en la posición indicada por el marcador de posición de elemento `font-`.

```
<font metrics-url="arial.xml" kerning="yes" embed-url="arial.ttf">
          <font-triplet name="Arial" style="normal" weight="normal"/>
          <font-triplet name="ArialMT" style="normal" weight="normal"/>
</font>
```

En el ejemplo anterior

`arial.xml`	es la URL del archivo de medidas. Es mejor usar una ruta de acceso absoluta.
`arial.ttf`	es el nombre del archivo TTF (que suele encontrarse en la carpeta `%DIRWIN%\Fuentes`).
`Arial`	indica que las medidas y los archivos TTF se usarán si la familia de fuentes se define como `Arial`.
`style="normal"`	indica que las medidas y los archivos TTF se usarán si el estilo de fuente se define como `normal` (en lugar de, por ejemplo, `italic`).
`weight="normal"`	indica que las medidas y los archivos TTF se usarán si el espesor de la fuente se define como `normal` (en lugar de, por ejemplo, `bold`).

© 2016-2022 Altova GmbH

Nota sobre el copyright de las fuentes: el uso de las fuentes está sujeto a las leyes sobre los derechos de autor y las condiciones de uso dependen de cada fuente. Antes de incrustar una fuente, sobre todo si desea incrustar la fuente entera, asegúrese de que la licencia de la fuente lo permite.

Juegos de caracteres

Tenga en cuenta que los juegos de caracteres de las fuentes varían dependiendo de la fuente. Las fuentes Base 14 abarcan los caracteres ISO-8859-1 más los glifos de las fuentes Symbol y Zapf Dingbats. Si su documento contiene un carácter que no está cubierto por las fuentes Base 14, tendrá que usar una fuente que contenga este carácter en su juego de caracteres. Algunas fuentes, como Arial Unicode, ofrecen los caracteres que abarca Unicode.

5.12 Gráficos

Cuando un documento XML está abierto en la vista Texto o Cuadrícula, en la ventana Gráficos (una de las ventanas de resultados[132]) se puede generar un gráfico (circular, de barras, etc.) que represente los datos seleccionados en el documento XML. El gráfico se puede exportar como archivo de imagen o se puede copiar como fragmento XSLT o XQuery en el portapapeles. La función de gráficos es muy práctica a la hora de representar gráficamente los datos numéricos seleccionados en un documento XML.

Estos son los tipos de gráficos que se pueden generar:

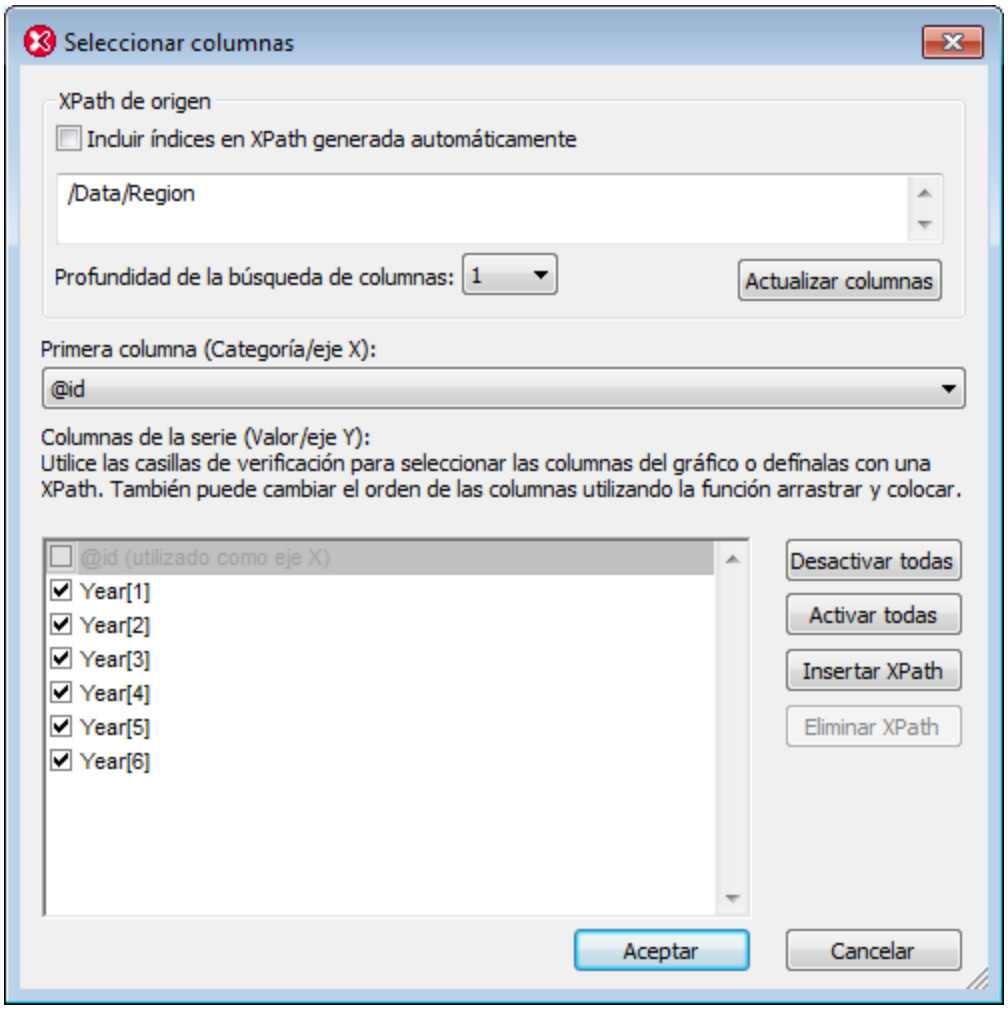
- Gráficos circulares (2D y 3D)
- Gráficos de barras simples (2D y 3D)
- Gráficos de barras agrupadas (2D y 3D)
- Gráficos de barras apiladas
- Gráficos de líneas de categorías
- Gráficos de líneas de valor
- Gráficos de áreas y áreas apiladas
- Gráficos de velas
- Medidores (radiales y lineales)
- Gráficos multicapa

Resumen: de la creación de gráficos a su exportación

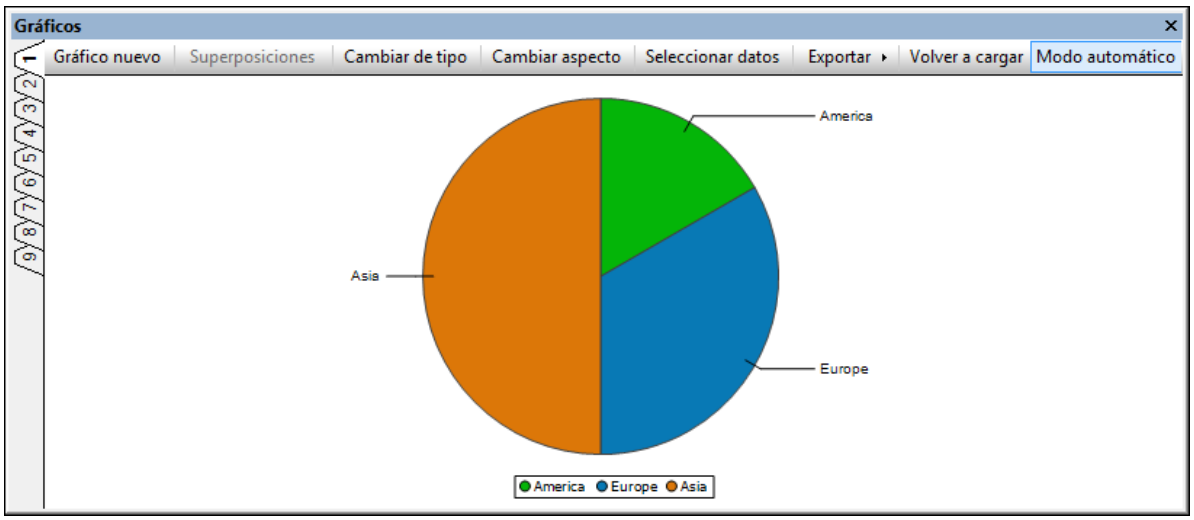
A continuación se describen los pasos generales para crear un gráfico. Para una descripción más detallada, consulte los apartados de esta sección.

1. En la vista Texto o Cuadrícula, seleccione el nodo que desea usar como nodo de contexto para la selección de datos. También puede seleccionar un rango de nodos. Las repercusiones de cada método de selección se explican en el apartado XPath de origen[353].
2. Haga clic con el botón derecho en el nodo o nodos seleccionados y seleccione el comando **Gráfico nuevo** del menú contextual que aparece. Otra opción es hacer clic en el botón **Gráfico nuevo** de la ventana Gráficos. A continuación aparece el cuadro de diálogo "Seleccionar columnas" (*imagen siguiente*), donde puede seleccionar los datos del eje X[357] y del eje Y[362] y modificar la XPath de origen[353].

© 2016-2022 Altova GmbH



3. Al hacer clic en el botón **Aceptar** se crea el gráfico en la ventana Gráficos (*imagen siguiente*).



4. Posteriormente se puede editar la selección de datos del gráfico y otras opciones de configuración. No solo se puede editar la selección de columnas y la XPath de origen. También puede modificar el tipo del gráfico y su aspecto Para editar la selección de datos para los ejes del gráfico haga clic en el botón **Seleccionar datos**. Para modificar el tipo del gráfico y su aspecto, haga clic en el botón **Cambiar de tipo** y **Cambiar aspecto** respectivamente.
5. El gráfico se puede exportar como archivo de imagen o se puede copiar como fragmento XSLT o XQuery en el portapapeles.

Otras características

La ventana Gráficos presenta otras prácticas características que facilitan el uso de la función de creación de gráficos:

- ***Varias pestañas:*** si desea crear un gráfico nuevo sin borrar el gráfico actual, puede crear el gráfico nuevo en una de las nueve pestañas de la ventana (*ver imagen anterior*). Tenga en cuenta que, incluso si se cierra el documento, los gráficos generados a partir de ese documento seguirán abiertos en las correspondientes pestañas de la ventana Gráficos.
- ***Recarga automática:*** si se activa el botón **Modo automático** (*ver imagen anterior*), el gráfico se vuelve a cargar automáticamente cada vez que se modifiquen los datos del documento XML. Si el botón no está activado, deberá actualizar el gráfico de forma manual con el botón **Volver a cargar**.

Archivo de ejemplo

En esta sección y en sus diferentes apartados usaremos como ejemplo el archivo XML `YearlySales.xml` para explicar cómo se trabaja con gráficos en XMLSpy Este archivo se encuentra en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Tutorial`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <Region id="Americas">
              <Year id="2005">30000</Year>
              <Year id="2006">90000</Year>
              <Year id="2007">120000</Year>
              <Year id="2008">180000</Year>
              <Year id="2009">140000</Year>
              <Year id="2010">100000</Year>
       </Region>
       <Region id="Europe">
              <Year id="2005">50000</Year>
              <Year id="2006">60000</Year>
              <Year id="2007">80000</Year>
              <Year id="2008">100000</Year>
              <Year id="2009">95000</Year>
              <Year id="2010">80000</Year>
       </Region>
       <Region id="Asia">
              <Year id="2005">10000</Year>
              <Year id="2006">25000</Year>
              <Year id="2007">70000</Year>
              <Year id="2008">110000</Year>
              <Year id="2009">125000</Year>
              <Year id="2010">150000</Year>
       </Region>
```

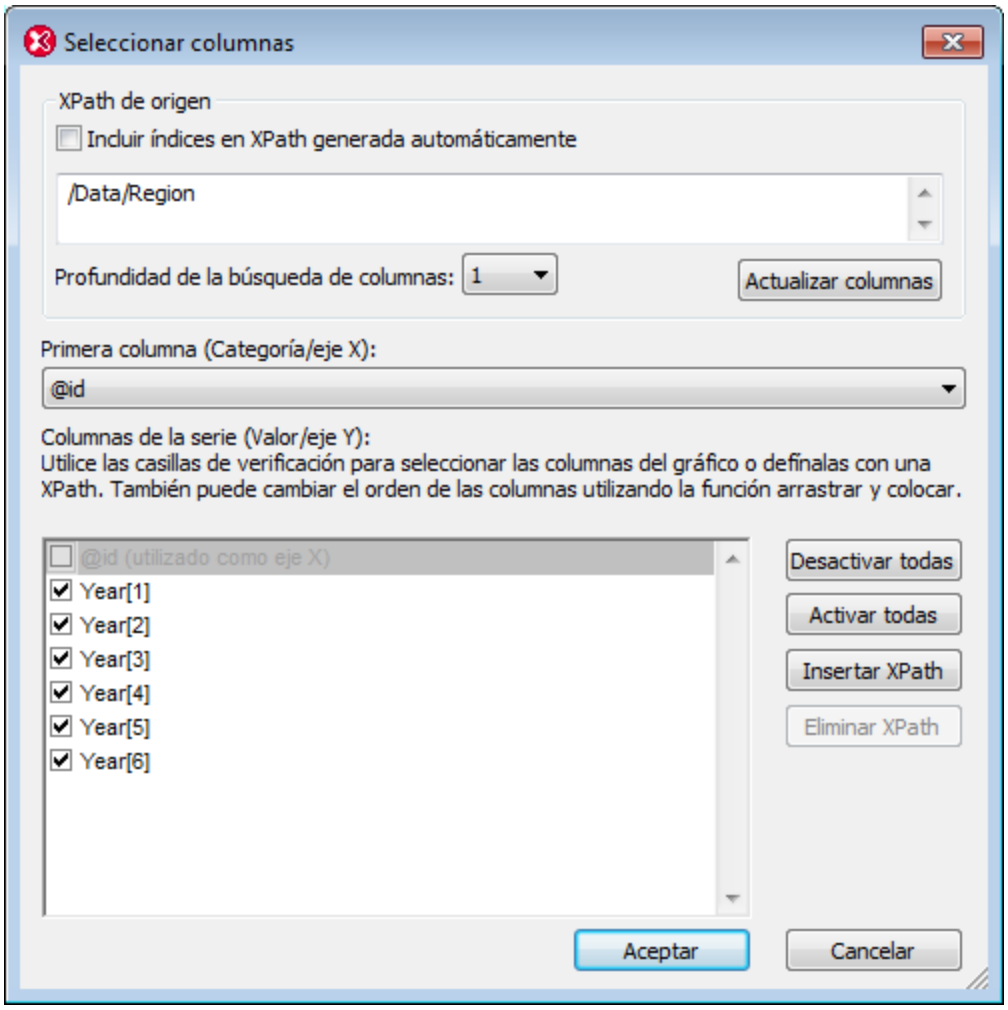

© 2016-2022 Altova GmbH

```
</Data>
```

5.12.1 Creación de gráficos

Al hacer clic en el botón **Gráfico nuevo** se abre el cuadro de diálogo "Seleccionar columnas" (*ver imagen siguiente*), donde puede especificar tres parámetros fundamentales de selección de datos para el gráfico. Estos parámetros (descritos a continuación) se utilizan para construir la tabla de datos del gráfico.

- *XPath de origen*: nada más abrirse el cuadro de diálogo se crea una expresión XPath de forma automática. La expresión apunta al nodo del documento XML que estaba seleccionado al abrir el cuadro de diálogo "Seleccionar columnas". La expresión se puede editar con el teclado. La casilla *Incluir índices* determina si se usarán o no filtros de predicado en la XPath (consulte el apartado XPath de origen 353 para más información). Los nodos descendientes del nodo seleccionado por la XPath de origen se podrán seleccionar como columnas de datos de los ejes X e Y. El cuadro combinado *Profundidad de la búsqueda de columnas* determina en cuántos niveles de descendientes se puede buscar para devolver nodos que se puedan usar para la selección de datos de los ejes X e Y. Después de editar la XPath de origen, haga clic en el botón **Actualizar columnas** para que surtan efecto los cambios realizados y para que se actualicen las opciones del eje X y del eje Y.
- *Eje X*: la opción seleccionada en este cuadro combinado especifica el nodo que se debe usar como eje X. La secuencia que se devuelve para esta selección serán las etiquetas que aparecen en el eje X. La opción *Enumeración automática* del cuadro combinado ofrece etiquetas numeradas automáticamente para el eje X. Recuerde que las expresiones XPath creadas para el eje Y también estarán disponibles en la lista del cuadro combinado del eje X.
- *Eje Y*: las opciones que se activen en este panel serán los nodos cuyos valores numéricos se representarán en el eje Y numérico. Los botones **Activar todas** y **Desactivar todas** seleccionan todas las opciones y anulan la selección de todas las opciones del panel Y respectivamente. El botón **Insertar XPath** permite generar una serie que no esté disponible por no ser un descendiente del nodo que devuelve la XPath de origen. El nodo o la expresión XPath seleccionados para el eje X no estará disponible en el panel del eje Y y aparecerá atenuado en color gris.



Cómo se crea la tabla de datos del gráfico

Los datos utilizados para el gráfico vienen determinados por las opciones seleccionadas en el cuadro de diálogo "Seleccionar columnas". Explicaremos cómo se seleccionan los datos del gráfico con ayuda de un ejemplo. Puesto que el documento XML (que aparece casi al final de este apartado) contiene tres elementos llamados `Region`, la XPath de origen `/Data/Region` selecciona cada uno de esos elementos. Siendo los tres elementos `Region` el nodo de contexto, las columnas de datos se generan. Por cada elemento `Region` seleccionado con la XPath ocurre lo siguiente:

1. La expresión del eje X genera la primera columna (esta columna es por defecto la columna que se utiliza para las etiquetas del eje X).
2. Por cada serie (por cada opción seleccionada para el eje Y) se genera una columna.

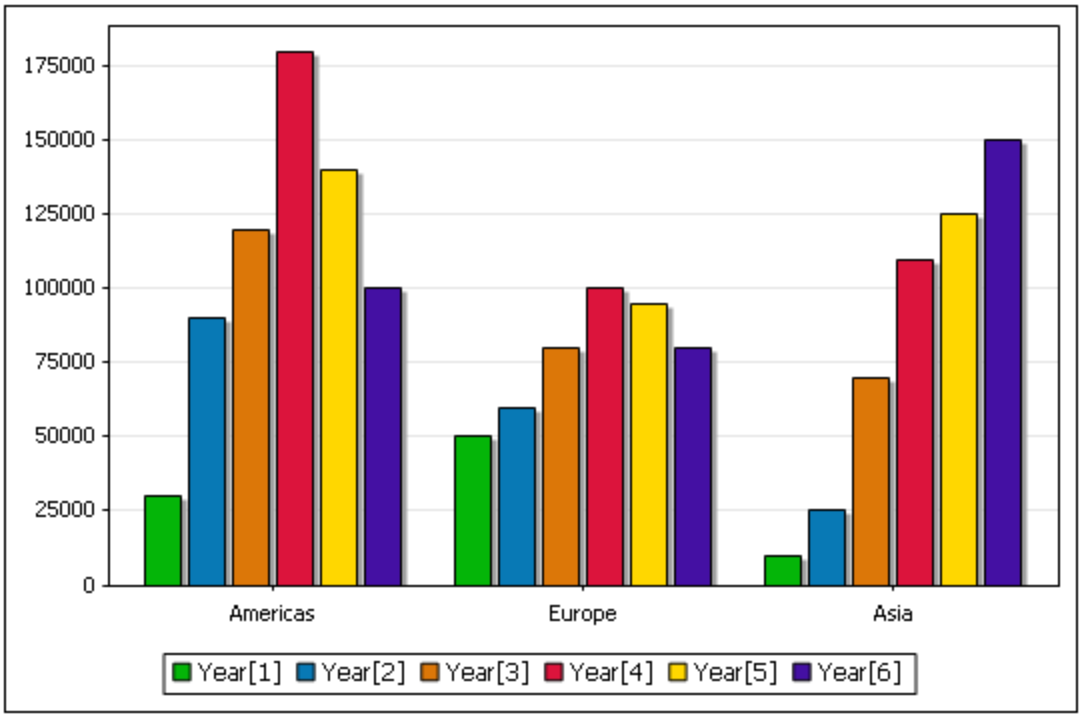
Los datos del gráfico generado en base a las opciones seleccionadas en el cuadro de diálogo "Seleccionar columnas" de la imagen anterior se pueden ver en esta tabla:

XPath de origen	Eje X	Eje Y (columnas de la serie)

© 2016-2022 Altova GmbH

Region[1]	@id	Year[1]	Year[2]	Year[3]	Year[4]	Year[5]	Year[6]
Region[2]	@id	Year[1]	Year[2]	Year[3]	Year[4]	Year[5]	Year[6]
Region[3]	@id	Year[1]	Year[2]	Year[3]	Year[4]	Year[5]	Year[6]

A continuación puede ver un gráfico de barras generado a partir de estos datos:



Es muy importante tener en cuenta estos aspectos:

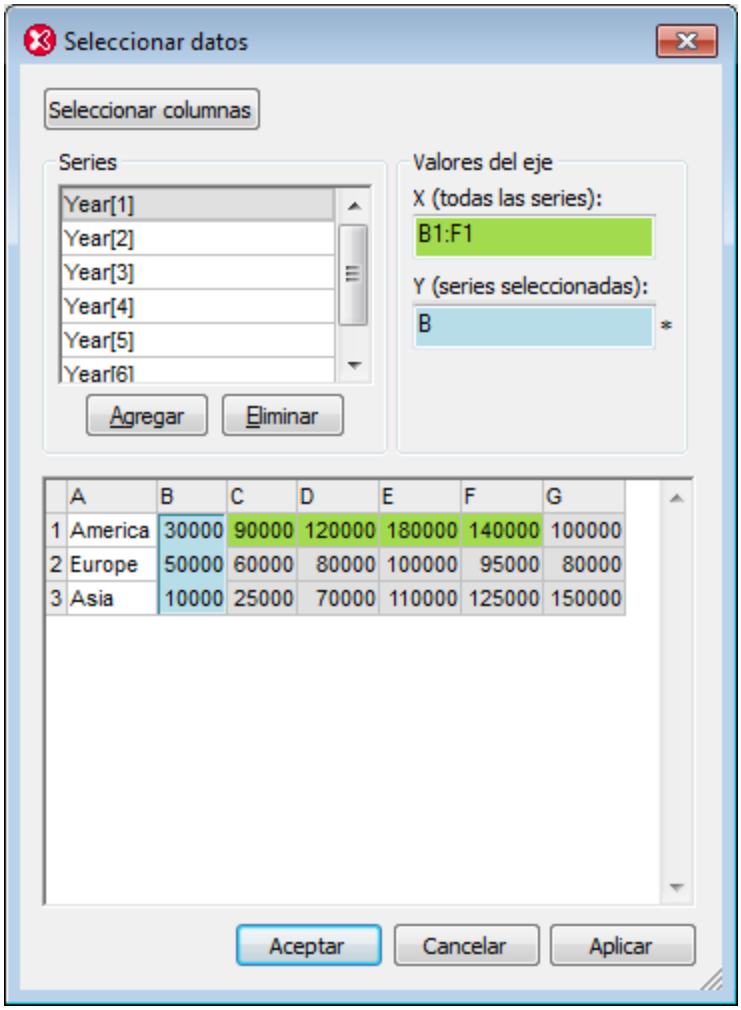
- El número de marcas de graduación del eje X dependerá del tamaño de la secuencia devuelta por la expresión XPath de origen (tres en este caso).
- Los nodos devueltos por la XPath de origen serán los nodos de contexto para generar dos conjuntos de datos para cada marca de graduación del eje X: (i) la marca de graduación del eje X (obtenida con las opciones seleccionadas en el eje X) y (ii) todas las series de datos que se deben dibujar para dicha marca de graduación (estas series se obtienen con las opciones seleccionadas en el eje Y). Las expresiones XPath introducidas para el eje X y el eje Y se evalúan como expresiones XPath en el contexto de estos nodos (de la XPath de origen).
- La secuencia devuelta por las opciones seleccionadas en el eje X pasará a ser la etiqueta de cada marca de graduación. Si hay menos etiquetas que marcas de graduación, algunas marcas de graduación no tendrán etiqueta.
- Cada serie (p. ej. la serie `Year[1]`) se evalúa una vez para cada nodo de contexto. En algunos gráficos, como los gráficos circulares o de barras simples, se puede usar una sola serie.
- Las leyendas del gráfico se obtienen de los nombres de los componentes de la serie.

A continuación aparece el documento XML utilizado para el ejemplo anterior. Se llama `YearlySales.xml` y se encuentra en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Tutorial`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <Region id="Americas">
              <Year id="2005">30000</Year>
              <Year id="2006">90000</Year>
              <Year id="2007">120000</Year>
              <Year id="2008">180000</Year>
              <Year id="2009">140000</Year>
              <Year id="2010">100000</Year>
       </Region>
       <Region id="Europe">
              <Year id="2005">50000</Year>
              <Year id="2006">60000</Year>
              <Year id="2007">80000</Year>
              <Year id="2008">100000</Year>
              <Year id="2009">95000</Year>
              <Year id="2010">80000</Year>
       </Region>
       <Region id="Asia">
              <Year id="2005">10000</Year>
              <Year id="2006">25000</Year>
              <Year id="2007">70000</Year>
              <Year id="2008">110000</Year>
              <Year id="2009">125000</Year>
              <Year id="2010">150000</Year>
       </Region>
</Data>
```

La selección de datos del cuadro de diálogo "Seleccionar columnas" que aparece más arriba se puede ver en la tabla del cuadro de diálogo "Seleccionar Datos". El diálogo "Seleccionar datos" se abre al hacer clic en el botón **Seleccionar datos** de la ventana Gráficos.

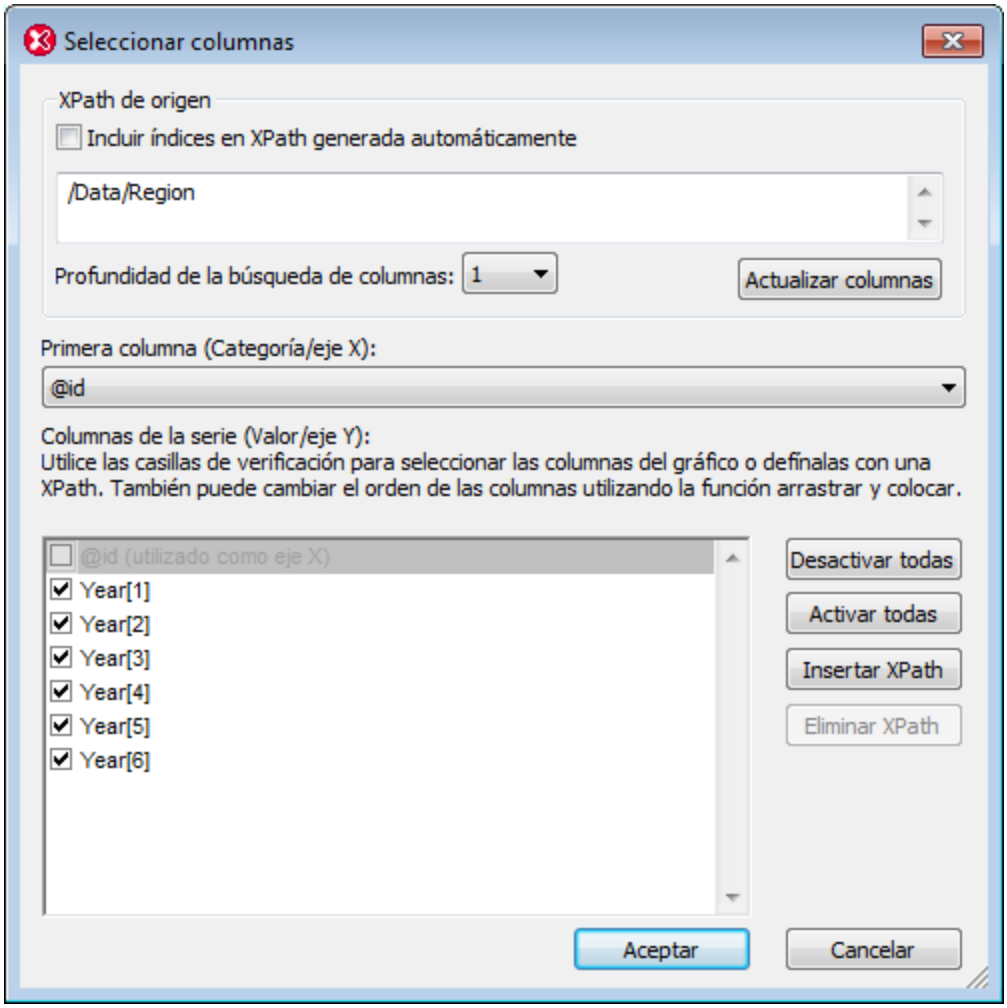
© 2016-2022 Altova GmbH



Para más información sobre los parámetros del cuadro de diálogo "Seleccionar columnas" consulte los apartados XPath de origen[353], Selección del eje X[357], Selección del eje Y[362] y Datos del gráfico[366].

5.12.2 XPath de origen

La XPath de origen se especifica en el cuadro de diálogo "Seleccionar columnas". Determina los nodos del documento que se podrán seleccionar como datos del eje X y del eje Y. El cuadro combinado *Profundidad de la búsqueda de columnas* determina en cuántos niveles de descendientes se buscará para devolver nodos que se puedan usar para los datos del eje X y del eje Y.



XPath de origen a partir de la posición del cursor

Para explicar qué XPath de origen se selecciona cuando se abre el cuadro de diálogo "Selccionar columnas", usaremos el documento XML que aparece a continuación. El documento se llama `YearlySales.xml` y se encuentra en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Tutorial`.

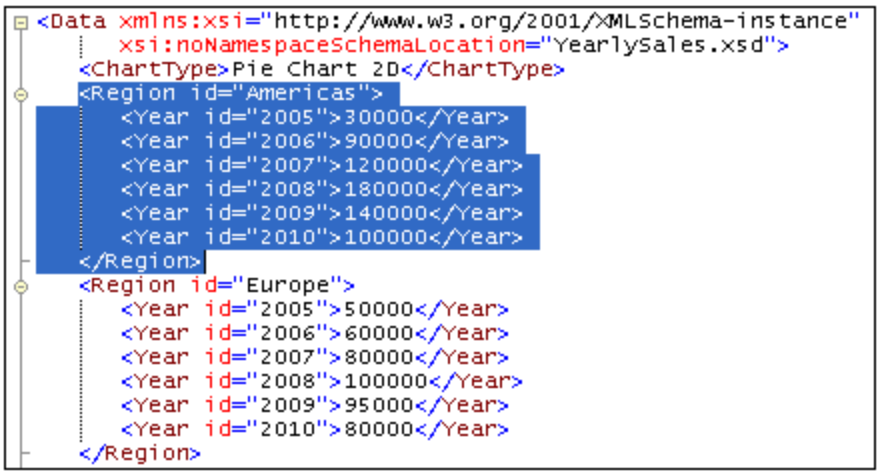
```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <Region id="Americas">
              <Year id="2005">30000</Year>
              <Year id="2006">90000</Year>
              <Year id="2007">120000</Year>
              <Year id="2008">180000</Year>
              <Year id="2009">140000</Year>
              <Year id="2010">100000</Year>
       </Region>
       <Region id="Europe">
```


© 2016-2022 Altova GmbH

```
            <Year id="2005">50000</Year>
            <Year id="2006">60000</Year>
            <Year id="2007">80000</Year>
            <Year id="2008">100000</Year>
            <Year id="2009">95000</Year>
            <Year id="2010">80000</Year>
      </Region>
      <Region id="Asia">
            <Year id="2005">10000</Year>
            <Year id="2006">25000</Year>
            <Year id="2007">70000</Year>
            <Year id="2008">110000</Year>
            <Year id="2009">125000</Year>
            <Year id="2010">150000</Year>
      </Region>
</Data>
```

Pueden darse todos estos casos:

- Si el cursor se sitúa en cualquier posición dentro de la etiqueta inicial (incluido el valor de atributo) o de la etiqueta final del elemento `Data` o en cualquier posición dentro del elemento `Data` pero no en un nodo descendiente, entonces la XPath de origen será: `/Data`
- Si el cursor se sitúa en cualquier posición dentro de la etiqueta inicial (incluido el valor de atributo) o de la etiqueta final de un elemento `Region` o en cualquier posición dentro de un elemento `Region` pero no en un nodo descendiente, entonces la XPath de origen será: `/Data/Region`
- Si el cursor se sitúa en cualquier posición dentro de la etiqueta inicial (incluido el valor de atributo) o de la etiqueta final de un elemento `Year` o en cualquier posición dentro de un elemento `Year`, entonces la XPath de origen será: `/Data/Region[N]/Year`. El filtro de predicado `[N]` selecciona el elemento `Region` dentro del cual está el elemento `Year` seleccionado. De ese modo, la selección de datos del eje X y del eje Y se limita a los elementos `Year` de este elemento `Region` concreto.
- Si desea seleccionar un solo elemento `Region`, por ejemplo `/Data/Region[1]`, resalte este elemento (es decir, el primer elemento `Region` tal y como aparece en la imagen siguiente).



- Asimismo, si resalta dos elementos `Region`, se genera una expresión XPath que selecciona estos dos elementos `Region` solamente.

Posteriormente puede editar la expresión XPath de origen en el cuadro de diálogo "Seleccionar columnas".

En la vista Cuadrícula los datos se seleccionan haciendo clic en un nodo o resaltando un conjunto de nodos. La XPath de origen se genera a partir de la selección de la vista Cuadrícula del mismo modo que en la vista Texto, tal y como se describe más arriba.

Incluir índices en la XPath

La casilla *Incluir índices* determina si se usan filtros de predicado o no en la expresión XPath, si estos filtros de predicado se insertan automáticamente nada más abrir el cuadro de diálogo "Seleccionar columnas" o si se introducen manualmente. Por ejemplo, si el cursor se sitúa dentro de un elemento descendiente del primer elemento `Region` y se marca la casilla *Incluir índices*, entonces la expresión XPath generada automáticamente será, por ejemplo: `/Data/Region[1]/Year`. Si no se marcara la casilla *Incluir índices*, entonces la expresión sería esta: `/Data/Region/Year`.

La casilla *Incluir índices* también determina si los predicados introducidos manualmente se deben conservar. Por ejemplo, si desea usar predicados en la expresión XPath de origen, debe marcar la casilla *Incluir índices*.

Repercusiones de la selección de la XPath de origen

Tenga en cuenta que la XPath de origen que seleccione tiene algunas repercusiones:

- El número de componentes en la secuencia devuelta por la XPath de origen determina el número de marcas de graduación presentes en el eje X. Para cambiar el número de marcas de graduación del eje X hay dos opciones: modificar la XPath de origen o seleccionar un número de etiquetas para cualquier serie mayor al número de marcas de graduación. Para más información sobre este caso, consulte el apartado selección del eje X 357.
- El nodo de la XPath de origen es el nodo antecesor de todos los nodos disponibles para la selección de datos del eje X y del eje Y y para todas las expresiones XPath permitidas.
- Como resultado de estos dos puntos, recuerde que los cambios realizados en la expresión XPath de origen no sólo afectan al número de marcas de graduación del eje X sino también al contexto de todas las expresiones XPath relacionadas con el gráfico.

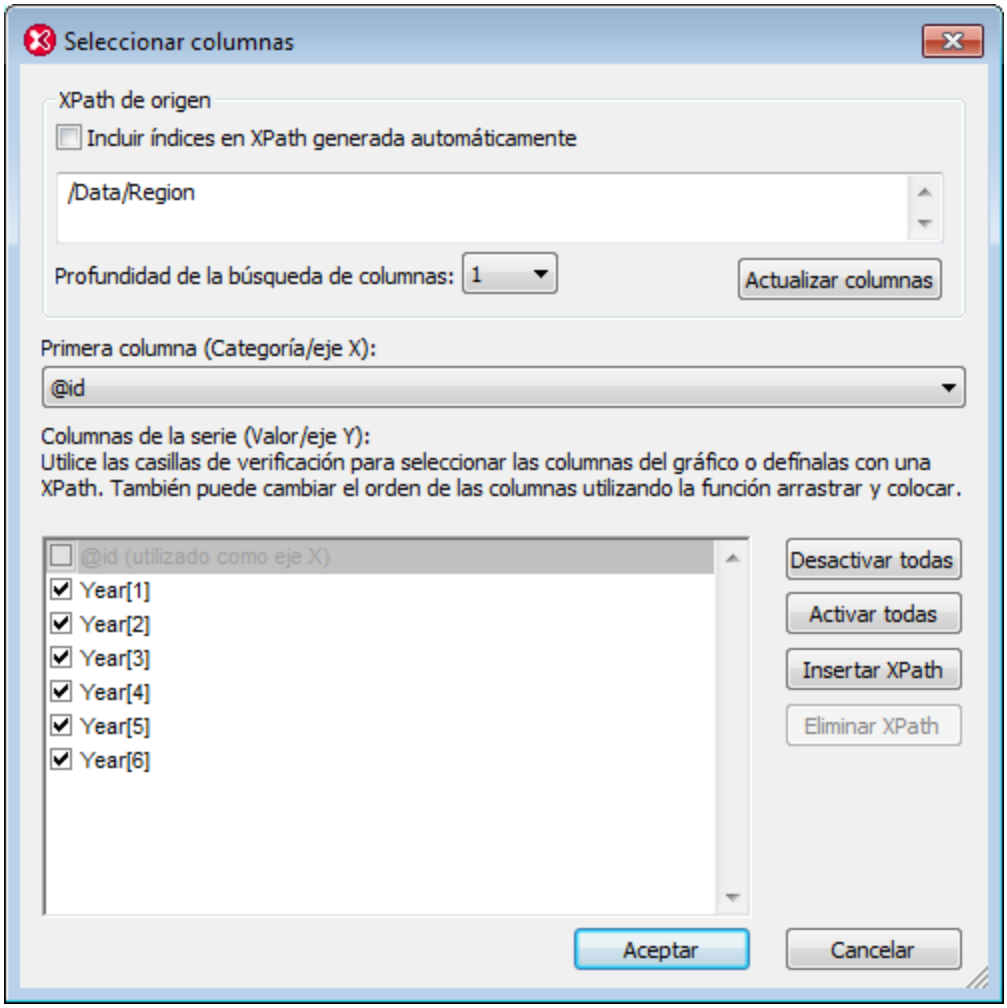
Por ejemplo, a continuación citamos algunas repercusiones de algunas expresiones XPath (usando el documento XML de los ejemplos anteriores):

- **`/Data/Region`**: devuelve los 3 elementos `Region`, es decir, el eje X tendrá 3 marcas de graduación. Cada elemento `Region` será a su vez el nodo de contexto para las expresiones XPath.
- **`/Data/Region/Year`**: devuelve 18 elementos `Year`, es decir, el eje X tendrá 18 marcas de graduación. Cada elemento `Year` será a su vez el nodo de contexto para las expresiones XPath.
- **`/Data/Region[1]/Year`**: devuelve los 6 elementos `Year` secundarios del primer elemento `Region`, es decir, el eje X tendrá 6 marcas de graduación. Cada elemento `Year` del primer elemento `Region` será a su vez el nodo de contexto para las expresiones XPath.
- **`distinct-values(//Year/@id)`**: devuelve 6 componentes (los valores diferentes del atributo `Year/@id`, es decir, `2005`, `2006`, `2007`, `2008`, `2009` y `2010`). Sin embargo, puesto que esta expresión XPath no devuelve un nodo, esta no puede servir de nodo de contexto para las expresiones XPath. Si los componentes de esta secuencia se deben usar como nodos de destino en el documento XML (usando, por ejemplo, la función `current()` cuya forma abreviada es "."), entonces la expresión XPath que use el componente actual debe empezar en la raíz del documento o no se podrá establecer el contexto. Por ejemplo: `/Data/Region[1]/Year[@id eq .]`.

© 2016-2022 Altova GmbH

5.12.3 Selección del eje X

La selección del eje X se especifica en el cuadro de diálogo "Seleccionar columnas" (*imagen siguiente*). Esta selección determina las etiquetas que aparecerán en el eje X. Las etiquetas se pueden editar posteriormente en el cuadro de diálogo "Seleccionar datos" (*ver más abajo*).



Tomemos este documento XML como ejemplo (el documento `YearlySales.xml` que se encuentra en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Tutorial`.) Situamos el cursor en la etiqueta inicial del primer elemento `Region` y hacemos clic en el botón **Gráfico nuevo** de la ventana Gráficos. Aparece el cuadro de diálogo "Seleccionar columnas", que incluye esta XPath de origen: `/Data/Region` (*ver imagen anterior*).

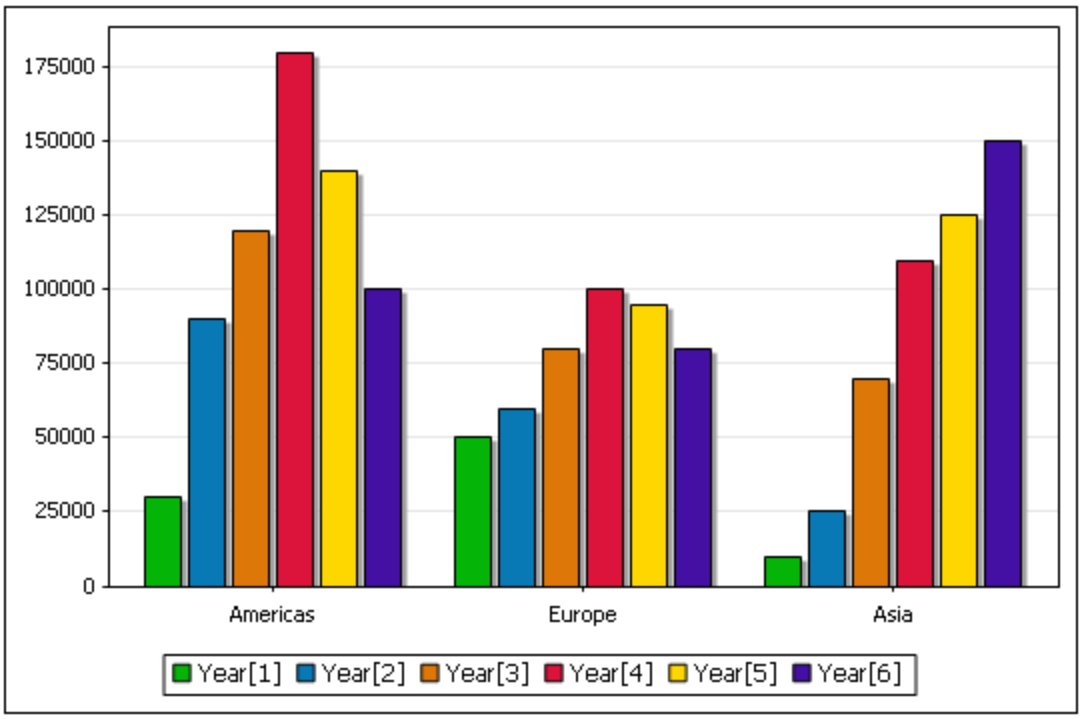
```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <Region id="Americas">
              <Year id="2005">30000</Year>
              <Year id="2006">90000</Year>
              <Year id="2007">120000</Year>
              <Year id="2008">180000</Year>
```

```
            <Year id="2009">140000</Year>
            <Year id="2010">100000</Year>
        </Region>
        <Region id="Europe">
            <Year id="2005">50000</Year>
            <Year id="2006">60000</Year>
            <Year id="2007">80000</Year>
            <Year id="2008">100000</Year>
            <Year id="2009">95000</Year>
            <Year id="2010">80000</Year>
        </Region>
        <Region id="Asia">
            <Year id="2005">10000</Year>
            <Year id="2006">25000</Year>
            <Year id="2007">70000</Year>
            <Year id="2008">110000</Year>
            <Year id="2009">125000</Year>
            <Year id="2010">150000</Year>
        </Region>
</Data>
```

Como se explica en el apartado XPath de origen 353, esta XPath de origen crea un gráfico con tres marcas de graduación en el eje X (porque la XPath de origen devuelve tres componentes: los tres elementos `Region`). En este gráfico queremos que el nombre de estas tres marcas de graduación sea el nombre de las tres regiones y, por tanto, seleccionamos el atributo `@id` en el cuadro combinado del eje X (*ver imagen anterior del cuadro de diálogo "Seleccionar columnas"*).

Para producir los datos de cada marca de graduación, se evalúa a su vez cada elemento `Region`. Por cada elemento `Region`, el atributo `id` genera la etiqueta correcta para la marca de graduación del eje X. El eje X tendría este aspecto.

© 2016-2022 Altova GmbH

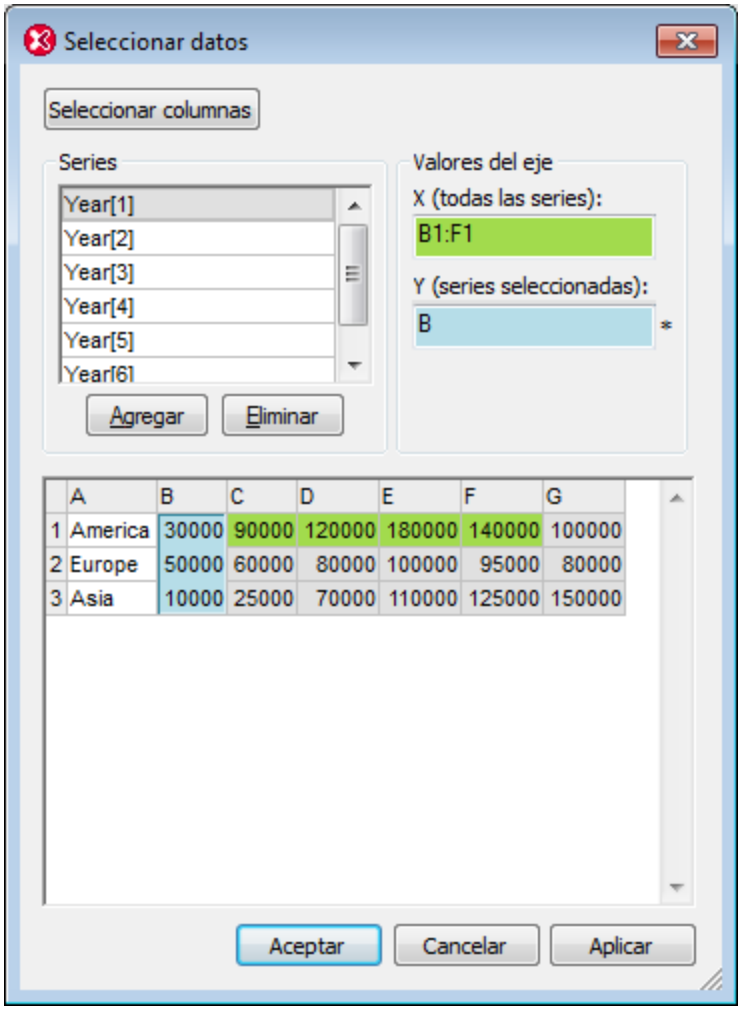


Si se selecciona otra expresión XPath diferente en el cuadro combinado del eje X, la expresión se evalúa dentro del contexto del elemento `Region` correspondiente y el resultado evaluado será la etiqueta de la marca de graduación correspondiente. La opción *Enumeración automática* genera una secuencia numérica que equivale al número de la marca de graduación, es decir, la primera marca será la marca 1, la segunda 2 y así sucesivamente.

Cambiar las etiquetas y los números de las marcas de graduación del eje X

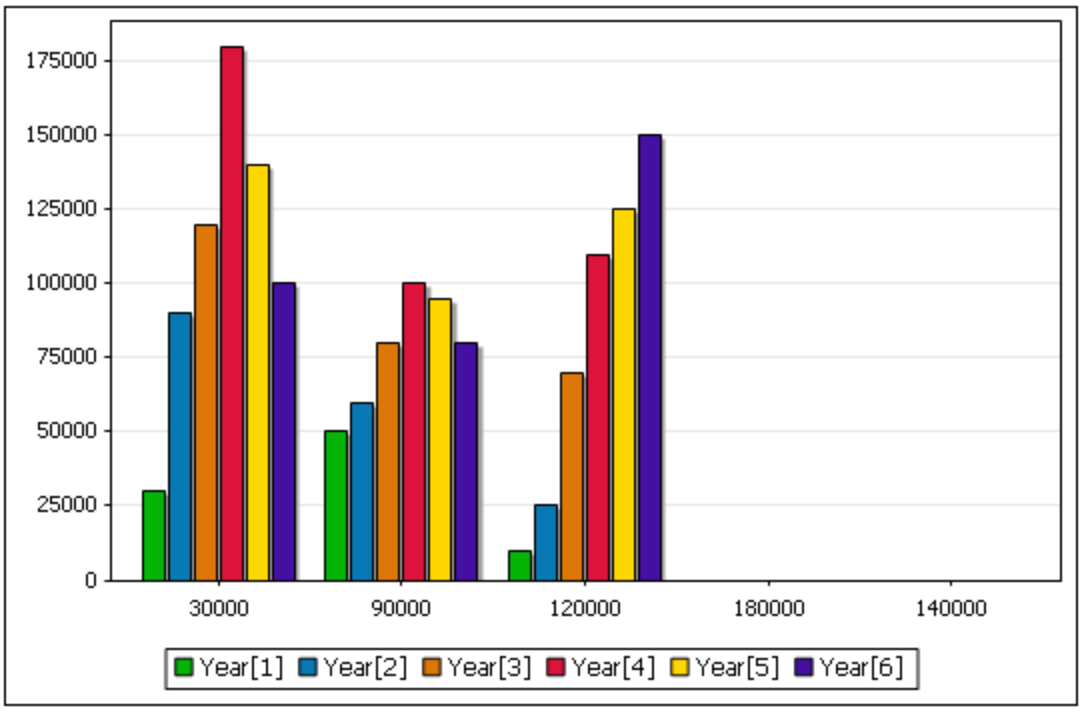
La selección de las etiquetas del eje X se puede modificar en el cuadro de diálogo "Seleccionar datos" (clic en el botón **Seleccionar datos** de la ventana Gráficos).

En el cuadro de diálogo "Seleccionar datos" que aparece a continuación, por ejemplo, haga clic en el cuadro de texto del eje X del panel "Valores del eje". Aquí puede modificar la selección del eje X. Haga clic en el campo `B1` y arrastre el ratón hasta el campo `F1`, seleccionando así `B1:F1`. Haga clic en **Aceptar** para ver el gráfico nuevo.



Esta nueva selección ofrece las etiquetas que aparecen a continuación para las marcas de graduación. Observe que, puesto que la nueva selección incluye cinco componentes, se generaron cinco marcas de graduación. Sin embargo, las únicas que tienen datos son las tres primeras. Esto se debe a que la XPath de origen devuelve tres nodos y estos son los nodos que se procesan para el gráfico. Estos tres nodos se corresponden con las filas de la tabla[350] que aparece en el cuadro de diálogo "Seleccionar datos". Recuerde que para cambiar el número de filas de la tabla es necesario cambiar la XPath de origen.

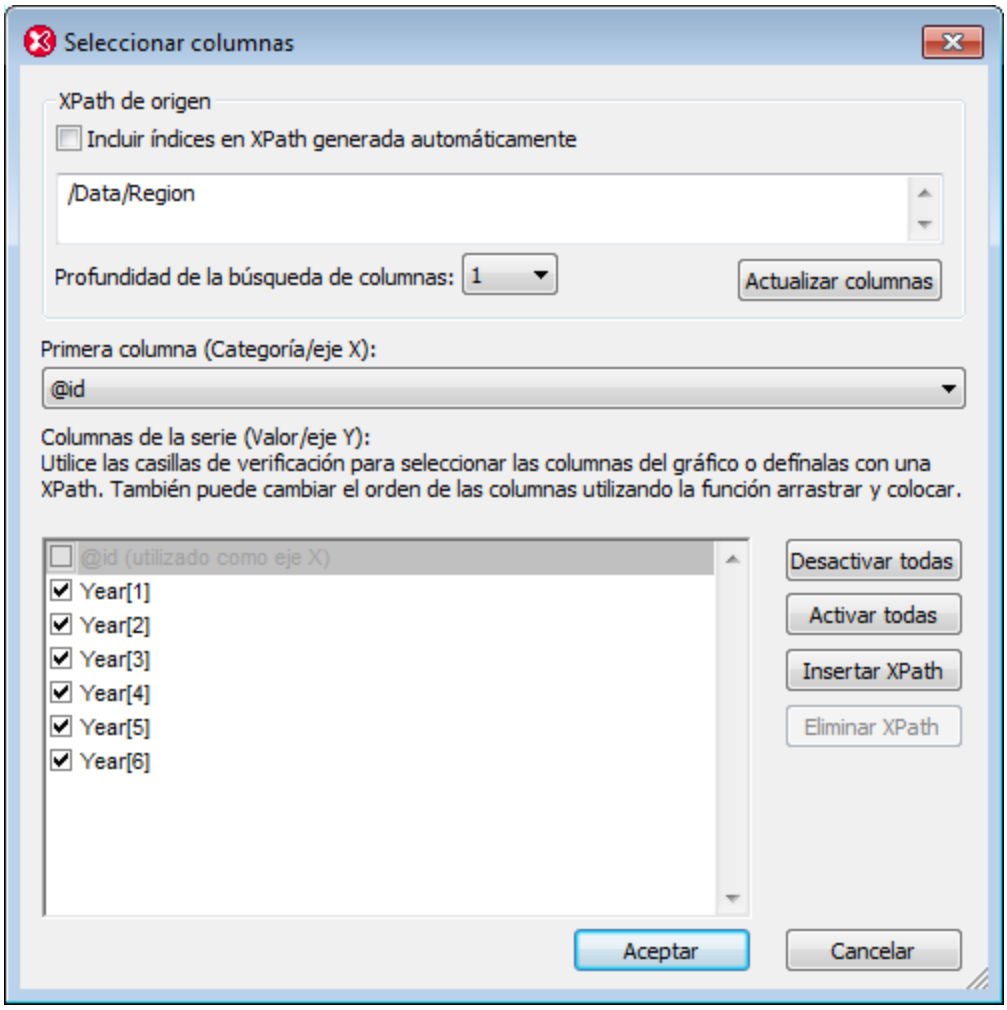
© 2016-2022 Altova GmbH



Para más información sobre la XPath de origen y la selección de datos para el eje X, consulte el apartado Creación de gráficos 350.

5.12.4 Selección del eje Y

El eje Y (*imagen siguiente*) también se denomina eje de serie.

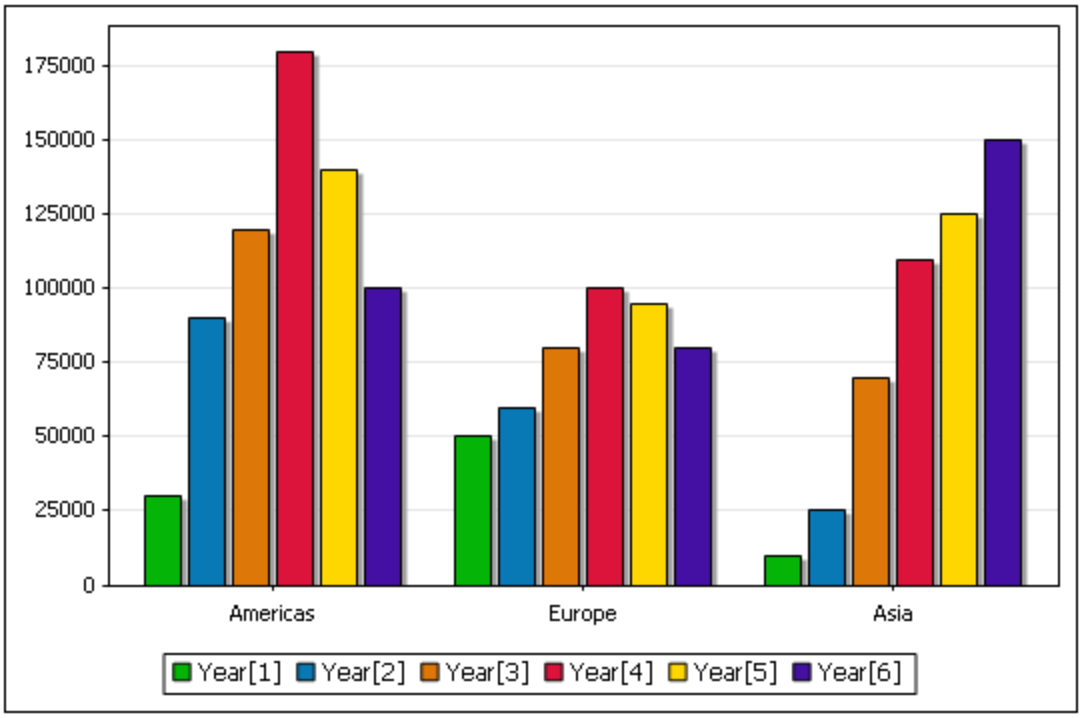


Las opciones seleccionadas para este eje determinan cuántas series se dibujan para cada marca de graduación del eje X. Si se selecciona una sola serie, entonces en cada marca de graduación del eje X se dibuja el valor devuelto por la expresión XPath para esa serie. Si se seleccionan más series, como en la imagen siguiente que muestra seis series, entonces la selección de datos del gráfico será la de la tabla que aparece a continuación. (El nodo de contexto[350] para la selección de datos del eje Y es el elemento `Region` correspondiente.)

XPath de origen	Eje X	Eje Y (columnas de la serie)					
`Region[1]`	`@id`	`Year[1]`	`Year[2]`	`Year[3]`	`Year[4]`	`Year[5]`	`Year[6]`
`Region[2]`	`@id`	`Year[1]`	`Year[2]`	`Year[3]`	`Year[4]`	`Year[5]`	`Year[6]`
`Region[3]`	`@id`	`Year[1]`	`Year[2]`	`Year[3]`	`Year[4]`	`Year[5]`	`Year[6]`

© 2016-2022 Altova GmbH

El gráfico que resulta de esta selección tendrá este aspecto:



Hay tres marcas de graduación en el eje X, cuyas etiquetas se corresponden con el valor de los respectivos atributos `Region/@id`. En cada marca de graduación del eje X se evalúa la expresión XPath para cada serie. En el ejemplo que nos ocupa, por cada marca de graduación del eje X, se evalúa y se dibuja cada una de las seis series. Por ejemplo, la primera serie (`Year[1]`) se dibuja para las tres regiones y lo mismo ocurre con el resto de las series (de la serie `Year[2]` a la serie `Year[6]`).

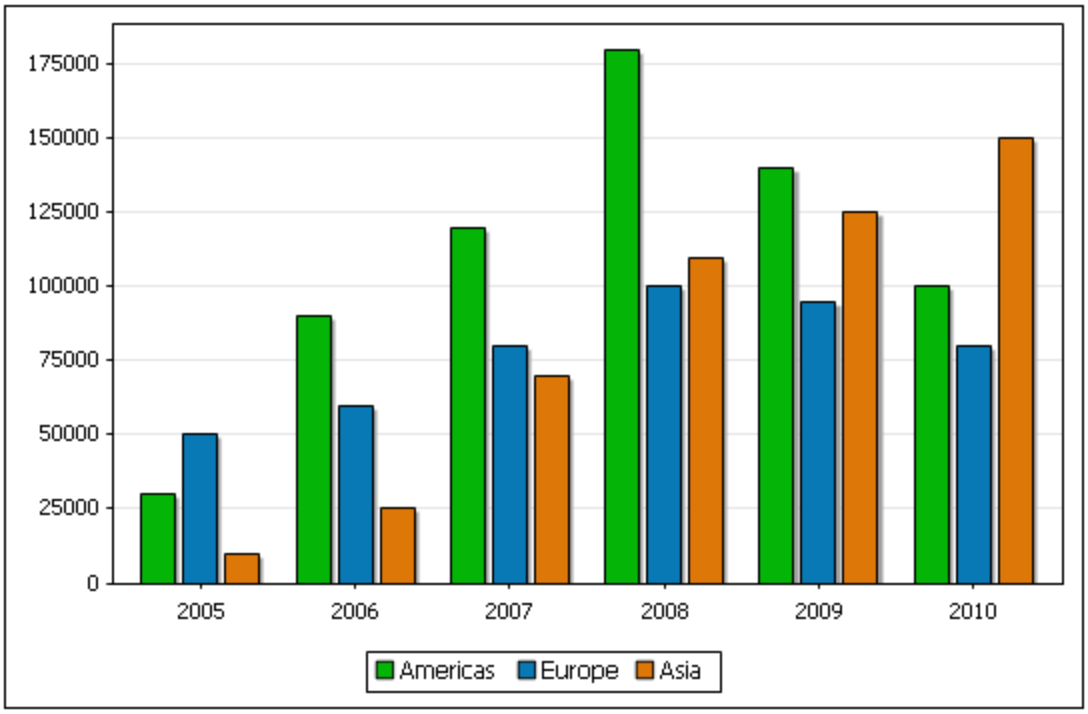
Nota: algunos gráficos, como los gráficos circulares y de barras simples, admiten un solo eje. En un gráfico de barras simples, por ejemplo, cada marca de graduación del eje X tendrá una sola barra: la barra que representa la serie única. En un gráfico circular, los valores de la serie única sumarán el 100% del círculo y cada valor se asigna a una marca de graduación del eje X.

Leyendas del eje Y

Las leyendas que aparecen debajo del gráfico son los nombres de las series. Estos nombres se pueden cambiar en el cuadro de diálogo Seleccionar datos 366.

Intercambiar la selección del eje X por la del eje Y

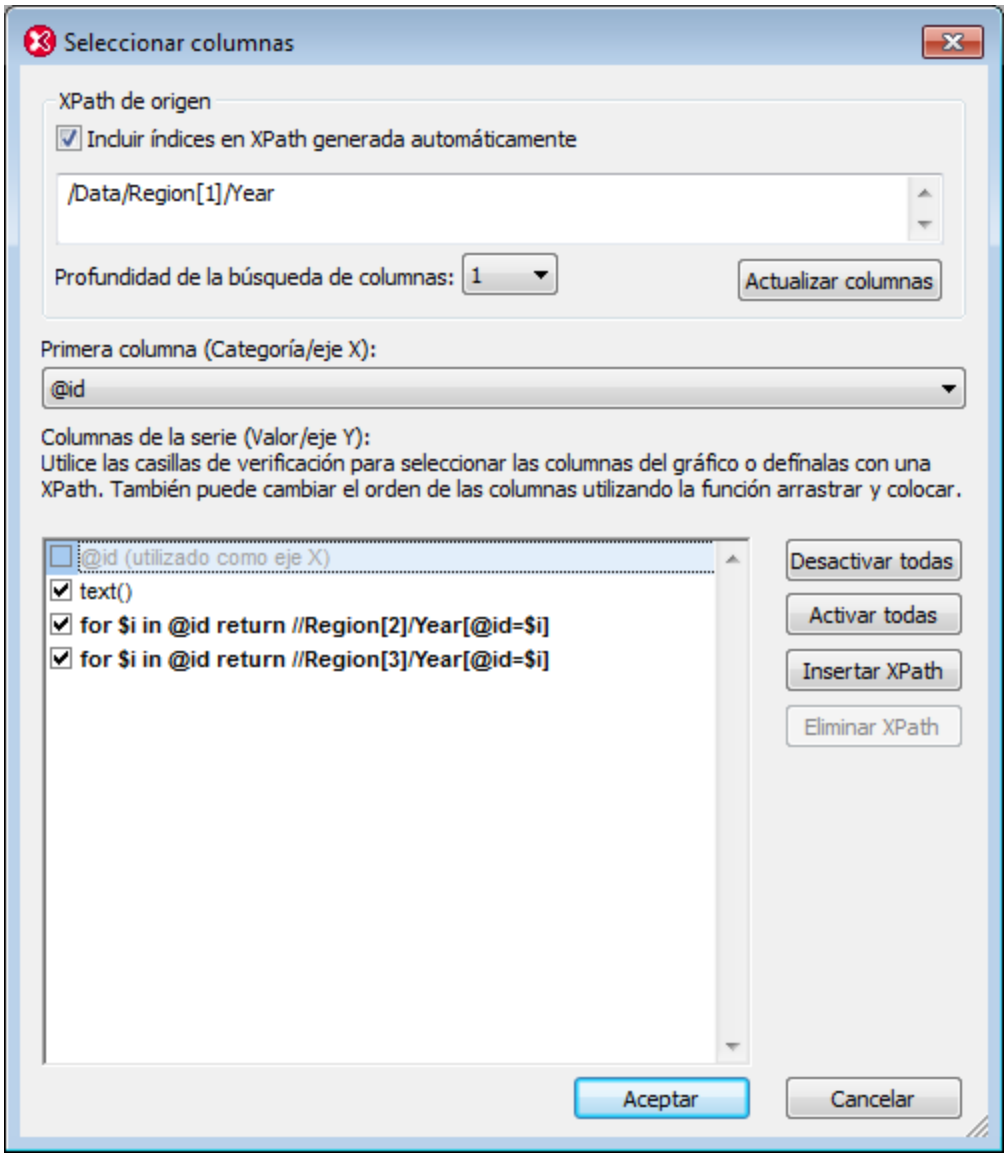
En el ejemplo anterior, las regiones están en el eje X y las ventas anuales están trazadas en el eje Y de cada región. Los elementos `Year` son las series. Pero, ¿y si preferimos dibujar los años en el eje X y comparar las ventas regionales de cada año, igual que en el gráfico de barras que aparece a continuación? Para ello necesitamos seis marcas de graduación (que se obtienen con la selección de la XPath de origen). Después necesitamos etiquetar las marcas de graduación del eje X con los años correspondientes y, por último, debemos seleccionar tres series (para las regiones), las cuales se representarán en cada marca de graduación. La imagen siguiente, del cuadro de diálogo "Seleccionar columnas", muestra cómo se consigue llegar a esta selección.



Seleccionar las series

Para convertir un nodo en una serie del gráfico, marque la casilla de dicho nodo. También puede cambiar la XPath de origen y el valor del cuadro combinado *Profundidad de la búsqueda de columnas* para que dicho nodo aparezca en el panel "Series". Otra opción es añadir una expresión XPath para seleccionar un nodo, tal y como se hizo en el ejemplo de la imagen siguiente. Para más información sobre este caso de uso, consulte el apartado Ejemplo de gráfico avanzado 401.

© 2016-2022 Altova GmbH



Referencia

A continuación mostramos el documento XML usado como ejemplo en este apartado. El documento se llama `YearlySales.xml` y se encuentra en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Tutorial`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <Region id="Americas">
              <Year id="2005">30000</Year>
              <Year id="2006">90000</Year>
              <Year id="2007">120000</Year>
              <Year id="2008">180000</Year>
              <Year id="2009">140000</Year>
```

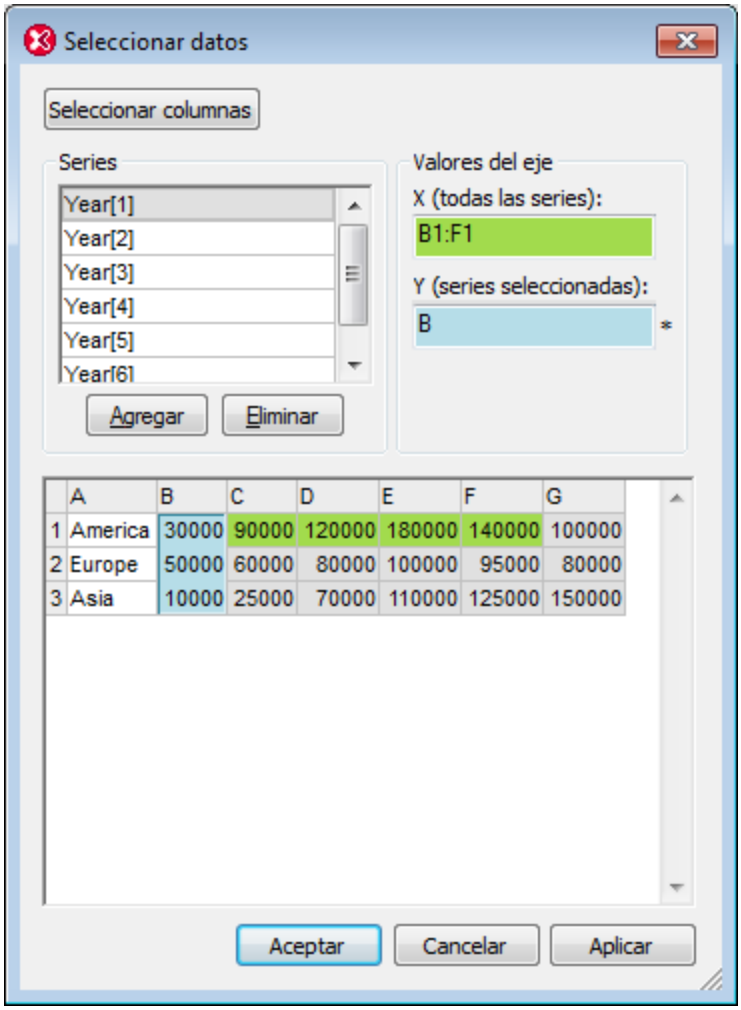
```
            <Year id="2010">100000</Year>
        </Region>
        <Region id="Europe">
            <Year id="2005">50000</Year>
            <Year id="2006">60000</Year>
            <Year id="2007">80000</Year>
            <Year id="2008">100000</Year>
            <Year id="2009">95000</Year>
            <Year id="2010">80000</Year>
        </Region>
        <Region id="Asia">
            <Year id="2005">10000</Year>
            <Year id="2006">25000</Year>
            <Year id="2007">70000</Year>
            <Year id="2008">110000</Year>
            <Year id="2009">125000</Year>
            <Year id="2010">150000</Year>
        </Region>
</Data>
```

5.12.5 Datos del gráfico

Al hacer clic en el botón **Seleccionar datos** se abre el cuadro de diálogo "Seleccionar datos" (*imagen siguiente*). El diálogo se divide en tres paneles: (i) el panel "Series", (ii) el panel "Valores del eje" y (iii) la tabla de datos del gráfico. A continuación describimos cada uno de estos paneles.

El botón **Seleccionar columnas** abre el cuadro de diálogo Seleccionar columnas 349, donde puede cambiar la XPath de origen y cambiar la selección de datos para los ejes X e Y.

 © 2016-2022 Altova GmbH



Series

El panel "Series" contiene las series seleccionadas en el cuadro de diálogo Seleccionar columnas[349]. Cuando haga clic en el botón **Aceptar**, las series de este panel serán las series que aparecerán en el gráfico. Estas son las tres operaciones que se pueden realizar en el panel Series:

- *Agregar y eliminar series:* así puede controlar el número de series que aparecerán en el gráfico.
- *Editar el nombre de las series*: el nombre de las series son las leyendas que aparecen en el gráfico.
- *Seleccionar datos para cada serie:* seleccione una serie en el panel "Series" y especifique los datos del eje X y del eje Y en el panel "Valores del eje" (ver subapartado siguiente).

Valores del eje

Los datos del eje X y del eje Y se pueden especificar en los respectivos cuadros de texto del panel "Valores del eje". Para ello haga clic en el cuadro de texto y aparece un asterisco que indica que puede editar el valor del cuadro de texto. Para seleccionar los datos introduzca un rango de la tabla de datos. Ese rango puede ser una columna, una fila entera o parte de una columna o de una fila. También puede introducir los datos con el teclado (escriba `A` o `3` o `B1:F1`). Para marcar el rango de datos, seleccione la primera celda del rango y

arrastre el cursor hasta la última celda del rango. Para marcar una columna o una fila entera, seleccione el encabezado de la columna o de la fila.

Los datos seleccionados para el eje X determinan las etiquetas de los nodos del eje X y se aplican a todas las series. Sin embargo, no afecta al número de marcas de graduación del eje X.

Los datos seleccionados para el eje Y determinan qué rango de celdas se usarán para la serie seleccionada. Si el número de celdas seleccionadas es inferior al número de marcas de graduación del eje X, entonces esta serie concreta no se representará en las marcas de graduación. Si el número de celdas seleccionadas es superior al número de marcas de graduación del eje X, se crearán más marcas de graduación. Las marcas de graduación adicionales equivaldrán al número de celdas adicionales. Los valores adicionales para esta serie se representarán en las marcas de graduación adicionales.

Tabla de datos del gráfico

La estructura de la tabla de datos del gráfico (panel inferior del cuadro de diálogo "Seleccionar datos") se obtiene de las opciones seleccionadas en el cuadro de diálogo Seleccionar columnas 349.

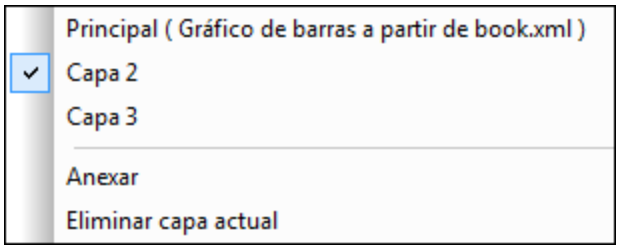
- El número de filas de la tabla es igual al número de componentes de la secuencia devuelta por la XPath de origen.
- El nombre de las columnas es una letra, empezando por la A. Este tipo de nomenclatura permite seleccionar las columnas con facilidad en el panel "Valores del eje" (`B1:F1`, por ejemplo).
- La primera columna se obtiene evaluando la opción seleccionada para el eje X en el cuadro de diálogo Seleccionar columnas 349 en los nodos de contexto devueltos por la expresión XPath de origen.
- Todas las columnas excepto la primera se obtienen evaluando las opciones seleccionadas para el eje Y en el cuadro de diálogo Seleccionar columnas 349. Cada serie de las opciones seleccionadas para el eje Y en el cuadro de diálogo Seleccionar columnas 349 corresponde a una columna de la tabla de datos del gráfico.

La tabla de datos del gráfico se puede entender como un supraconjunto de datos que se selecciona usando los parámetros del cuadro de diálogo Seleccionar columnas 349. En este supraconjunto puede seleccionar los rangos de datos que desea utilizar (panel "Valores del eje") para cada una de las series.

5.12.6 Gráficos multicapa

Un gráfico multicapa incluye varias capas superpuestas sobre el gráfico base. Para agregar una capa:

1. Haga clic en el botón **Multicapas** para abrir el menú de opciones (*imagen siguiente*).



2. Haga clic en **Anexar** para anexar una capa nueva.
3. Ahora, con la nueva capa seleccionada, haga clic en el botón **Gráfico nuevo** y seleccione datos para el gráfico tal y como se describe en el apartado Creación de gráficos 349.

 © 2016-2022 Altova GmbH

4. Para cambiar de tipo de gráfico y modificar su aspecto haga clic en los botones **Cambiar de tipo** y **Cambiar aspecto** respectivamente.

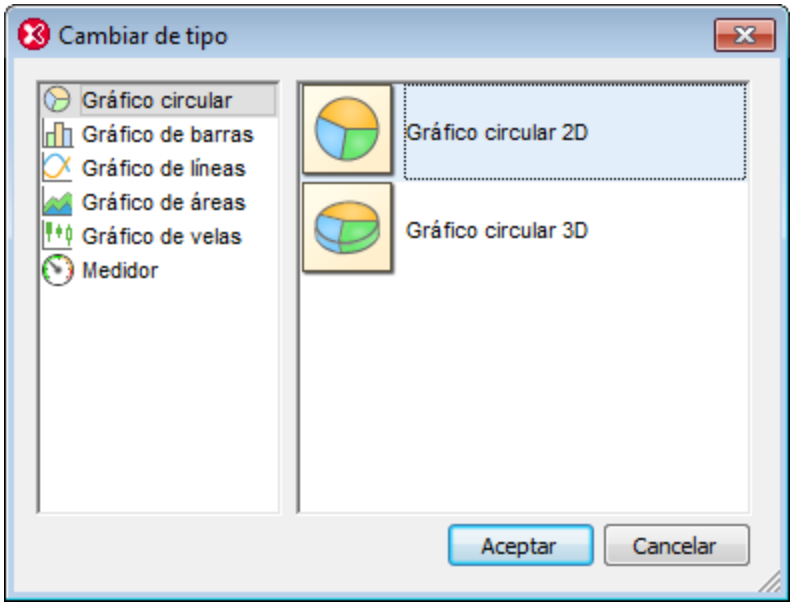
Puede añadir tantas capas como desee al gráfico multicapa. Cada capa nueva se anexará a las capas que ya existen y se sobrepondrán en el gráfico multicapa. Si desea cambiar el orden de las capas deberá volver a crearlas en el orden correcto. Para eliminar la capa que está seleccionada haga clic en **Eliminar capa actual**.

Nota: cada capa ocultará las capas que están debajo. A la hora de ordenar los gráficos debe tener en cuenta que los únicos gráficos que pueden ser transparentes son los gráficos de áreas. Por ejemplo, si coloca una capa de gráfico de barras encima de un gráfico de líneas, algunas partes del gráfico de líneas se ocultarán.

5.12.7 Configuración de gráficos: referencia rápida

Tipo de gráfico

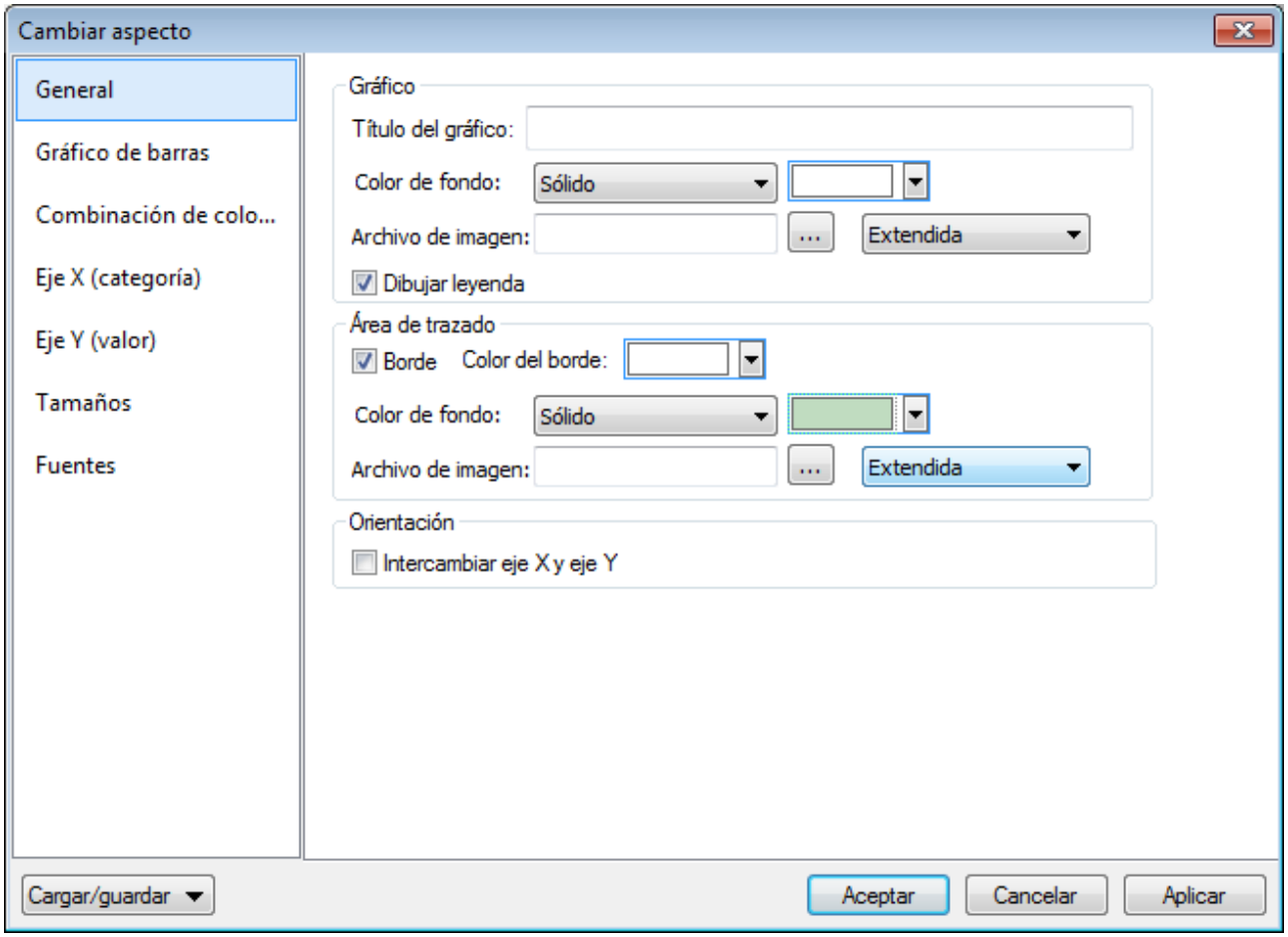
Para seleccionar el tipo de gráfico en el cuadro de diálogo "Cambiar de tipo" (*imagen siguiente*), seleccione el tipo de gráfico que desea utilizar y haga clic en **Aceptar**. Para abrir el cuadro de diálogo "Cambiar de tipo" haga clic en el botón **Cambiar de tipo**.



Después de seleccionar el tipo de gráfico, debe cambiar las opciones de configuración (título, altura, ancho, etc.) en el cuadro de diálogo "Cambiar aspecto" (*imagen siguiente*) y también debe especificar los datos del gráfico. En los apartados Creación de gráficos 349 y Datos del gráfico 366 encontrará información sobre cómo seleccionar datos para sus gráficos.

Aspecto del gráfico

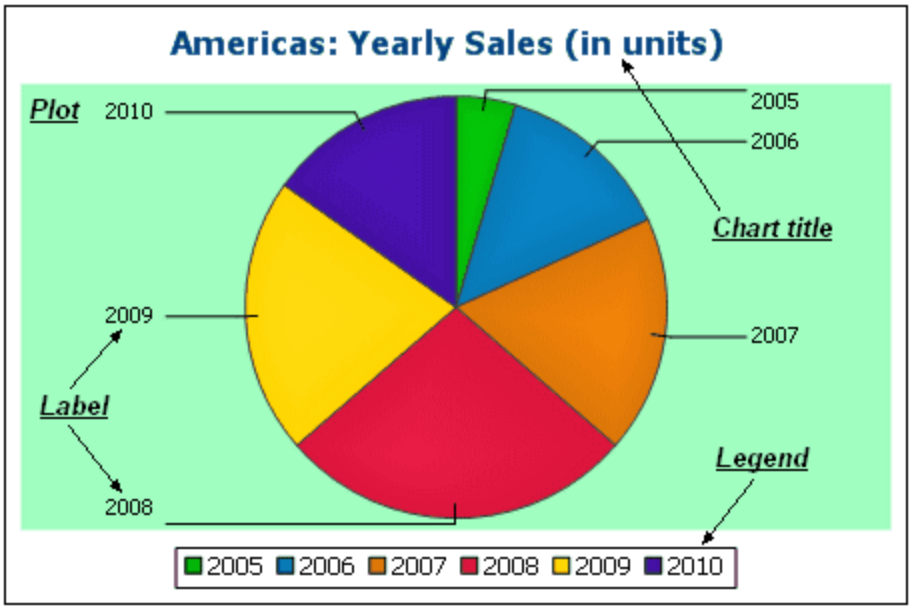
Las opciones de configuración del gráfico (título del gráfico, combinación de colores, tamaño de la fuente, etc.) se editan en el cuadro de diálogo "Cambiar aspecto" (*véase la imagen siguiente, que muestra la configuración de un gráfico de barras*). Para abrir este cuadro de diálogo haga clic en el botón **Cambiar aspecto**. Las opciones del cuadro de diálogo dependen del tipo de gráfico seleccionado.



Las diferentes opciones de configuración están organizadas por pestañas:

- ***General:*** en esta pestaña puede editar el título del gráfico (*imagen siguiente*), así como el color de fondo y el borde del área de trazado y su color de fondo. En la imagen siguiente, el área de trazado tiene un color de fondo verde pálido. Las leyendas están situadas debajo del gráfico y explican los códigos de colores del gráfico. Estas leyendas se pueden activar o desactivar..

© 2016-2022 Altova GmbH



- ***Combinación de colores:*** en esta pestaña puede elegir entre cuatro combinaciones de colores predefinidas o puede crear su propia combinación de colores. También puede cambiar las combinaciones de colores predefinidas añadiendo o eliminando colores. La combinación de colores seleccionada en la pestaña *Combinación de colores* es la combinación que se usará en el gráfico.
- ***Tamaños:*** en esta pestaña puede especificar el tamaño de los distintos aspectos del gráfico, ya sea en píxeles o en porcentajes.
- ***Fuentes:*** en esta pestaña puede especificar las propiedades de las fuentes del título del gráfico, de las etiquetas y de las leyendas. El tamaño de la fuente se puede indicar en píxeles o como porcentaje del tamaño del gráfico.

Además, cada tipo de gráfico tiene opciones de configuración propias:

- ***Gráficos circulares:*** opciones para configurar (i) en qué ángulo del círculo se debe empezar a dibujar la primera sección, (ii) en qué dirección se deben dibujar las secciones, (iii) el color del contorno del gráfico, (iv) si los colores se deben resaltar (en los gráficos circulares 3D puede elegir si se debe usar transparencia y sombra paralela), (v) si se deben dibujar etiquetas y (vi) si se deben añadir valores y porcentajes a las etiquetas y cuántos decimales deben llevar los porcentajes.
- ***Gráficos de barras:*** pestaña *General*: puede intercambiar el eje X por el eje Y (en los gráficos de barras 2D se genera un gráfico de barras horizontal); pestaña *Gráfico de barras:* puede configurar el contorno (y la sombra paralela en los gráficos de barras 2D); pestaña *Eje X*: puede configurar la etiqueta y el color del eje X y las líneas verticales de la cuadrícula; pestaña *Eje Y*: puede configurar la etiqueta y el color del eje Y, las líneas horizontales de la cuadrícula, el intervalo de valores que se debe mostrar y las marcas de graduación del eje Y; pestaña *Eje Z* (sólo en gráficos 3D): puede configurar la etiqueta y el color del eje Z; pestaña *Ángulos de visualización* (sólo en gráficos 3D): puede configurar la inclinación vertical, la rotación horizontal y el ancho del campo visual.
- ***Gráficos de líneas:*** pestaña *General*: puede intercambiar el eje X por el eje Y; pestaña *Gráfico de líneas*: puede incluir o no puntos de las líneas (formas de conexión); pestaña *Eje X*: puede configurar la etiqueta y el color del eje X y las líneas verticales de la cuadrícula; pestaña *Eje Y*: puede configurar la etiqueta y el color del eje Y, las líneas horizontales de la cuadrícula, el intervalo de valores que se debe mostrar y las marcas de graduación del eje Y.
- ***Medidor:*** opciones para configurar (i) en qué angulo empieza el medidor y el ángulo de barrido de la escala (sólo en medidores radiales), (ii) el intervalo de valores que se debe mostrar, (iii) el intervalo y el

color de las marcas de graduación principales y secundarias y (iv) el color del medidor, del borde y del puntero.

5.12.8 Configuración y aspecto

Podemos distinguir entre varias opciones de configuración de los gráficos:

- Opciones de configuración básicas 372: la opción de configuración más básica es el tipo de gráfico. Para seleccionar el tipo de gráfico haga clic en el botón **Cambiar de tipo** de la barra de herramientas de la ventana Gráficos. Este botón abre el cuadro de diálogo Cambiar de tipo 372.

- Opciones de configuración avanzadas 378: sirven para cambiar el aspecto del gráfico (su título, leyenda, los colores, la fuente del texto, etc.). La configuración avanzada de un gráfico se define en el cuadro de diálogo Cambiar aspecto 378. Para abrir este cuadro de diálogo haga clic en el botón **Cambiar aspecto** de la barra de herramientas de la ventana Gráficos.

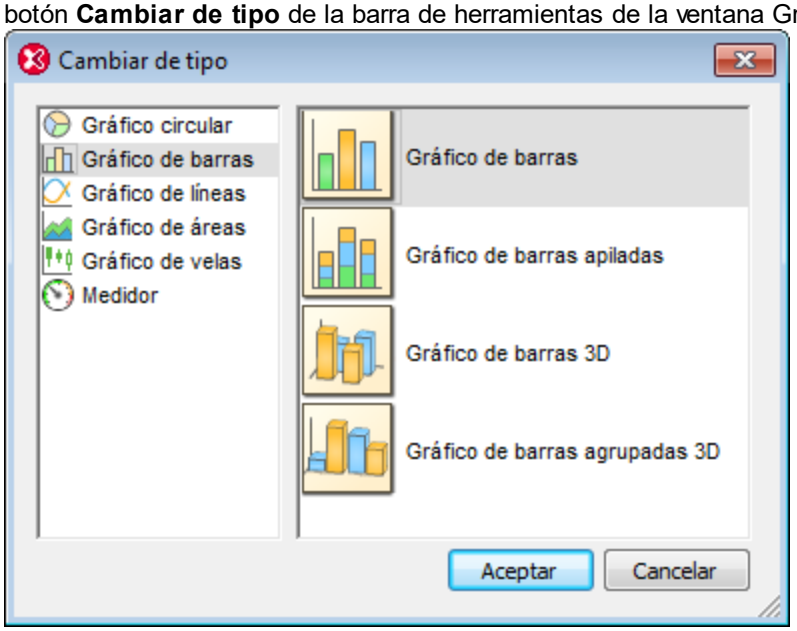
5.12.8.1 Configuración básica

Temas de este apartado:

- Definir el tipo de gráfico 372
- Tipos de gráfico compatibles 373
- Otras opciones de configuración básicas 378

Definir el tipo de gráfico

La opción de configuración más básica es el tipo de gráfico. Para seleccionar el tipo de gráfico haga clic en el botón **Cambiar de tipo** de la barra de herramientas de la ventana Gráficos.



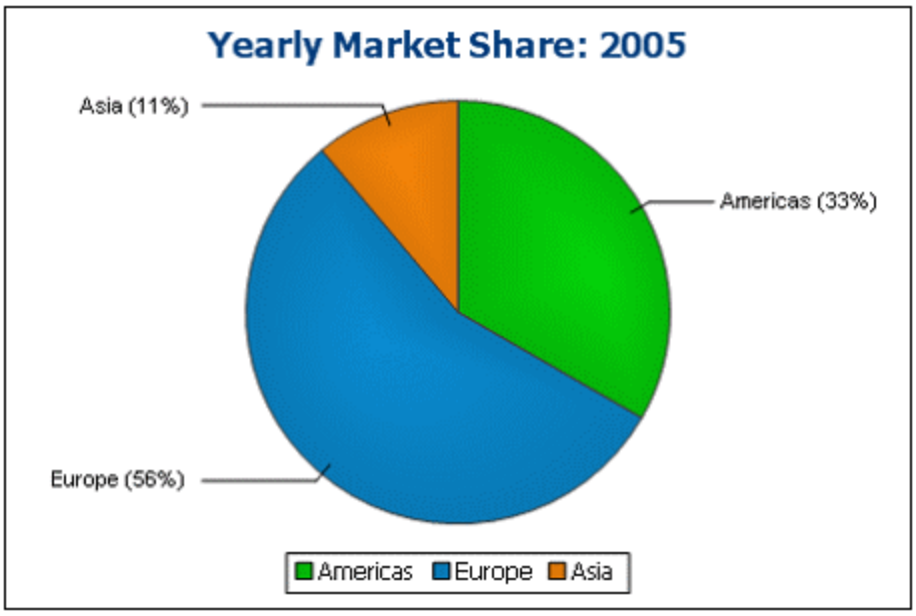
© 2016-2022 Altova GmbH

Tipos de gráficos

A continuación puede ver los tipos de gráfico compatibles con XMLSpy. En el cuadro de diálogo Cambiar de tipo [372] (*imagen anterior*), seleccione el tipo de gráfico y haga clic en **Aceptar**.

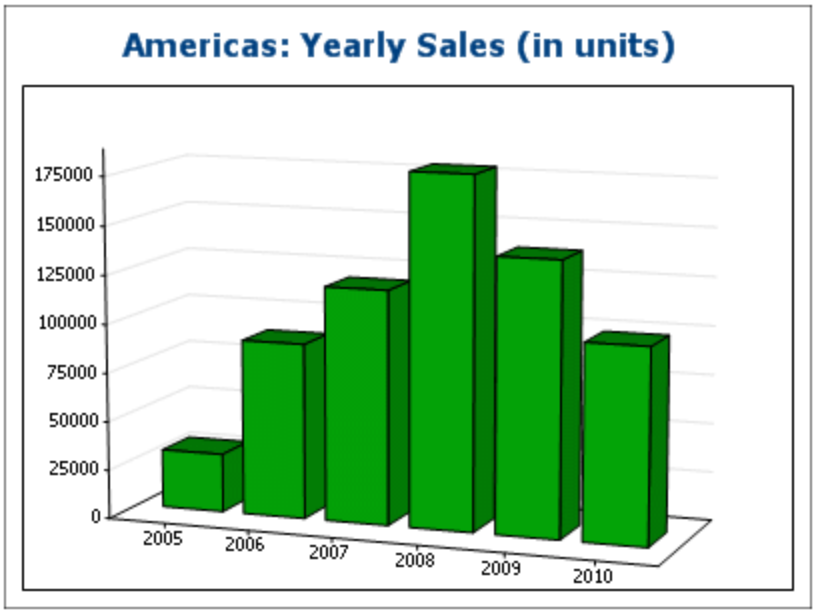
- *Gráficos circulares*

  En los gráficos circulares hay una columna/un eje que aporta los valores y otra columna/otro eje que ofrece las etiquetas para estos valores. La columna/el eje que tiene la etiqueta puede tomar valores no numéricos.

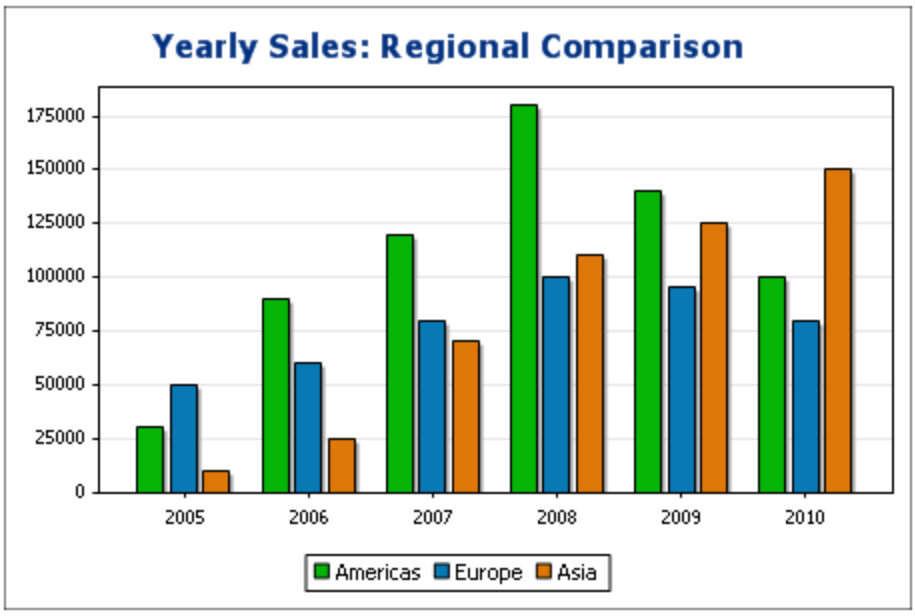


- *Gráficos de barras*

  Los gráficos de barras pueden tener dos conjuntos de valores en dos ejes.

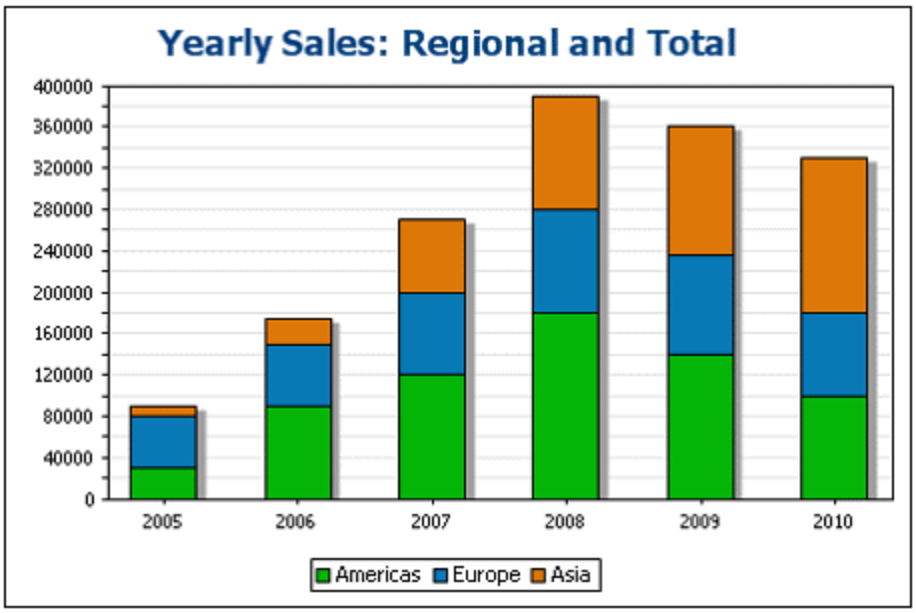


También pueden usar tres conjuntos de valores, como en el gráfico siguiente: (i) el continente, (ii) el año y (iii) el volumen de ventas. Los gráficos de barras pueden ser 2D (*abajo*) o 3D (*arriba*).



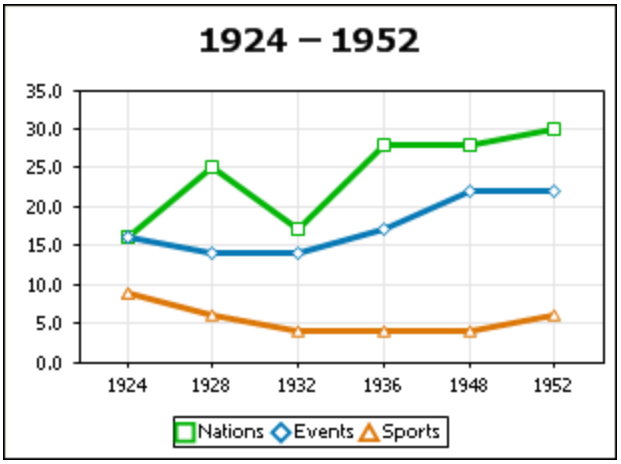
Los gráficos de barras de tres ejes también pueden tener los valores apilados para ver valores totales. Por ejemplo, compare el gráfico de barras apiladas que aparece a continuación con el gráfico anterior. Las barras apiladas muestran las ventas totales de todos los continentes.

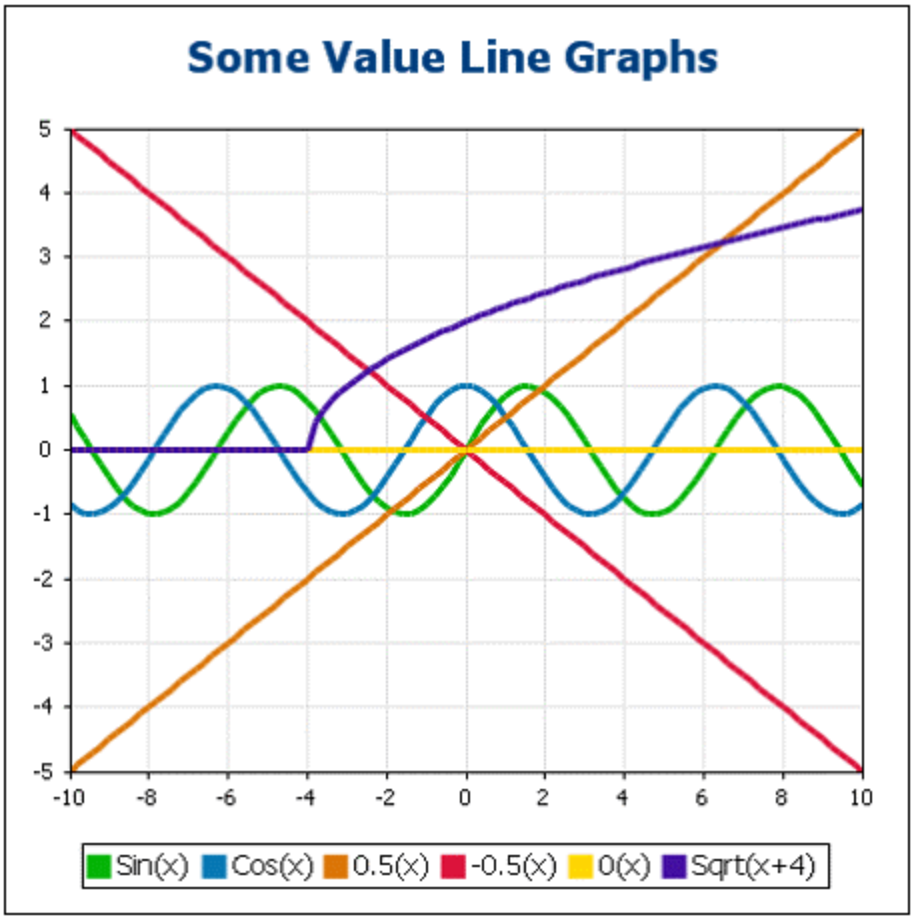
© 2016-2022 Altova GmbH



- *Gráficos de líneas*

  La diferencia entre un gráfico de líneas (*abajo a la izda*) y un gráfico de líneas de valores (*abajo a la dcha*) es que los gráficos de líneas de valores solamente toman valores numéricos para el eje X. Si quiere que el gráfico de líneas muestre valores de texto en el eje X, es mejor utilizar gráficos de líneas.

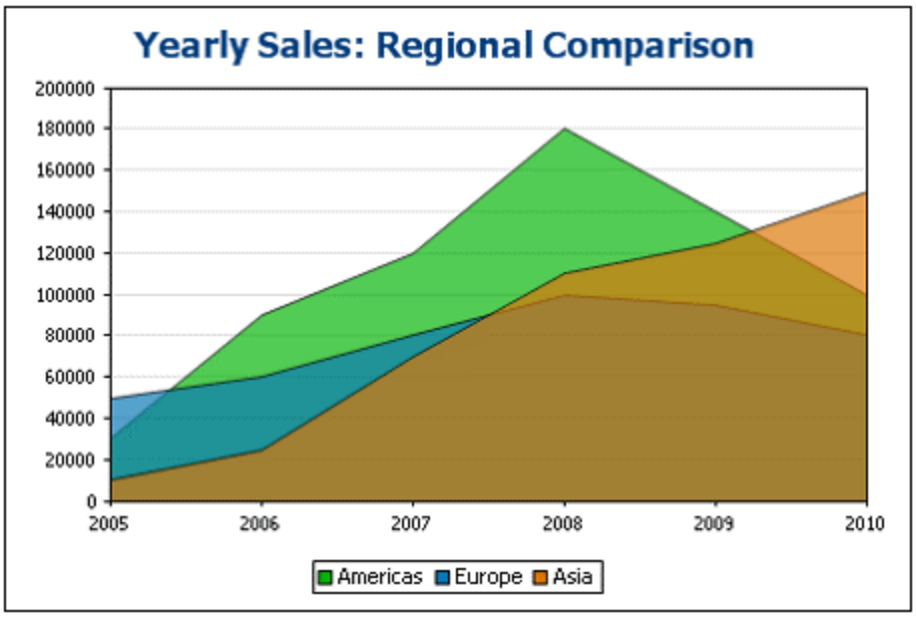




- *Gráficos de áreas*

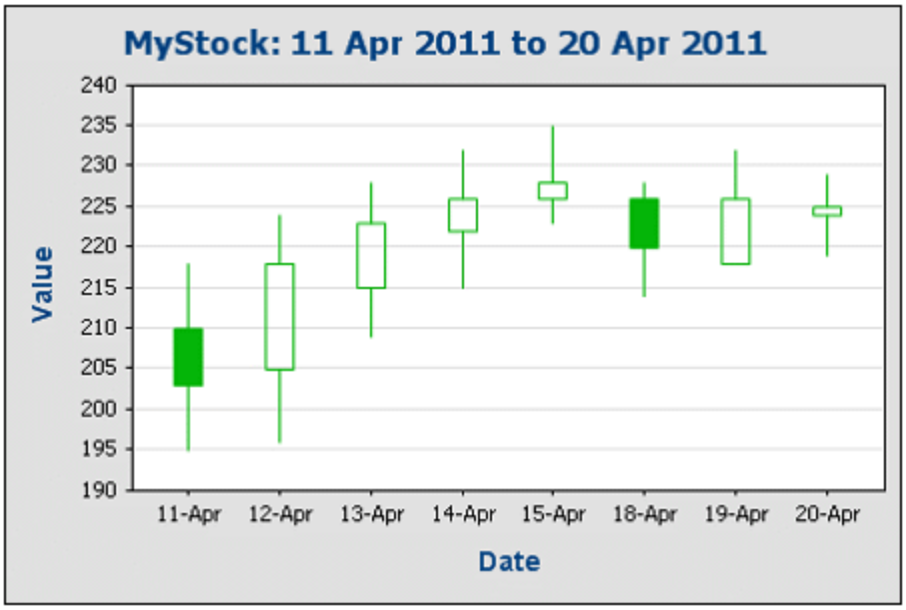
  Los gráficos de áreas son un tipo de gráfico de líneas en el que se colorean las áreas situadas bajo las líneas. También hay gráficos de áreas apiladas.

© 2016-2022 Altova GmbH



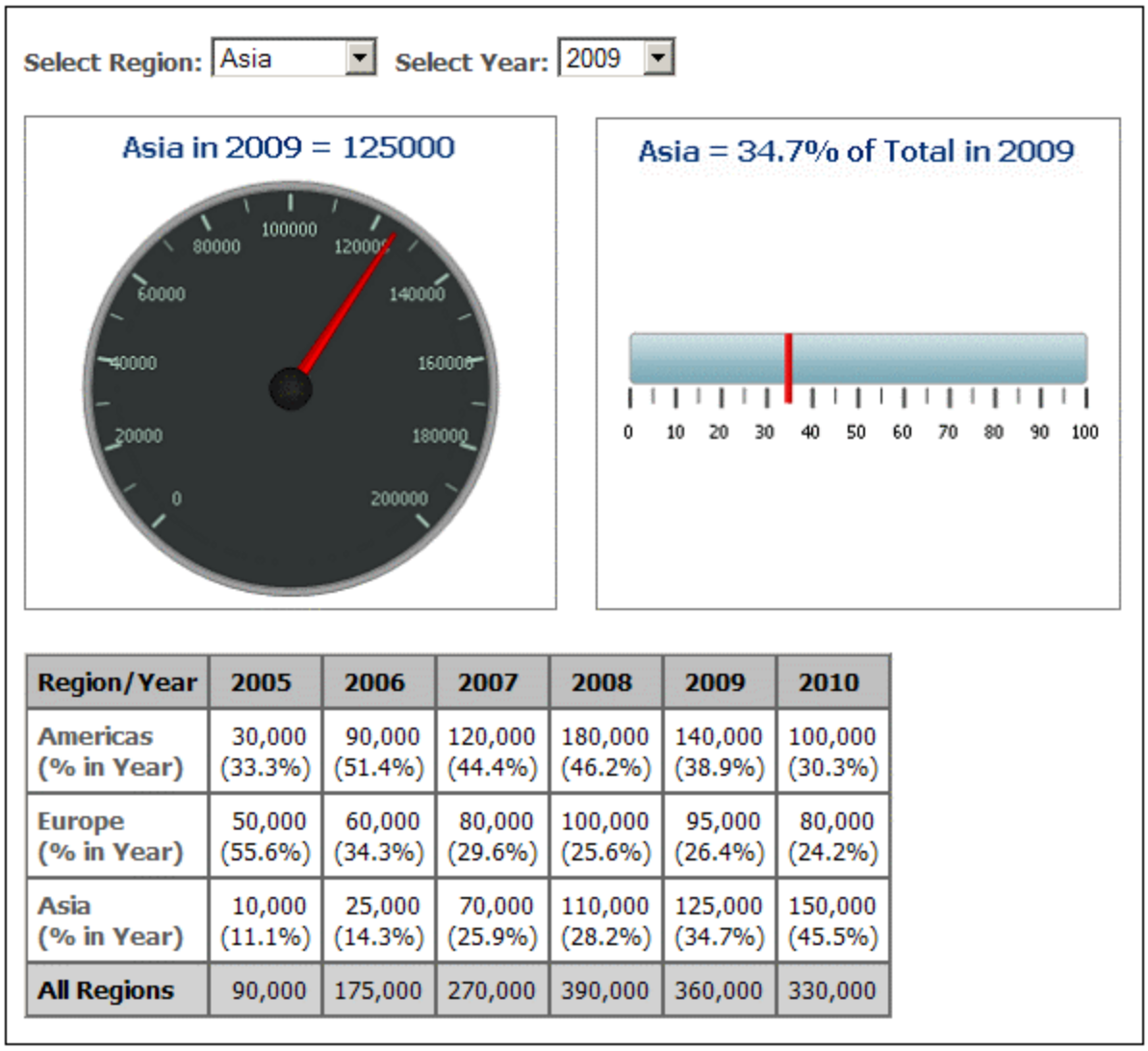
- *Gráficos de velas*

  Los gráficos de velas sirven para representar las oscilaciones del precio de valores, materias primas, divisas, etc. durante un período de tiempo. El gráfico no solo muestra el desarrollo de los precios durante un período, sino que además muestra el precio de apertura (opcional), el precio de cierre, el precio más bajo y el precio más alto. El eje Y toma tres o cuatro series (cierre (opcional), alto, bajo y apertura). A continuación puede ver un gráfico de velas con cuatro series.



▼ *Medidores*

Los medidores sirven para representar un solo valor y su relación con un valor máximo y un valor mínimo.



Region/Year	2005	2006	2007	2008	2009	2010
Americas (% in Year)	30,000 (33.3%)	90,000 (51.4%)	120,000 (44.4%)	180,000 (46.2%)	140,000 (38.9%)	100,000 (30.3%)
Europe (% in Year)	50,000 (55.6%)	60,000 (34.3%)	80,000 (29.6%)	100,000 (25.6%)	95,000 (26.4%)	80,000 (24.2%)
Asia (% in Year)	10,000 (11.1%)	25,000 (14.3%)	70,000 (25.9%)	110,000 (28.2%)	125,000 (34.7%)	150,000 (45.5%)
All Regions	90,000	175,000	270,000	390,000	360,000	330,000

Otras opciones de configuración básicas

En el panel *Configuración del gráfico* también puede definir el título del gráfico (*imagen siguiente*).

5.12.8.2 Configuración avanzada

Temas de este apartado:

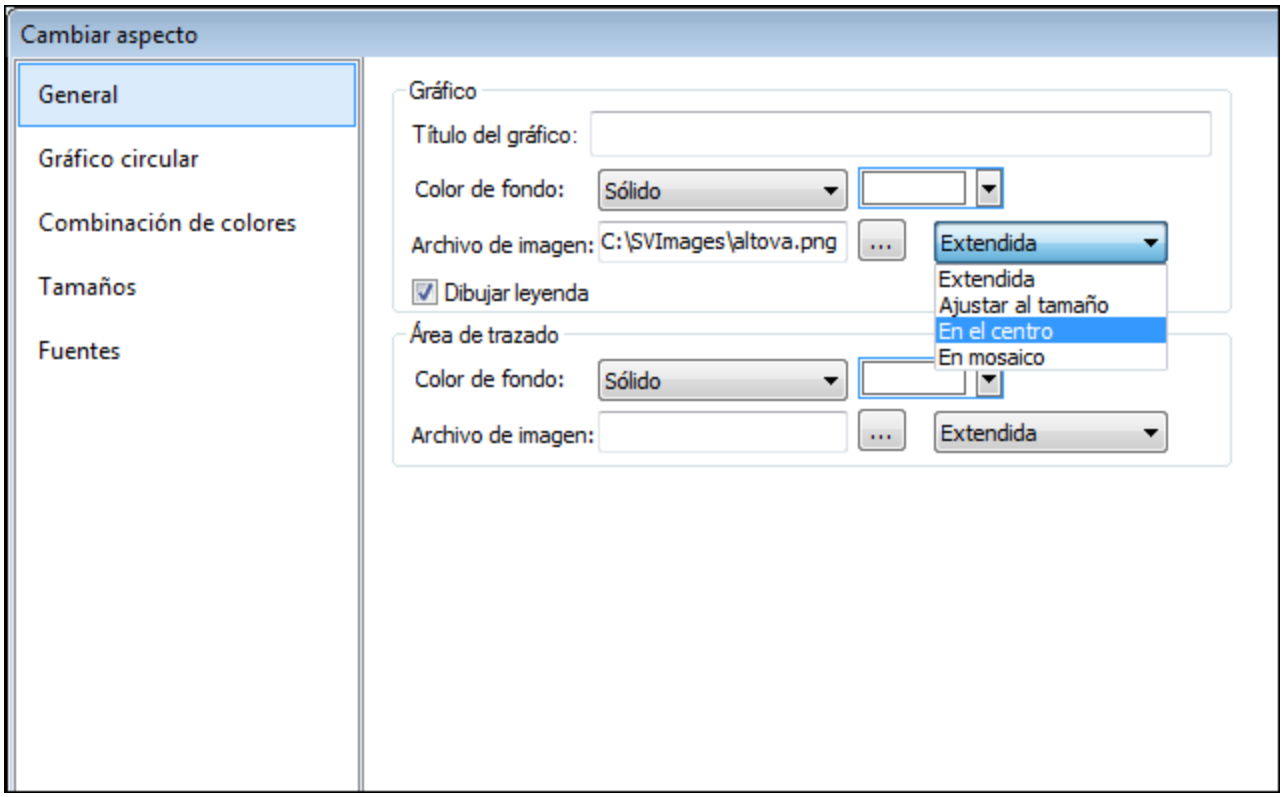
- Definir opciones de configuración avanzadas 379
- Resumen de opciones de configuración avanzadas 379

© 2016-2022 Altova GmbH

- Cargar, guardar y restaurar la configuración de un gráfico 382

Definir opciones de configuración avanzadas

Para definir las opciones de configuración avanzadas haga clic en el botón **Cambiar aspecto** de la barra de herramientas de la ventana Gráficos. Esto abre el cuadro de diálogo "Cambiar aspecto" propio del tipo de gráfico con el que está trabajando. Por ejemplo, la imagen siguiente muestra el cuadro de diálogo "Cambiar aspecto" de un gráfico circular.



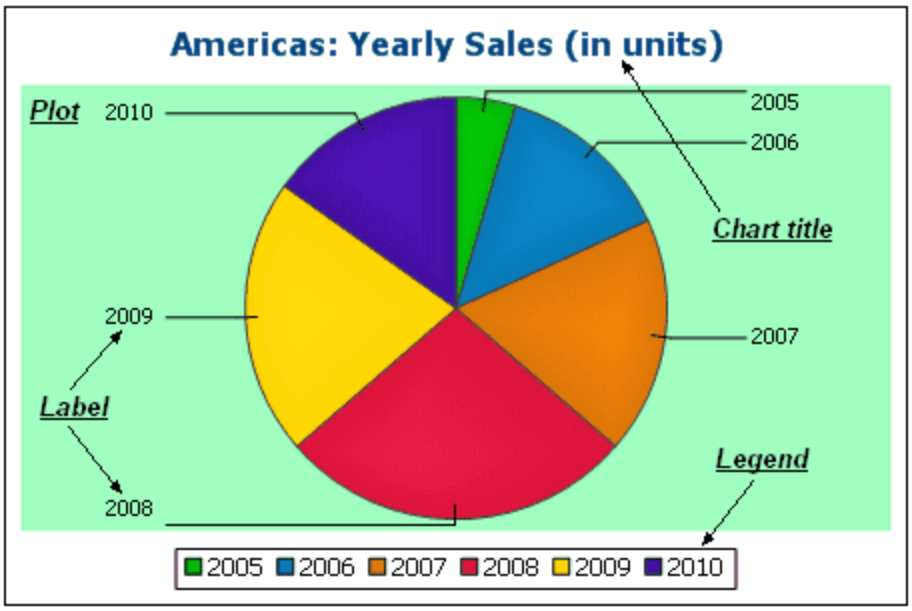
Resumen de opciones de configuración avanzadas

En el cuadro de diálogo "Cambiar aspecto" las opciones de configuración se dividen en pestañas, algunas comunes a todos los tipos de gráfico y otras propias de cada tipo.

Opciones de configuración comunes

▼ *General*

Aquí puede definir el título del gráfico (*imagen siguiente*), el color de fondo del gráfico y del área de trazado. En la imagen siguiente, por ejemplo, el área de trazado es de color verde pálido, mientras que el gráfico en sí tiene el color de fondo blanco. También puede elegir una imagen de fondo para el gráfico o para el área de trazado. La imagen se puede estirar para abarcar todo el gráfico o todo el área de trazado. También se puede ajustar al tamaño del gráfico o del área de trazado o se puede colocar en el centro o en mosaico. La leyenda es la clave de los códigos de colores del gráfico y se puede activar o desactivar.



- *Combinación de colores*

  Puede elegir entre cuatro combinaciones de colores predefinidas o definir combinaciones nuevas. También puede modificar las combinaciones de colores predefinidas añadiendo o eliminando colores. La combinación de colores seleccionada en esta pestaña es la que se utiliza para el gráfico.

- *Tamaños*

  Aquí puede configurar el tamaño de los diferentes componentes del gráfico, tanto en píxeles como en una relación porcentual.

- *Fuentes*

  Aquí puede definir las propiedades de las fuentes del título del gráfico, de las leyendas y de las etiquetas. El tamaño de las fuentes se puede indicar como porcentaje del tamaño del gráfico o con un valor absoluto en puntos.

- *Botón Cargar/Guardar*

  La configuración elegida se puede guardar en un archivo XML o puede cargar una configuración definida en un archivo XML, siempre y cuando el archivo tenga la estructura correcta. Para ver cómo debe ser la estructura, guarde la configuración de un gráfico y abra el archivo XML. Este botón también sirve para restaurar la configuración predeterminada del gráfico.

Opciones propias de cada tipo de gráfico

- *Gráficos circulares*

© 2016-2022 Altova GmbH

En la pestaña *Gráfico circular* puede definir (i) con qué angulo se dibuja la primera sección del gráfico circular; (ii) en qué dirección se dibujan las secciones del gráfico; (iii) el color del contorno del gráfico; (iv) si se deben resaltar los colores; (v) si se dibujan etiquetas y (vi) si se añaden valores y porcentajes a las etiquetas y cuántos decimales llevan los porcentajes.

Gráficos de barras

Las opciones propias de los gráficos de barras son: (en la pestaña *General*) puede intercambiar el eje X y el eje Y para generar un gráfico de barras horizontales; (en la pestaña *Gráfico de barras*) puede dibujar el contorno y sombras paralelas (solo en gráficos de barras 2D); (en la pestaña *Eje Y*) puede definir la etiqueta y el color del eje Y, líneas de cuadrícula horizontales, el rango de valores y las marcas de graduación del eje Y; (en la pestaña *Eje Z* de los gráficos de barras 3D) puede elegir la etiqueta y el color del eje Z; (en la pestaña *Ángulos de visualización 3D*) puede elegir la inclinación vertical, la rotación horizontal y el ancho de la vista.

Gráficos de líneas

Las opciones propias de los gráficos de líneas son: (en la pestaña *General*) puede intercambiar el eje X y el eje Y; (en la pestaña *Gráfico de líneas*) puede elegir si se incluyen los puntos del área de trazado; (en la pestaña *Eje X*) puede definir la etiqueta y el color del eje X y las líneas de cuadrícula verticales; (en la pestaña *Eje Y*) puede elegir la etiqueta y el color del eje Y, las líneas de cuadrícula horizontales, el rango de valores y las marcas de graduación del eje Y.

Medidores

En la pestaña *Medidor* puede definir (i) el ángulo en el que empieza el medidor y el barrido de la escala; (ii) el rango de valores; (iii) el intervalo y el color de las marcas de graduación principales y secundarias; (iv) los colores del puntero, de la cabeza del puntero y del borde del medidor.

Gráficos de áreas

La transparencia de las áreas pueden tener un valor comprendido entre o (no transparentes) hasta 255 (totalmente transparentes). En el caso de los gráficos de áreas no apiladas, la transparencia hace que las partes de las áreas que están debajo de otras áreas se puedan ver. También puede dibujar el contorno de las áreas.

Gráficos de velas

Puede elegir el color de relleno de las velas para estos dos casos: (i) cuando el valor de cierre es mayor que el valor de apertura y (ii) cuando el valor de apertura es mayor que el valor de cierre. En este último caso también puede usar el color de la serie. El color de la serie se especifica en la pestaña *Combinación de colores* del cuadro de diálogo "Cambiar aspecto".

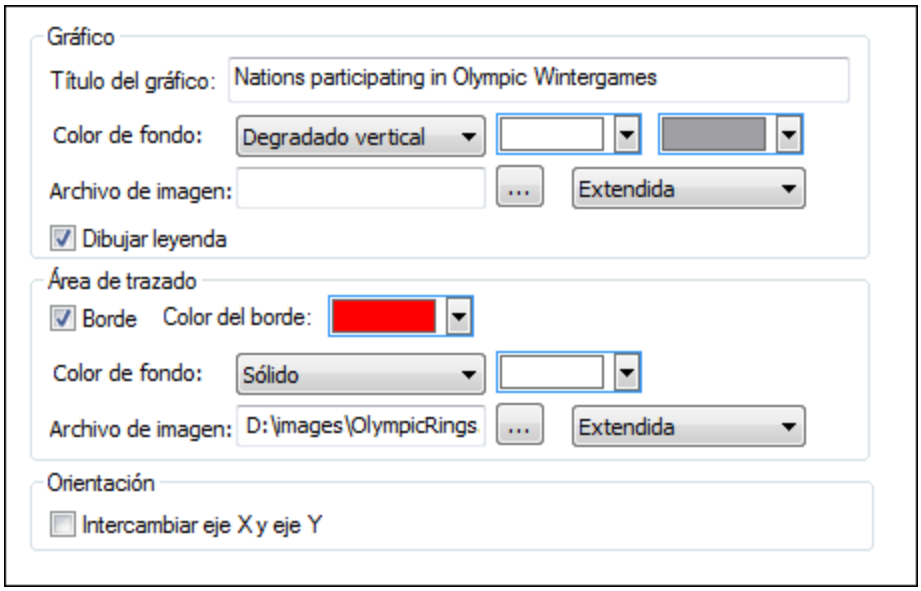
Cargar, guardar y restaurar la configuración del gráfico

La configuración del gráfico elegida por el usuario se puede guardar en un archivo XML. Esta configuración se puede cargar más tarde como configuración de otro gráfico. Estas son las opciones que aparecen al hacer clic en el botón **Cargar/guardar** (*ver imagen* 378) del cuadro de diálogo "Cambiar aspecto":

- **Restaurar valores predeterminados**: haga clic en esta opción para rechazar los cambios realizados en el cuadro de diálogo y recuperar la configuración predeterminada de todas las pestañas.
- **Cargar desde el archivo...**: haga clic en esta opción para importar una configuración guardada previamente en un archivo XML. El comando abre el cuadro de diálogo "Abrir", donde puede elegir el archivo.
- **Guardar en el archivo...**: esta opción abre el cuadro de diálogo "Guardar como". Aquí puede indicar en qué archivo XML se guarda la configuración. El archivo tendrá las opciones de configuración que no son opciones de configuración predeterminadas.

5.12.8.2.1 Generales

En la sección **General** del cuadro de diálogo "Cambiar aspecto" puede definir el título del gráfico, añadir o quitar una leyenda y definir imágenes y colores de fondo. Además, para los gráficos de barras, líneas, áreas y velas, también puede definir la orientación del gráfico.



Gráfico

Escriba un título descriptivo para el gráfico en el campo *Título del gráfico* y seleccione un color de fondo para todo el gráfico en la lista desplegable. Puede elegir entre un fondo sólido, con degradado vertical o con degradado horizontal. Si usa un fondo degradado, también puede definir el color inicial del degradado y el color final. Además, puede usar una imagen de fondo y definir una de estas opciones de presentación para la imagen:

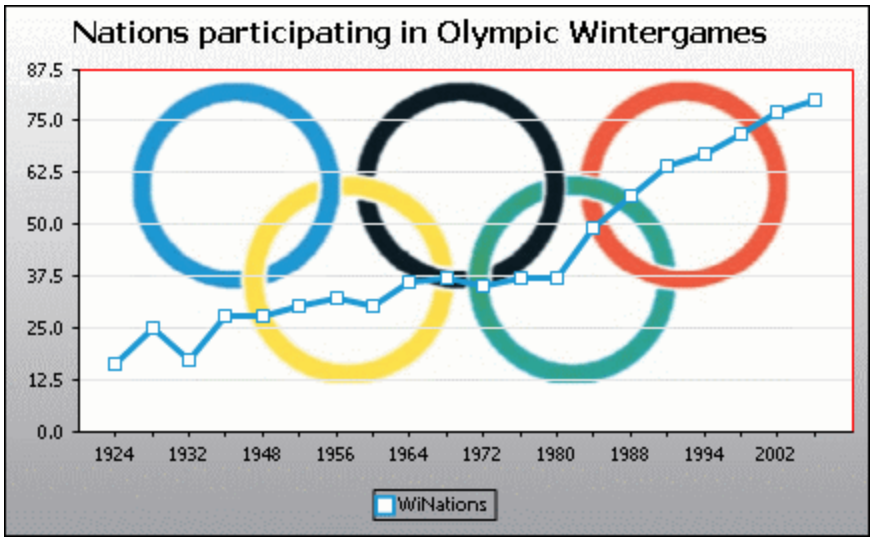
© 2016-2022 Altova GmbH

- Extendida: la imagen se estira para ocupar toda la superficie del gráfico
- Ajustar al tamaño: la imagen se ajusta al marco del gráfico y se conserva la relación de aspecto de la imagen
- En el centro: la imagen se muestra en el centro del gráfico en su tamaño original
- En mosaico: si el tamaño de la imagen es menor que el tamaño del gráfico, se crea un mosaico con copias de la imagen y el mosaico ocupa toda la superficie del gráfico

La casilla *Dibujar leyenda* está marcada por defecto. Si no desea incluir una leyenda en el gráfico, desactive esta casilla.

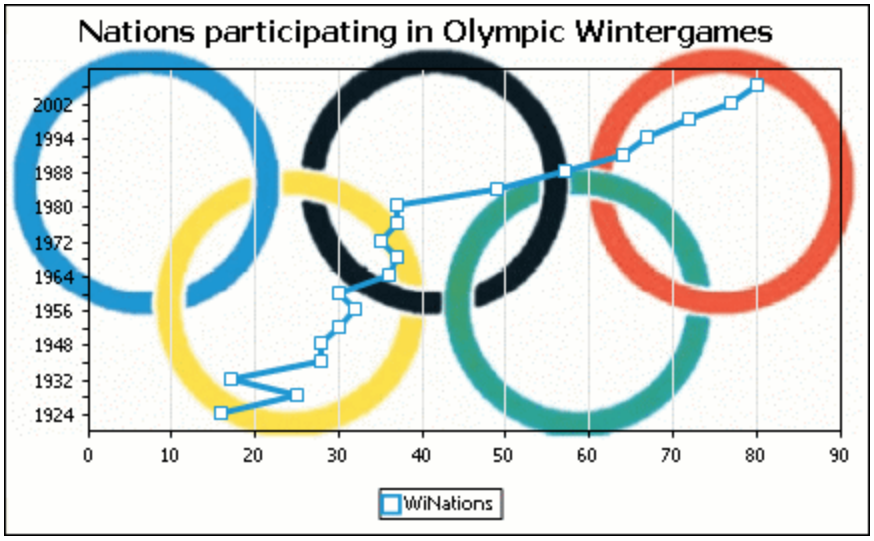
Área de trazado

El área de trazado es la superficie donde se representan los datos del gráfico. Puede dibujar un borde alrededor del área de trazado y especificar un color y una imagen de fondo distintos para el área de trazado. En la imagen siguiente, el gráfico tiene un color de fondo gris con un degradado vertical, mientras que el área de trazado es de color blanco, tiene un borde de color rojo y tiene una imagen de fondo.



Orientación

Si tiene una serie pequeña de valores grandes, quizás sea mejor intercambiar el eje X con el eje Y para obtener una representación más clara de los datos (*ver imagen siguiente*). Observe que en la imagen siguiente, se definió una imagen de fondo para todo el gráfico y el color de fondo del área de trazado es "transparente" en lugar de sólido como en el gráfico anterior.



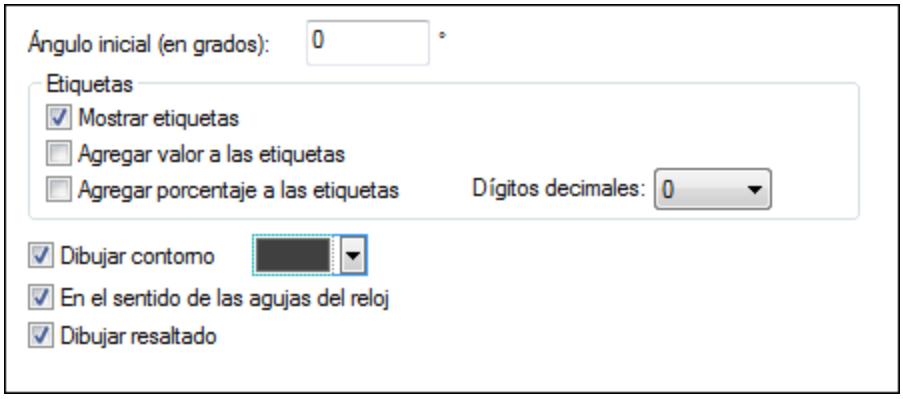
Cabe mencionar que en los gráficos circulares y en los medidores no es posible intercambiar el eje X con el eje Y. Por tanto, esta opción no está disponible para ese tipo de gráficos.

5.12.8.2.2 Opciones propias de cada tipo de gráfico

Dependiendo del tipo de gráfico, e incluso del subtipo de gráfico, el cuadro de diálogo "Cambiar aspecto" ofrece una sección adicional donde se pueden definir las características propias del tipo de gráfico seleccionado.

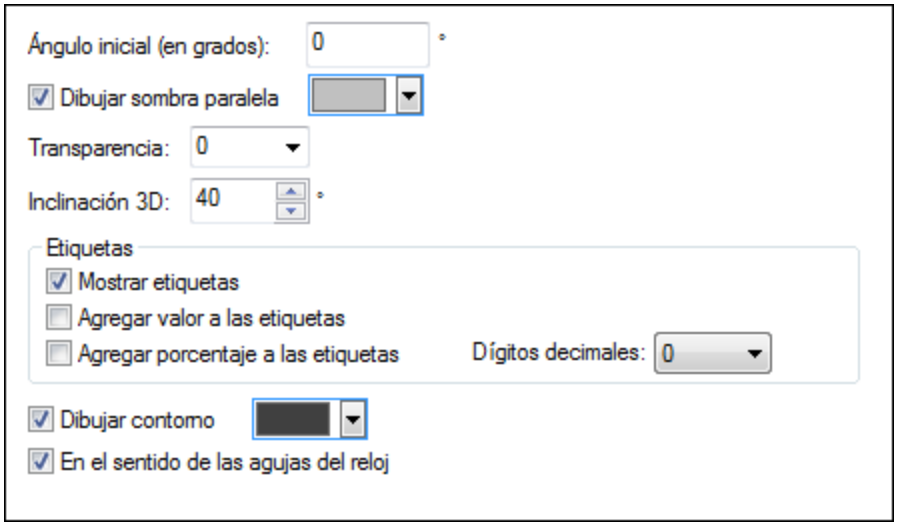
Gráficos circulares

La mayoría de las opciones de configuración son las mismas para la versión 2D y la versión 3D de los gráficos circulares. En los gráficos circulares 2D también puede dibujar resaltados.



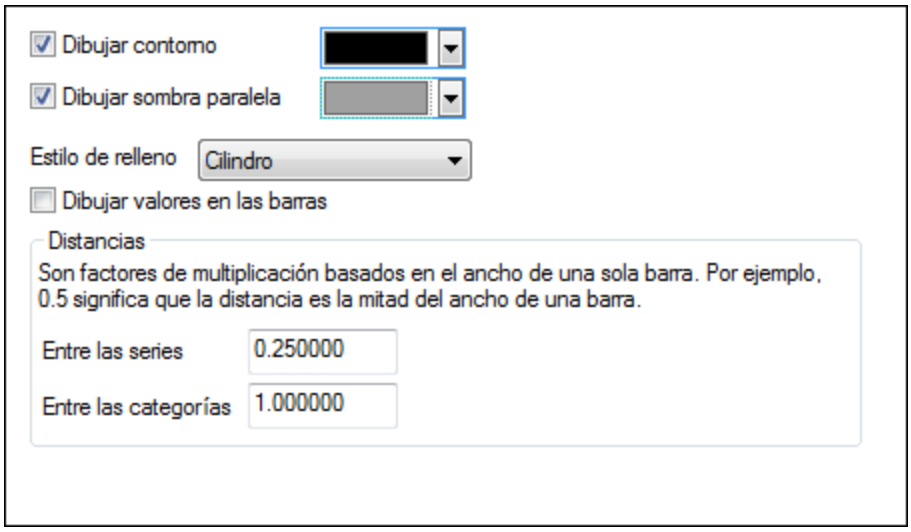
En los gráficos circulares 3D puede dibujar sombras paralelas, añadir transparencia y definir la inclinación 3D del gráfico.

© 2016-2022 Altova GmbH



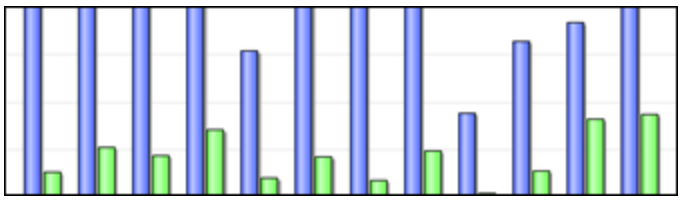
El valor *Ángulo inicial* define en qué posición se dibujará la primera sección del gráfico. Un ángulo de 0 grados equivale a las 12 en punto de un reloj.
Además de la leyenda puede incluir etiquetas en el gráfico, añadir valores y porcentajes a las etiquetas y definir el número de decimales que puede llevar el porcentaje.
Para dibujar un contorno alrededor del gráfico y de cada sección del gráfico, marque la casilla *Dibujar contorno*. También puede seleccionar el color del contorno. La casilla *En el sentido de las agujas del reloj* permite especificar si las secciones del gráfico se deben representar en el sentido de las agujas del reloj o en el sentido contrario.
En los gráficos circulares 3D puede dibujar una sombra paralela y definir su color, añadir transparencia al gráfico y definir su inclinación 3D. En los gráficos circulares 2D, la opción *Dibujar resaltado* permite añadir una estructura adicional al gráfico.

Gráficos de barras



En los gráficos de barras están disponibles estas opciones de configuración:

- Puede añadir un contorno a las barras y definir el color de dicho contorno.
- En los gráficos de barras 2D, también puede dibujar una sombra paralela y definir su color (esta opción no está disponible para gráficos de barras 3D).
- Por defecto, las barras tienen forma de cilindro, pero también puede elegir la opción *Degradado vertical* o *Sólido* de la lista desplegable *Estilo de relleno* (esta opción solo está disponible para gráficos de barras 2D).
- En cada barra puede dibujar sus valores (que corresponden a la altura que tiene una barra en el eje Y). La fuente de los valores se puede especificar en el grupo de opciones *Fuentes* (esta opción solo está disponible para gráficos de barras 2D).
- La distancia que separa las series de barras y los grupos de barras se puede especificar como fracción decimal del ancho de una barra. Por ejemplo, en la imagen siguiente puede ber grupos de barras formados por una serie azul y una serie verde. La distancia que separa las series se estableció en un 25% (`=0.25`) del ancho de cada barra. La distancia que separa los grupos de barras se estableció en un 100% (`=1.0`) del ancho de cada barra.



Gráficos de líneas



Para dibujar formas de conexión que marquen los valores en los gráficos de líneas, es necesario activar como mínimo una casilla del cuadro de grupo *Dibujar formas de conexión*. Hay cinco tipos de formas de conexión diferentes, una para cada serie: cuadrados, rombos, triángulos, triángulos invertidos y círculos. Si su gráfico contiene más de cinco series, puede combinar las formas de conexión para crear más tipos de formas. Para ello, seleccione más de una opción en el cuadro de grupo "Dibujar formas de conexión". Para la imagen siguiente se seleccionaron las opciones *Con relleno* y *Con barra diagonal*. Es decir, a partir de la sexta serie, se utiliza el tipo de forma de conexión *Con barra diagonal*.

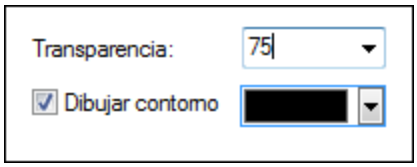
La opción *Dibujar línea* sirve para establecer si el gráfico se dibuja (i) con formas de conexión solamente o (ii) con formas de conexión unidas por medio de una línea.



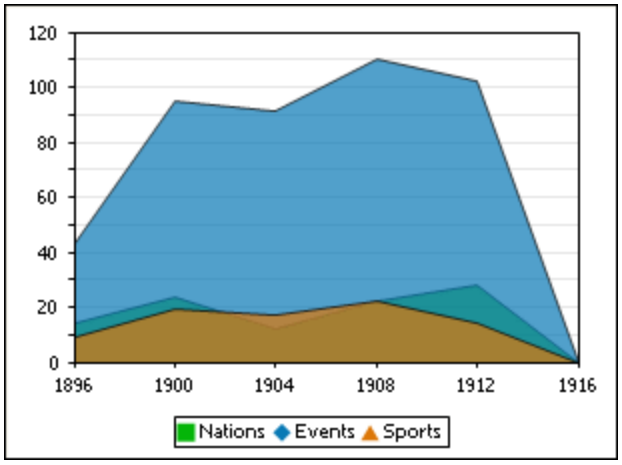
Las formas de conexión se pueden utilizar tanto en gráficos de líneas como en gráficos de líneas de valor.

© 2016-2022 Altova GmbH

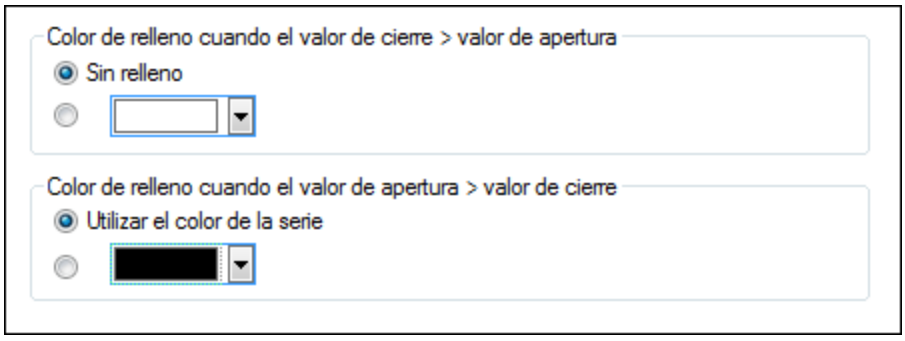
Gráficos de áreas



En los gráficos de áreas puede cambiar la transparencia. Así puede evitar que una serie quede oculta por otra serie del gráfico. Además, puede añadir un contorno a cada área de datos y definir el color de dicho contorno (*imagen siguiente*).



Gráficos de velas



Si tanto el valor de cierre como el de apertura se definen como serie, puede elegir el color de la serie y si la vela se debe rellenar o no cuando el valor de cierre sea superior al valor de apertura.

Medidores



En el cuadro de grupo *Ángulos* el valor *Inicio* define la posición de la marca 0 y el valor *Barrido* equivale al ángulo utilizado para la representación del medidor. En el cuadro de grupo *Intervalo de valor* puede definir el valor mínimo y el valor máximo que desea representar en el medidor.
Las marcas de graduación se muestran con (marcas de graduación principales) o sin (marcas de graduación secundarias) el valor correspondiente. Puede definir un color para las marcas de graduación principales y otro para las marcas de graduación secundarias. En el cuadro de grupo *Colores* puede definir el color de relleno del medidor, el color del puntero, el color de la cabeza del puntero (oculta el inicio del puntero en el centro del medidor) y el color del borde que rodea el medidor. Además, puede presentar en cualquier ángulo del medidor el valor actual y una etiqueta adicional.

5.12.8.2.3 Colores

Dependiendo del tipo de gráfico seleccionado, XMLSpy ofrece dos secciones diferentes para definir los colores que se usarán en el gráfico:

- **Combinación de colores** en caso de usar gráficos circulares, de barras, de líneas, de áreas y de velas
- **Intervalo de colores** en caso de usar medidores

© 2016-2022 Altova GmbH

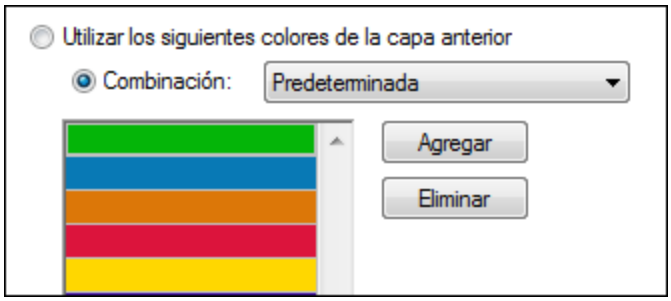
Combinación de colores

La sección **Combinación de colores** del cuadro de diálogo "Cambiar aspecto" ofrece cuatro combinaciones de colores predeterminadas (es decir, predeterminada, escala de grises, multicolor y tonos pasteles). Estas combinaciones se pueden personalizar y, si lo prefiere, puede crear desde cero su propia combinación de colores.



El primer color de la lista se usará para la primera serie, el segundo color para la segunda serie y así sucesivamente. Para cambiar el orden de los colores basta con seleccionar un color y arrastrarlo a una posición nueva. Para añadir un color nuevo o eliminar un color no deseado, haga clic en el botón correspondiente. En los gráficos de velas se utiliza sólo el primer color de la lista.

Si está trabajando con un gráfico multicapa en una ventana de gráficos, la sección **Combinación de colores** del cuadro de diálogo "Cambiar aspecto" contiene un botón de opción adicional llamado *Utilizar los siguientes colores de la capa anterior*, que está activado por defecto.



Si este botón de opción está activado, se usará la combinación de colores de la capa anterior y no podrá elegir una combinación de colores diferente para el gráfico multicapa. Las series de la capa activa se dibujarán con

los siguientes colores de la combinación de colores de la capa anterior. De esta manera todas las series de la ventana de gráficos tendrán colores diferentes y se podrán distinguir más fácilmente.

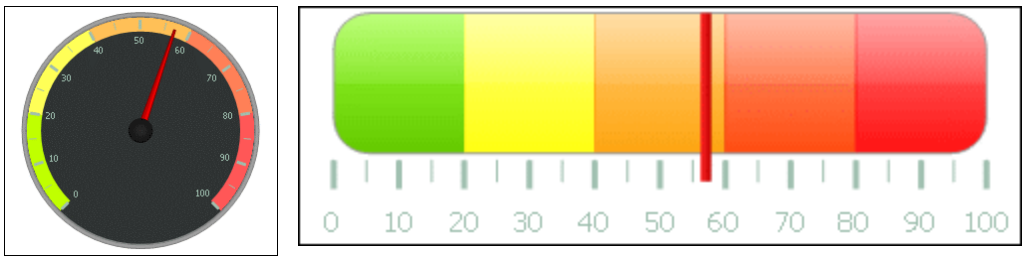
Para evitarlo, en la siguiente capa que añada, puede elegir una combinación de colores distinta que se podrá reutilizar en capas posteriores.

Intervalo de colores

En los medidores puede personalizar el aspecto del medidor con solo aplicar colores a determinados intervalos de valores.



Las opciones de configuración de la imagen anterior darían lugar a estos medidores:

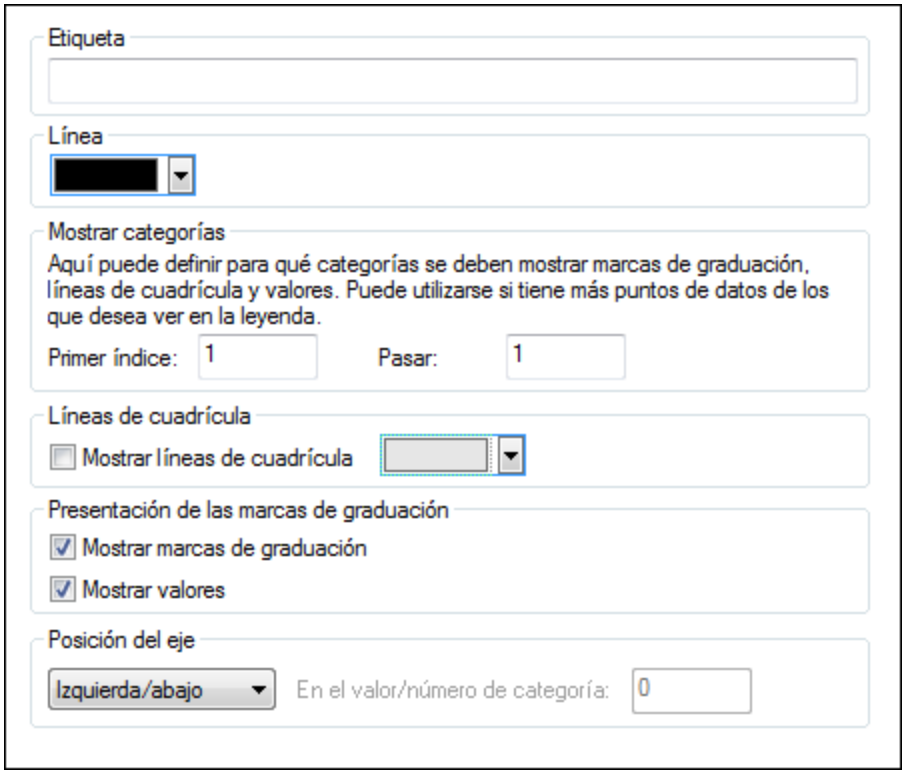


5.12.8.2.4 Eje X

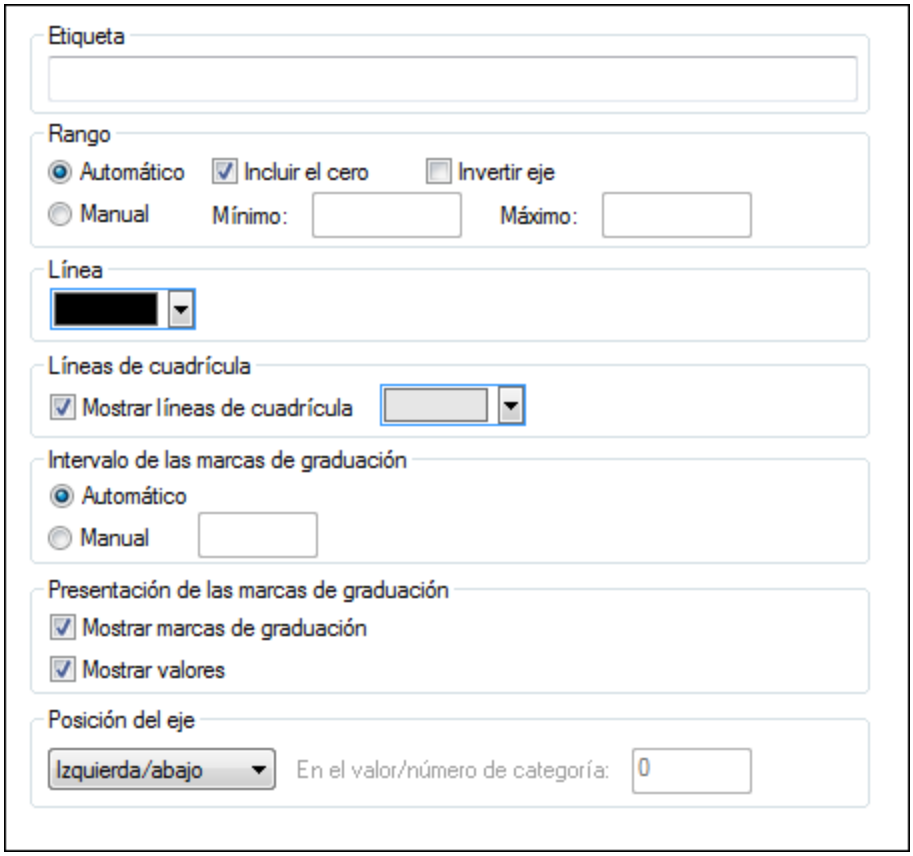
En la sección **Eje X** del cuadro de diálogo "Cambiar aspecto" puede escribir la etiqueta del eje, definir el color de la línea del eje y las líneas de la cuadrícula (si no se oculta la cuadrícula). También puede definir si se muestran o no las marcas de graduación y los valores del eje. Esta sección es idéntica para los gráficos de barras, líneas y velas.

Las opciones del grupo *Mostrar categorías* sirven para especificar si solamente se presenta un subconjunto de las categorías (valores del eje X), es decir, si solamente se presentan las marcas de graduación, las líneas de cuadrícula y los valores de las categorías seleccionadas. Para crear el subconjunto de las categorías que se debe presentar basta con (i) introducir el índice del primer valor que se debe mostrar e (ii) introducir el número de índices que se deben recorrer. Por ejemplo, si hay 101 categorías (`1900, 1901, 1902 ... 1999, 2000`), entonces puede presentar cada diez años empezando desde `1900` y terminando en el año `2000` si elige el valor `1` para la opción *Primer índice* y el valor `10` para la opción *Pasar*.

© 2016-2022 Altova GmbH



Sin embargo, en los gráficos de líneas de valor también puede definir el rango de valores y definir con qué intervalo se deben mostrar las marcas de graduación.



Etiqueta
El texto que escriba en el campo Etiqueta se imprimirá debajo el eje X, a modo de descripción.

Rango
El botón de opción *Automático* está seleccionado por defecto en el cuadro de grupo *Rango*. Si desea mostrar parte del gráfico con más detalle, seleccione el botón de opción *Manual* e introduzca el valor mínimo y el valor máximo en los campos correspondientes.

Si la columna que se usa para el eje X no incluye el valor 0, puede desactivar la casilla *Incluir el cero* y el eje X empezará con el valor mínimo de la serie. La opción Invertir eje permite invertir los valores del eje X. Por ejemplo, si los valores van del `0` a `360` y selecciona esta opción, el eje X se generará con el valor `360` en la base del eje y los valores irán descendiendo hasta el `0` a medida que asciende el eje X.

Línea
El eje se representa en el color elegido en la lista desplegable *Línea*. Puede usar uno de los colores predeterminados o hacer clic en el botón **Otro color...** para elegir un color estándar o definir un color personalizado. Haga clic en el botón **Seleccionar...** de la pestaña *Personalizado* y con la pipeta seleccione cualquier color de su pantalla.

Líneas de cuadrícula
Si marca la casilla *Mostrar líneas de cuadrícula*, puede seleccionar un color de la lista desplegable para las líneas de la cuadrícula.

© 2016-2022 Altova GmbH

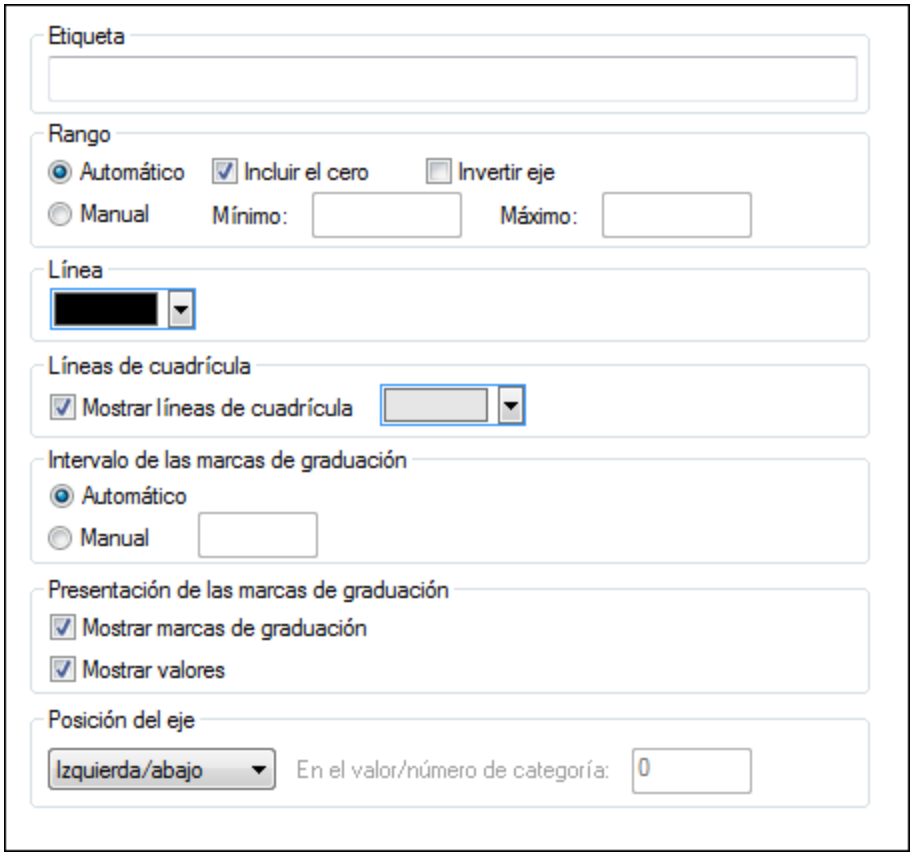
Intervalo de las marcas de graduación
Si no está satisfecho con las marcas de graduación predeterminadas, puede activar el botón de acción *Manual* del cuadro de grupo *Intervalo de las marcas de graduación* y escribir la diferencia entre las marcas de graduación en el campo correspondiente.

Presentación de las marcas de graduación
En este cuadro de grupo puede activar o desactivar las marcas de graduación y los valores del eje.

Posición del eje
Elija la posición del eje de la lista desplegable. Si selecciona la opción *En el valor/número de categoría* también puede especificar la posición concreta del eje.

5.12.8.2.5 Eje Y

En la sección **Eje Y** del cuadro de diálogo "Cambiar aspecto" puede escribir la etiqueta del eje, definir el color de la línea del eje y las líneas de la cuadrícula (si no se oculta la cuadrícula). También puede definir el rango de valores y decidir si se deben mostrar o no las marcas de graduación y en qué posición. Por último puede decidir si se deben mostrar o no los valores del eje. Esta sección del cuadro de diálogo es idéntica para los gráficos de barras y de líneas.



Etiqueta
El texto que escriba en el campo *Etiqueta* se imprimirá a la izquierda del eje Y, a modo de descripción.

Rango
El botón de opción *Automático* está seleccionado por defecto en el cuadro de grupo *Rango*. Si desea mostrar parte del gráfico con más detalle, seleccione el botón de opción *Manual* e introduzca el valor mínimo y el valor máximo en los campos correspondientes.

Si la columna que se usa para el eje Y no incluye el valor 0, puede desactivar la casilla *Incluir el cero* y el eje Y empezará con el valor mínimo de la serie. La opción Invertir eje permite invertir los valores del eje Y. Por ejemplo, si los valores van del `0` a `360` y selecciona esta opción, el eje Y se generará con el valor `360` en la base del eje y los valores irán descendiendo hasta el `0` a medida que asciende el eje Y.

Línea
El eje se representa en el color elegido en la lista desplegable *Línea*. Puede usar uno de los colores predeterminados o hacer clic en el botón **Otro color...** para elegir un color estándar o definir un color personalizado. Haga clic en el botón **Seleccionar...** de la pestaña *Personalizado* y con la pipeta seleccione cualquier color de su pantalla.

Líneas de cuadrícula
Si marca la casilla *Mostrar líneas de cuadrícula*, puede seleccionar un color de la lista desplegable para las líneas de la cuadrícula.

Intervalo de las marcas de graduación
Si no está satisfecho con las marcas de graduación predeterminadas, puede activar el botón de acción *Manual* del cuadro de grupo *Intervalo de las marcas de graduación* y escribir la diferencia entre las marcas de graduación en el campo correspondiente.

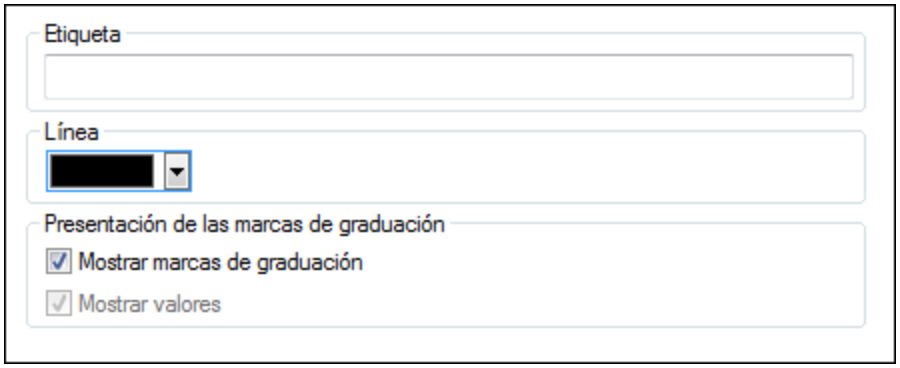
Presentación de las marcas de graduación
En este cuadro de grupo puede activar o desactivar las marcas de graduación y los valores del eje.

Posición del eje
Elija la posición del eje de la lista desplegable. Si selecciona la opción *En el valor/número de categoría* también puede especificar la posición concreta del eje.

5.12.8.2.6 Eje Z

En la sección **Eje Z** del cuadro de diálogo "Cambiar aspecto" puede escribir la etiqueta del eje y definir el color de la línea del eje. También puede decidir si se deben mostrar o no las marcas de graduación en el eje. Esta sección es idéntica para todos los gráficos de barras 3D (gráficos de barras 3D y gráficos de barras agrupadas 3D).

© 2016-2022 Altova GmbH



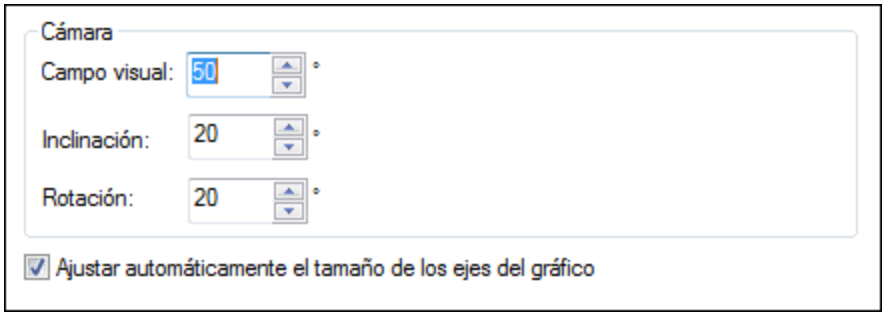
Etiqueta
El texto que escriba en el campo *Etiqueta* se imprimirá a la derecha del eje Z, a modo de descripción.

Línea
El eje se representa en el color elegido en la lista desplegable *Línea*. Puede usar uno de los colores predeterminados o hacer clic en el botón **Otro color...** para elegir un color estándar o definir un color personalizado. Haga clic en el botón **Seleccionar...** de la pestaña *Personalizado* y con la pipeta seleccione cualquier color de su pantalla.

Presentación de las marcas de graduación
En este cuadro de grupo puede activar o desactivar las marcas de graduación del eje.

5.12.8.2.7 Ángulos de visualización 3D

En los gráficos de barras 3D puede personalizar el aspecto 3D del gráfico en la sección **Ángulos de visualización 3D** del cuadro de diálogo "Cambiar aspecto".



La opción *Campo visual* determina si el diagrama se muestra como si se viera desde mayor o menor distancia. Los valores válidos oscilan entre el 1 y el 120. El diagrama se muestra a mayor distancia conforme mayores son los valores.

El valor *Inclinación* determina la rotación sobre el eje X, mientras que el valor *Rotación* define la rotación sobre el eje Y. Para adaptar automáticamente el tamaño de los ejes del gráfico al ancho de la ventana del gráfico, marque la casilla *Ajustar automáticamente el tamaño de los ejes del gráfico*.

Si marca la casilla *Ajustar automáticamente el tamaño de los ejes del gráfico*, XMLSpy calculará automáticamente el tamaño óptimo del eje X y del eje Y para la ventana de gráficos actual. El ancho y alto del gráfico se cambiará de forma dinámica cada vez que ajuste el tamaño de la ventana.

5.12.8.2.8 Tamaños

En la sección **Tamaños** del cuadro de diálogo "Cambiar aspecto" puede definir los distintos márgenes del gráfico así como el tamaño de los ejes y de las marcas de graduación. Tenga en cuenta que no todas las propiedades descritas a continuación están disponibles en todos los tipos de gráficos.

General

Margen exterior	El espacio entre el área de trazado y el borde de la ventana del gráfico.
Desde el título al área de trazado	El espacio entre el título del gráfico y el borde superior del área de trazado.
Desde la leyenda al área de trazado	El espacio entre el borde inferior del área de trazado y la leyenda.

Gráfico circular

Desde el área de trazado a las etiquetas	En gráficos circulares, el espacio entre el borde del círculo y las etiquetas.
Alto del gráfico circular	En gráficos circulares 3D, el alto del círculo.
Sombra paralela del gráfico circular	En gráficos circulares 3D, la longitud de la sombra (si se activa esta opción en la sección Gráfico circular).

Eje X

Desde el eje X a su etiqueta	En gráficos de barras y líneas, el espacio entre el eje X y su etiqueta.
Desde el eje X al área de trazado	En gráficos de barras y líneas 2D, el espacio entre el eje X y el área de trazado.
Tamaño de las marcas de graduación del eje X	En gráficos de barras y líneas, la longitud de las marcas de graduación del eje X.

Eje Y

Desde el eje Y a su etiqueta	En gráficos de barras y líneas, el espacio entre el eje Y y su etiqueta.
Desde el eje Y al área de trazado	En gráficos de barras y líneas 2D, el espacio entre el eje Y y el área de trazado.
Tamaño de las marcas de graduación del eje Y	En gráficos de barras y líneas, la longitud de las marcas de graduación del eje Y.

Eje Z

Desde el eje Z a su etiqueta	En gráficos de barras 3D, el espacio entre el eje Z y su etiqueta.
Tamaño de las marcas de graduación del eje Z	En gráficos de barras 3D, la longitud de las marcas de graduación del eje Z.

Presentación de las líneas

Tamaño de la forma de conexión	En gráficos de líneas, el tamaño de los recuadros que marcan los valores en el gráfico.

Tamaños de ejes 3D

© 2016-2022 Altova GmbH

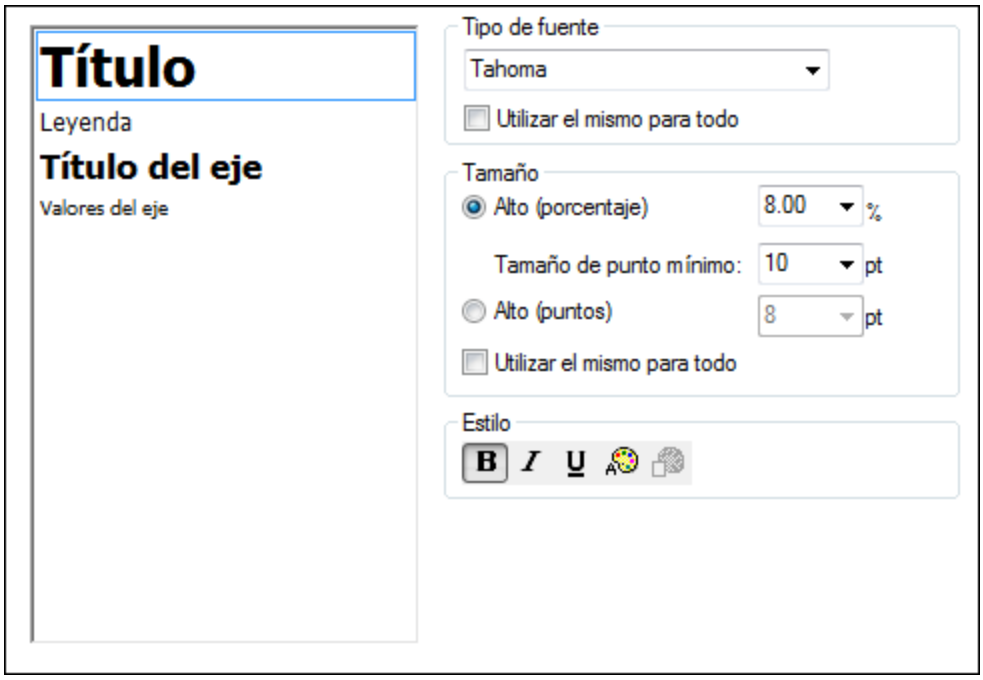
Ajuste manual del tamaño del eje X de la base	En gráficos de barras 3D, define la relación entre la longitud del eje X y el tamaño de la ventana del gráfico. Tenga en cuenta que debe desactivar la casilla *Ajustar automáticamente el tamaño de los ejes del gráfico* en la sección **Ángulos de visualización 3D**. De lo contrario, el tamaño se calcula automáticamente independientemente del tamaño indicado en esta sección.
Ajuste manual del tamaño del eje Y de la base	En gráficos de barras 3D, define la relación entre la longitud del eje Y y el tamaño de la ventana del gráfico. Tenga en cuenta que debe desactivar la casilla *Ajustar automáticamente el tamaño de los ejes del gráfico* en la sección **Ángulos de visualización 3D**. De lo contrario, el tamaño se calcula automáticamente independientemente del tamaño indicado en esta sección.
Margen entre series de eje Z	En gráficos de barras 3D, la distancia entre las distintas series del eje Z.
Medidor	
Ancho del marco	En medidores radiales, el ancho del marco que rodea el medidor.
Marcas de graduación del medidor	
Desde el marco a las marcas de graduación	En medidores radiales, el espacio entre el borde interno del marco y las marcas de graduación.
Longitud de las marcas de graduación principales	En medidores radiales, la longitud de las marcas de graduación principales (es decir, las marcas que llevan una etiqueta).
Ancho de las marcas de graduación principales	En medidores radiales, el ancho de las marcas de graduación principales (es decir, las marcas que llevan una etiqueta).
Longitud de las marcas de graduación secundarias	En medidores radiales, la longitud de las marcas de graduación que no tienen un valor.
Ancho de las marcas de graduación secundarias	En medidores radiales, el ancho de las marcas de graduación que no tienen un valor.
Puntero del medidor	
Longitud del puntero	En medidores radiales, la longitud del puntero. (Tenga en cuenta que el porcentaje se calcula con el diámetro del medidor. Si elige un valor superior a 50%, el puntero apuntará fuera del medidor.)
Ancho de la cabeza del puntero	En medidores radiales, el ancho del puntero en el centro del medidor.
Radio de la cabeza del puntero	En medidores radiales, el radio de la base que cubre el centro del puntero.
Intervalo de colores del medidor	
Distancia desde el borde al intervalo de colores	En medidores radiales, el espacio entre el borde interior del marco del medidor y el borde exterior del intervalo de colores 390.
Ancho del intervalo de colores	En medidores radiales, el ancho del intervalo de colores personalizable. (Tenga en cuenta que el porcentaje se calcula con el diámetro del medidor.)
Valor del medidor	
Desplazamiento con respecto al centro	Distancia desde el centro a la que se muestra el valor del medidor.

Valor suplementario

Desplazamiento con respecto al centro	Distancia desde el centro a la que se muestra la etiqueta suplementaria (*definida en el apartado Opciones de gráficos* 384).

5.12.8.2.9 Fuentes

La sección **Fuentes** del cuadro de diálogo "Cambiar aspecto" permite configurar las fuentes de los objetos de la ventana del gráfico.



Opciones de configuración de las fuentes

Puede elegir el tipo, el tamaño y el estilo de la fuente de cada elemento que aparece en la ventana del gráfico. Puede definir el tamaño de la fuente como porcentaje del tamaño del gráfico y definir un tamaño mínimo en puntos o especificar un valor absoluto (en puntos). Para aplicar el mismo tipo de fuente y el mismo tamaño de fuente a todos los elementos de texto, active donde corresponda la casilla *Utilizar el mismo para todo.*

Los nombres de los elementos del cuadro de lista son:

- **Título**: el nombre del gráfico
- **Leyenda**: texto que explica los colores usados en el gráfico
- **Etiquetas**: la designación de las secciones del gráfico circular
- **Título del eje**: el nombre de los ejes X, Y y Z en un gráfico de barras o líneas
- **Valores del eje**: las unidades que aparecen en un eje de un gráfico de barras o líneas
- **Valores de las marcas de graduación:** las unidades que aparecen en un medidor
- **Valores:** los valores que aparecen en las barras de un gráfico de barras

© 2016-2022 Altova GmbH

5.12.9 Exportación

Al hacer clic en el botón **Exportar** aparece un menú con estas opciones:

- *Guardar gráfico como archivo:* los formatos de archivo de imagen disponibles son PNG, GIF, BPM y JPG.
- *Copiar imagen actual al portapapeles*: permite pegar el gráfico desde el portapapeles en otra aplicación.
- *Copiar imagen con el tamaño ajustado al portapapeles*: permite ajustar el tamaño de la imagen antes de copiarla en el portapapeles y después pegarla en otra aplicación.
- *Imprimir este gráfico:* envía la imagen a la impresora. Puede definir el ancho y alto de la imagen como porcentaje del tamaño de la página.
- *Copiar código XSLT o XQuery al portapapeles:* crea un fragmento de código XSLT o XQuery. Básicamente se trata de la función de extensión `CreateChart`. Esta función se puede utilizar con otras funciones de extensión de Altova y se pueden procesar con XMLSpy para generar gráficos. A fin de ayudarle a usar la función de extensión `CreateChart`, con cada fragmento se crea también un ejemplo de uso con comentarios.

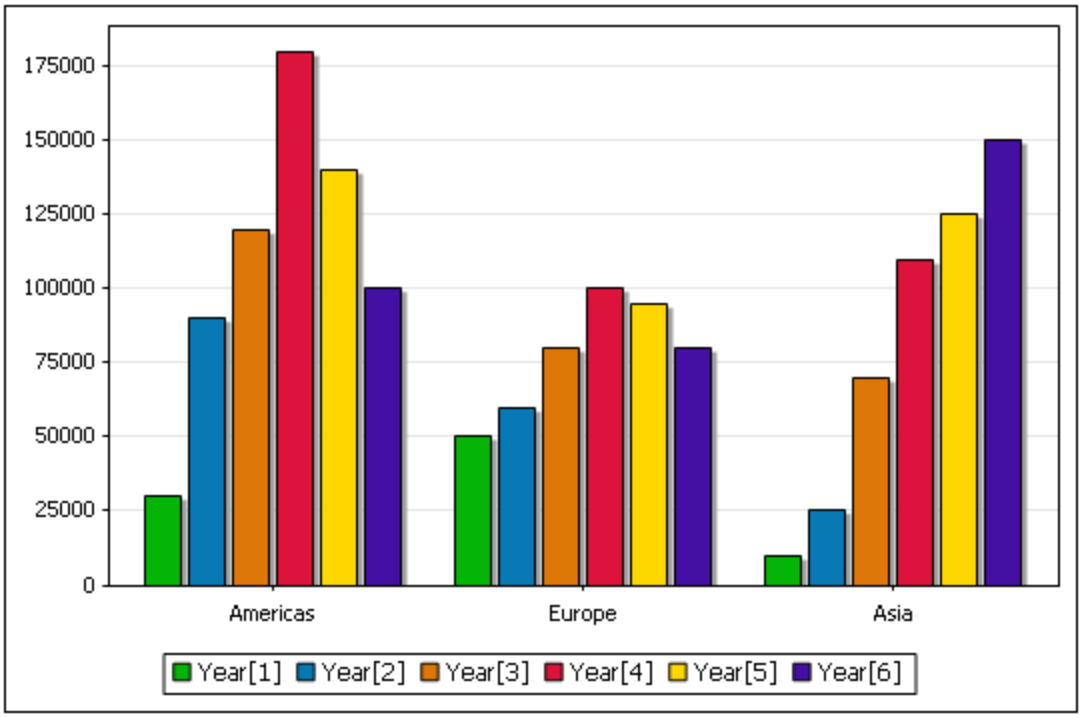
5.12.10 Ejemplo de gráfico básico

Tomemos este documento XML como ejemplo. (Es el archivo `YearlySales.xml` de la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Tutorial`.)

```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <Region id="Americas">
              <Year id="2005">30000</Year>
              <Year id="2006">90000</Year>
              <Year id="2007">120000</Year>
              <Year id="2008">180000</Year>
              <Year id="2009">140000</Year>
              <Year id="2010">100000</Year>
       </Region>
       <Region id="Europe">
              <Year id="2005">50000</Year>
              <Year id="2006">60000</Year>
              <Year id="2007">80000</Year>
              <Year id="2008">100000</Year>
              <Year id="2009">95000</Year>
              <Year id="2010">80000</Year>
       </Region>
       <Region id="Asia">
              <Year id="2005">10000</Year>
              <Year id="2006">25000</Year>
              <Year id="2007">70000</Year>
              <Year id="2008">110000</Year>
              <Year id="2009">125000</Year>
              <Year id="2010">150000</Year>
       </Region>
```

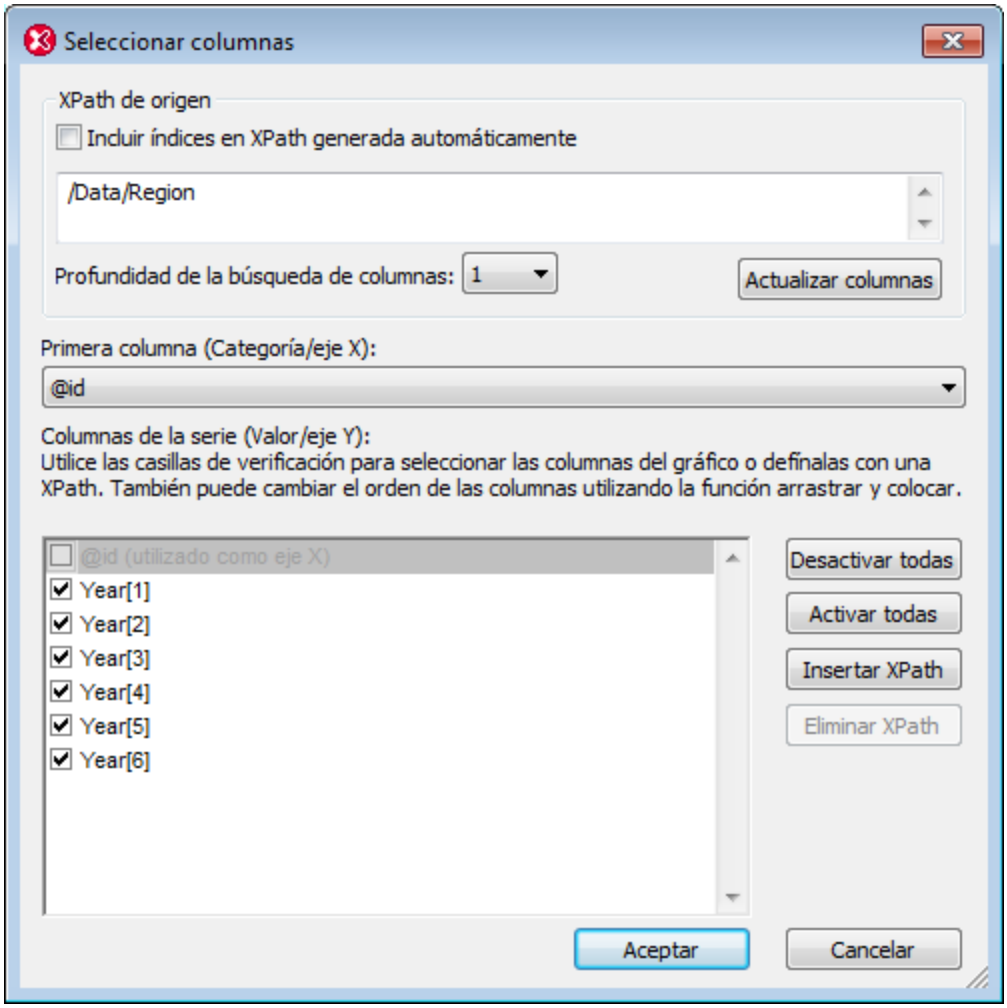
```
</Data>
```

Queremos generar un gráfico en el que las tres regiones (América, Europa y Asia) estén representadas en el eje X y que muestre las ventas anuales en cada región. Nuestro gráfico debería ser como este:



Crear este gráfico es bastante sencillo porque podemos seleccionar el elemento `Region` como XPath de origen. La expresión XPath de origen devuelve una secuencia de tres componentes: los tres elementos `Region`. Cada elemento `Region` será a su vez el nodo de contexto a la hora de seleccionar datos para los ejes X e Y.

© 2016-2022 Altova GmbH



Para la serie queremos los elementos `Year` de cada región, así que es suficiente fijar el nivel de profundidad de la búsqueda en 1. Para el eje X seleccionamos el atributo `id` del elemento `Region`. Los valores del atributo `id` se usarán, por tanto, como etiquetas de las tres marcas de graduación del eje X. A continuación marcamos todas las series `Year` porque vamos a incluir todos los elementos `Year` en la tabla de datos del gráfico.

Ahora hacemos clic en **Aceptar** y se genera el gráfico que queríamos crear. Para ver un ejemplo más avanzado, consulte el siguiente apartado: Ejemplo de gráfico avanzado 401.

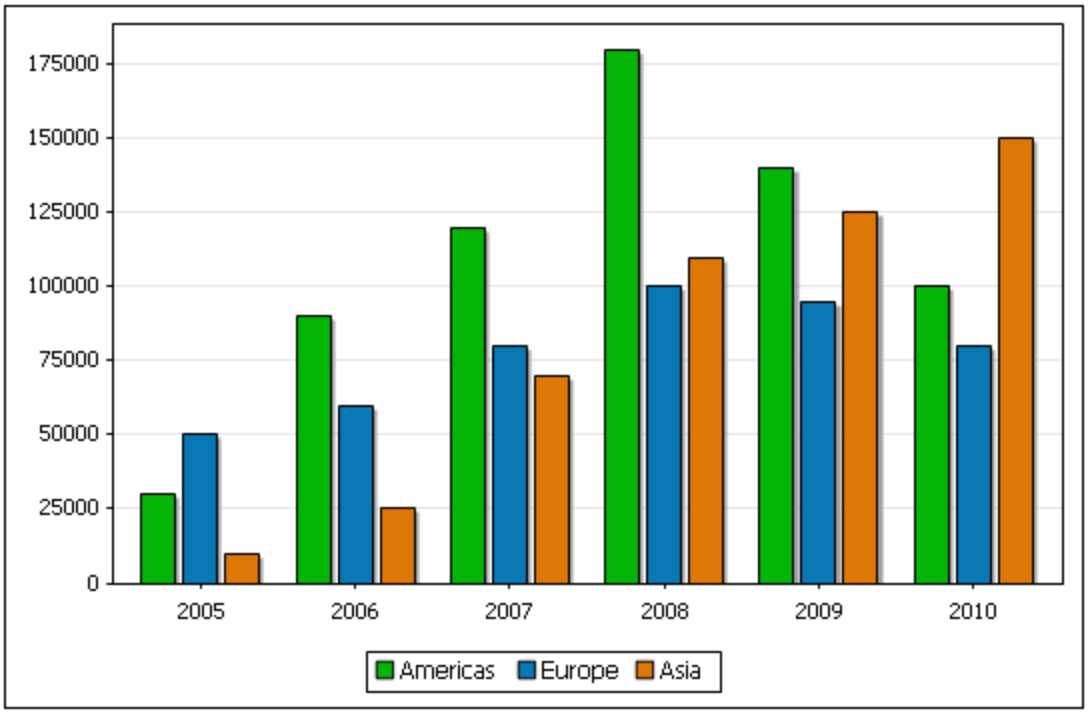
5.12.11 Ejemplo de gráfico avanzado

Tomemos este documento XML como ejemplo. (Es el archivo `YearlySales.xml` de la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Tutorial`.)

```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <Region id="Americas">
              <Year id="2005">30000</Year>
```

```
            <Year id="2006">90000</Year>
            <Year id="2007">120000</Year>
            <Year id="2008">180000</Year>
            <Year id="2009">140000</Year>
            <Year id="2010">100000</Year>
        </Region>
        <Region id="Europe">
            <Year id="2005">50000</Year>
            <Year id="2006">60000</Year>
            <Year id="2007">80000</Year>
            <Year id="2008">100000</Year>
            <Year id="2009">95000</Year>
            <Year id="2010">80000</Year>
        </Region>
        <Region id="Asia">
            <Year id="2005">10000</Year>
            <Year id="2006">25000</Year>
            <Year id="2007">70000</Year>
            <Year id="2008">110000</Year>
            <Year id="2009">125000</Year>
            <Year id="2010">150000</Year>
        </Region>
</Data>
```

Queremos generar un gráfico que muestre los años en el eje X y compare las ventas de cada región en cada año. Nuestro gráfico debería ser como este:



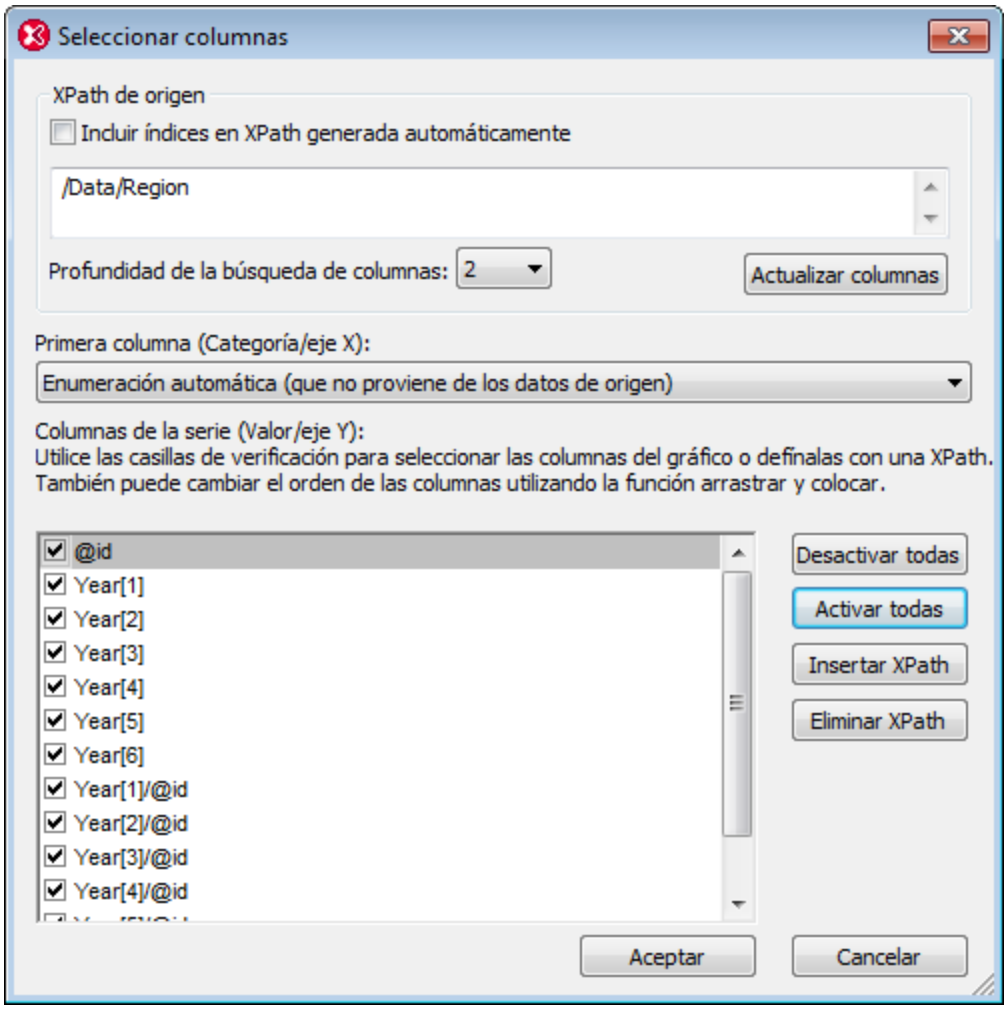
© 2016-2022 Altova GmbH

Hay dos maneras de crear este gráfico y ambas se describen a continuación. En este apartado aprenderá cómo se pueden combinar los parámetros de selección de datos para producir los resultados deseados.

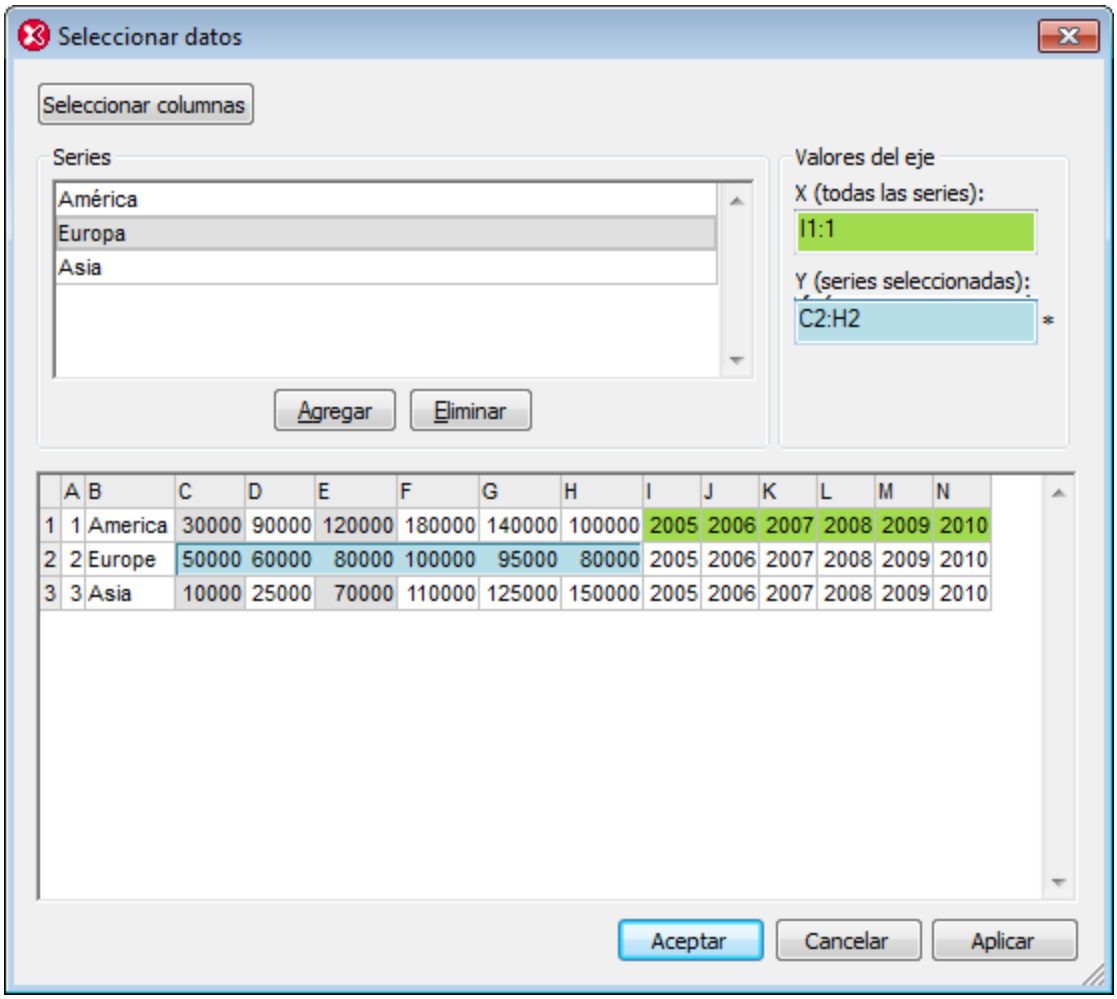
Método 1: modificar los valores de los ejes

Este primer método consiste en modificar en el cuadro de diálogo **Seleccionar datos** los ejes que se seleccionaron en el cuadro de diálogo **Seleccionar columnas**. Para ello es imprescindible que los datos necesarios estén disponibles en la tabla de datos del gráfico en el cuadro de diálogo **Seleccionar datos**.

1. En el cuadro de diálogo **Seleccionar columnas**, compruebe que los nodos necesarios estarán disponibles para los ejes X e Y. En la imagen siguiente puede observar que la profundidad de la búsqueda de columnas se fijó en `2` para que se puedan seleccionar también los atributos `Year/@id`.



2. En el cuadro de diálogo **Seleccionar datos** (*imagen siguiente*) la tabla de datos del gráfico tiene estas columnas: la columna A es la selección del eje X (con numeración automática), el resto de columnas son las columnas de la serie (eje Y), es decir, los atributos `Region/@id`, el contenido de los elementos `Year` y los atributos `Year/@id`. Además, como puede ver, hay sólo tres filas y por tanto el gráfico tendrá sólo tres marcas de graduación. Por último, observe que hay 13 columnas de series.



3. En el panel "Series" elimine 10 de las 13 series (cualesquiera) y cambie el nombre de las tres series restantes a `América`, `Europa` y `Asia` (*ver imagen anterior*). El orden seleccionado en este panel será el orden en el que se etiqueten las marcas de graduación del eje X.
4. En el panel "Series" seleccione la serie `América`. En el panel "Valores del eje" haga clic en el cuadro del eje X. Después haga clic en la celda `I1` de la tabla de datos del gráfico y arrastre el ratón hasta la celda `N1`. Después haga clic en el cuadro del eje Y y escriba `C1:H1` o seleccione este rango de celdas en la tabla de datos del gráfico.
5. Para las series `Europa` y `Asia`, seleccione el rango de celdas `C2:H2` y `C3:H3` respectivamente para el eje X. Para el eje Y la selección puede ser la misma que la utilizada en la serie `América`.
6. Haga clic en **Aceptar** y se genera el gráfico.

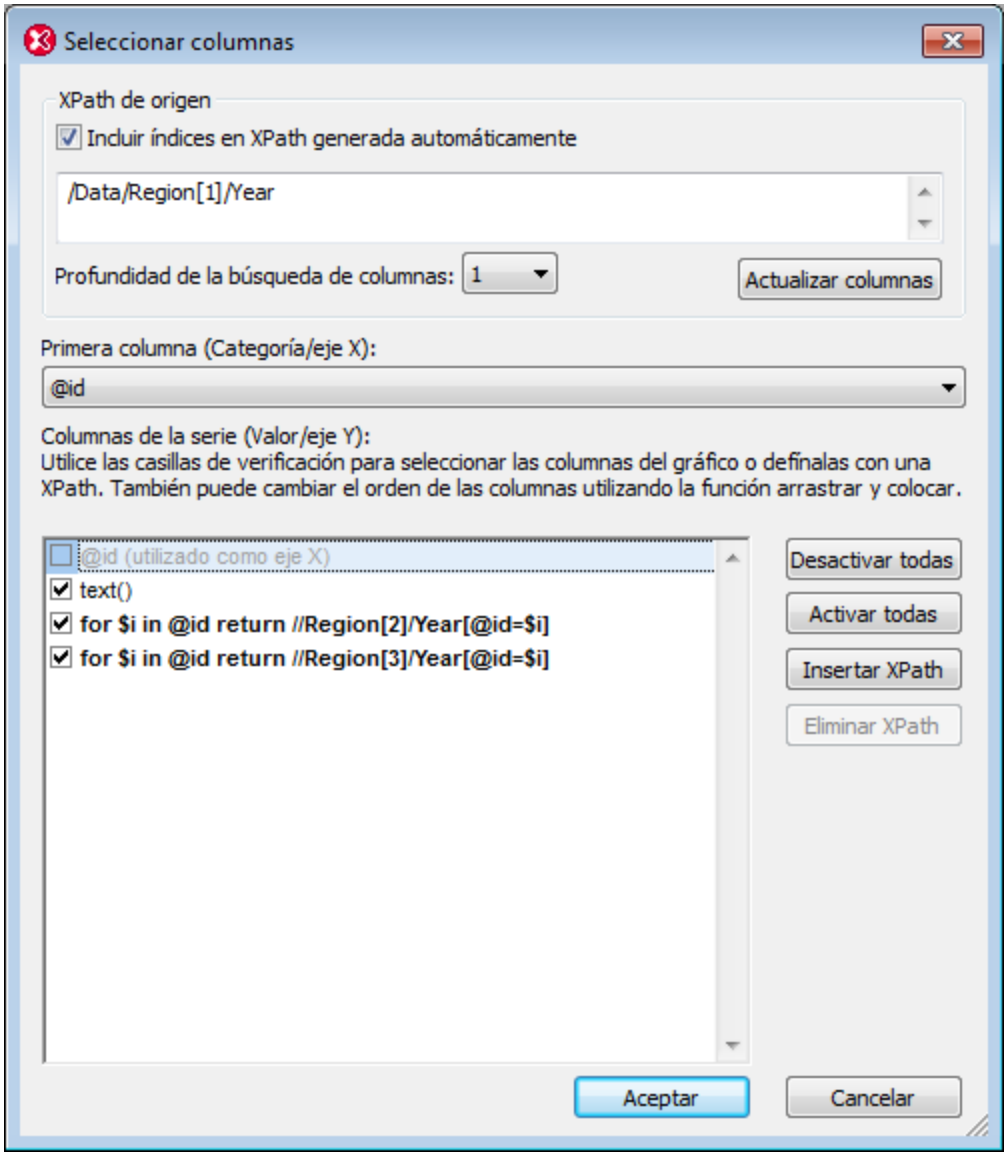
Nota: El número de marcas de graduación del eje X (cuyo número predeterminado equivale al número de filas de la tabla de datos del gráfico) sube de tres a seis porque el número de etiquetas del eje X es seis.

Método 2: generar series mediante expresiones XPath

El segundo método consiste en insertar expresiones XPath para generar series de datos. Esto es necesario porque la XPath de origen (*ver imagen siguiente*) no tiene como descendientes los nodos que deseamos utilizar en las series. Sin embargo, la XPath de origen genera seis marcas de graduación para el eje X (porque selecciona los seis elementos `Year` del primer elemento `Region`). Para poder seleccionar el primer elemento

 © 2016-2022 Altova GmbH

`Region` usando el predicado `[1]` es necesario marcar la casilla *Incluir índices* y hacer clic en el botón **Actualizar columnas**.

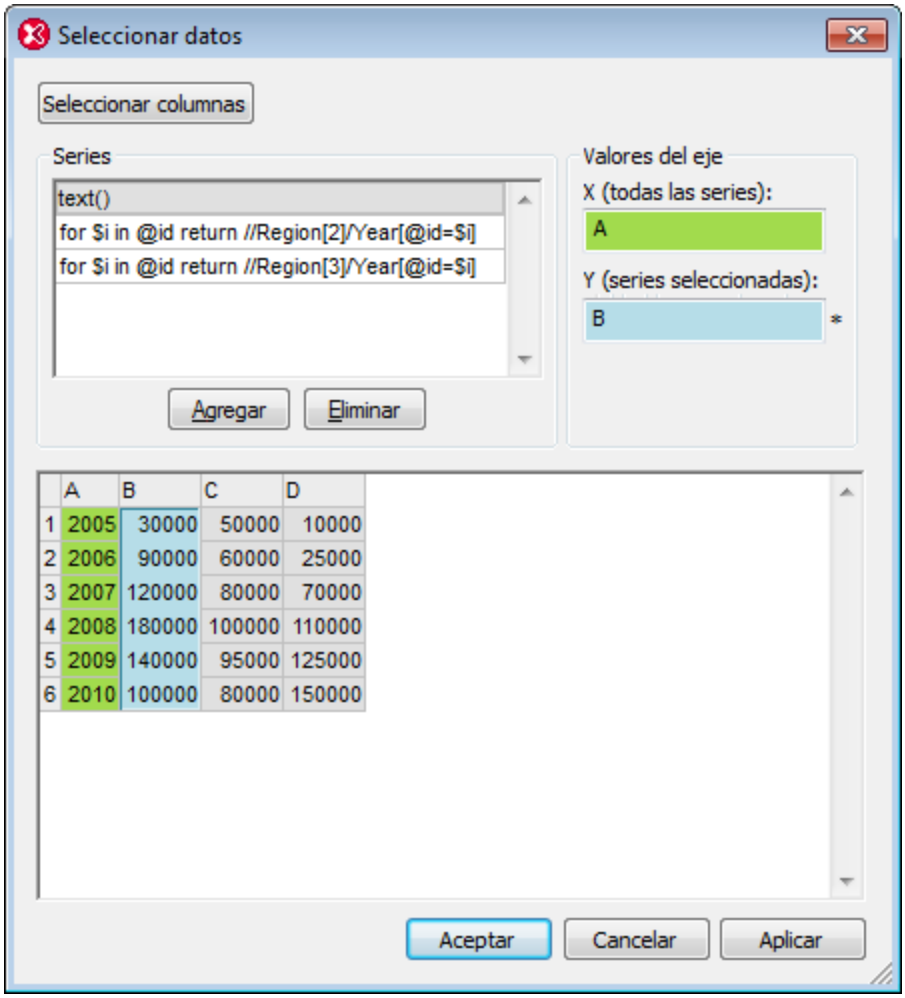


En el cuadro de diálogo **Seleccionar columnas** el elemento `Region[1]/Year` tiene sólo dos descendientes: `@id` y `text()`. Seleccione el atributo `@id` para el eje X. Así se genera la etiqueta correcta para cada una de las marcas de graduación del eje X. Esta es la tabla de datos del gráfico que se recupera:

XPath de origen	Eje X	Eje Y (columnas de la serie)		
`Region[1]/Year[1]`	`@id`	`text()`	XPath-1	XPath-2
`Region[1]/Year[2]`	`@id`	`text()`	XPath-1	XPath-2
`Region[1]/Year[3]`	`@id`	`text()`	XPath-1	XPath-2
`Region[1]/Year[4]`	`@id`	`text()`	XPath-1	XPath-2
`Region[1]/Year[5]`	`@id`	`text()`	XPath-1	XPath-2
`Region[1]/Year[6]`	`@id`	`text()`	XPath-1	XPath-2

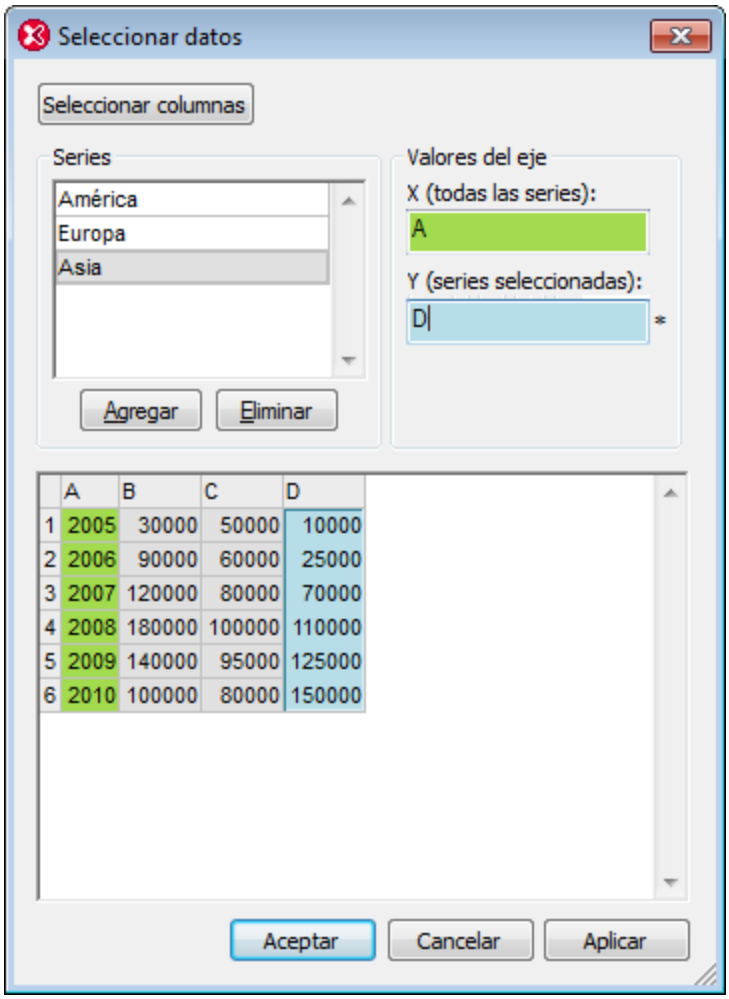
Recuerde que el nodo de contexto es cada uno de los seis elementos `Region[1]/Year`. La primera expresión XPath busca el valor del atributo `Year/@id` actual y devuelve el elemento `Region[2]/Year` que tenga el mismo valor `Year/@id` que el valor `@id` del elemento `Region[1]/Year` actual. La segunda expresión XPath hace lo mismo con los elementos `Region[3]/Year`. De este modo, por cada uno de los seis años, las tres series del eje Y son los descendientes del elemento `Year` de cada uno de los tres elementos `Region`. (El nodo `text()` devuelve el contenido de los elementos `Region[1]/Year`.)

La tabla de datos del gráfico tendría este aspecto en el cuadro de diálogo **Seleccionar datos**.



El nombre de las series del cuadro de diálogo **Seleccionar datos** se puede cambiar. En lugar de expresiones XPath (*como en la imagen anterior*) puede usar nombres más significativos, en este caso, el nombre de las regiones (*ver imagen siguiente*). En el panel "Valores del eje" puede asignar la columna de datos pertinente a cada serie (para ello haga clic en la serie, después en el cuadro del eje Y y después seleccione la columna de la tabla de datos).

© 2016-2022 Altova GmbH



Los dos métodos descritos en este apartado generan el mismo gráfico. El objetivo de describir ambos métodos es explicar cómo se pueden usar los diferentes parámetros de selección de datos.

5.12.12 Ejemplo de gráfico de velas

Los gráficos de velas se suelen usar para representar la oscilación de las cotizaciones bursátiles. Hay dos tipos de gráficos de velas:

- Gráficos de velas de cuatro series: representan el valor de apertura, el valor máximo, el valor mínimo y el valor de cierre (en un día).
- Gráficos de velas de tres series: representan el valor máximo, el valor mínimo y el valor de cierre (en un día).

El archivo `Candlestick.xml`, guardado en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Tutorial`, es un ejemplo de documento XML cuya estructura se puede usar para generar gráficos de velas. A continuación mostramos un fragmento del archivo para que pueda observar su estructura básica.

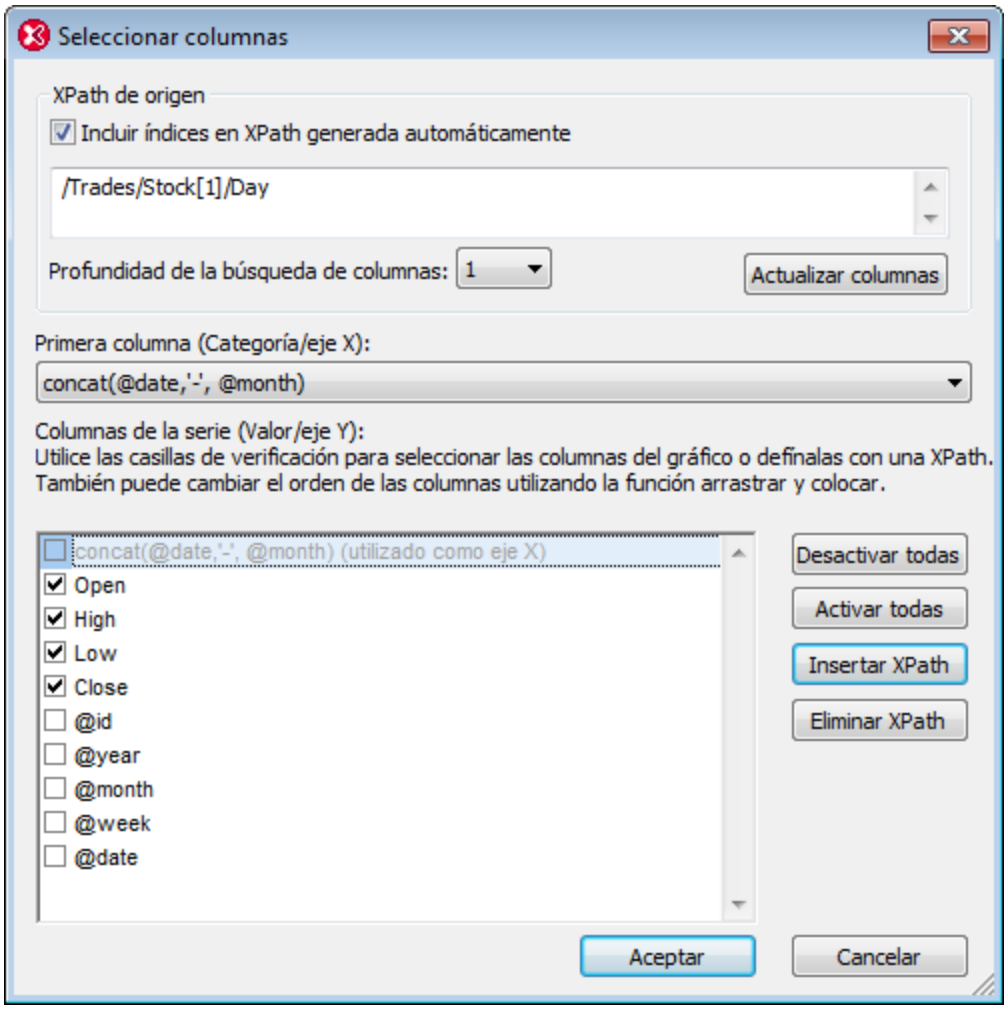
```
<Trades>
  <Stock name="MyStock">
    <Day id="20110103" year="2011" month="Jan" week="01" date="03">
      <Open>90</Open>
      <High>110</High>
      <Low>88</Low>
      <Close>105</Close>
    </Day>

    .....

  </Stock>
</Trades>
```

A continuación describimos cómo crear un gráfico de velas para este archivo:

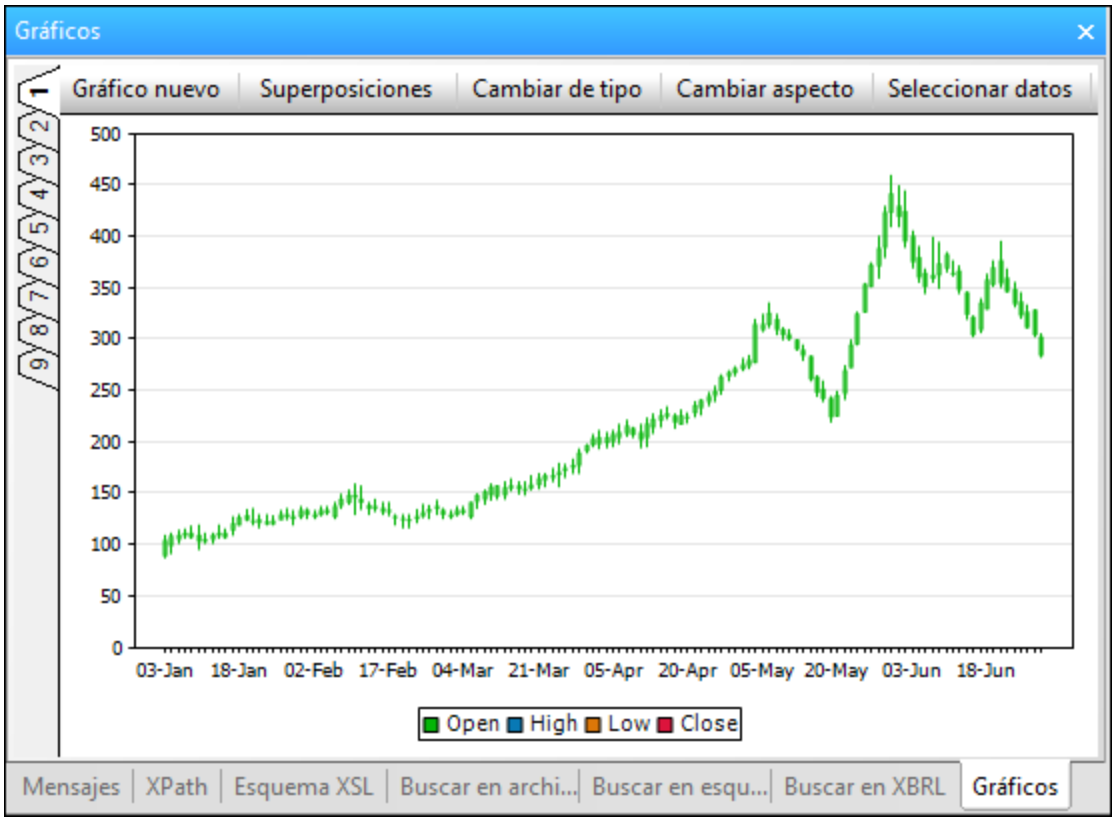
1. Sitúe el cursor dentro de la etiqueta del elemento `Day` y haga clic en el botón **Gráfico nuevo** de la ventana de gráficos 346. Aparece el cuadro de diálogo "Seleccionar columnas" (*ver imagen siguiente*). Marque la casilla *Incluir índices* (si no está activada ya) y haga clic en **Actualizar columnas**.



2. Haga clic en el botón **Insertar XPath** e inserte la expresión XPath `concat(@date, '-', @month)`.
3. En la lista desplegable del cuadro combinado "Primera columna" seleccione la expresión XPath recién creada. Esta expresión utilizará la fecha y el mes de cada elemento `Day` como etiqueta de las marcas de graduación del eje X.

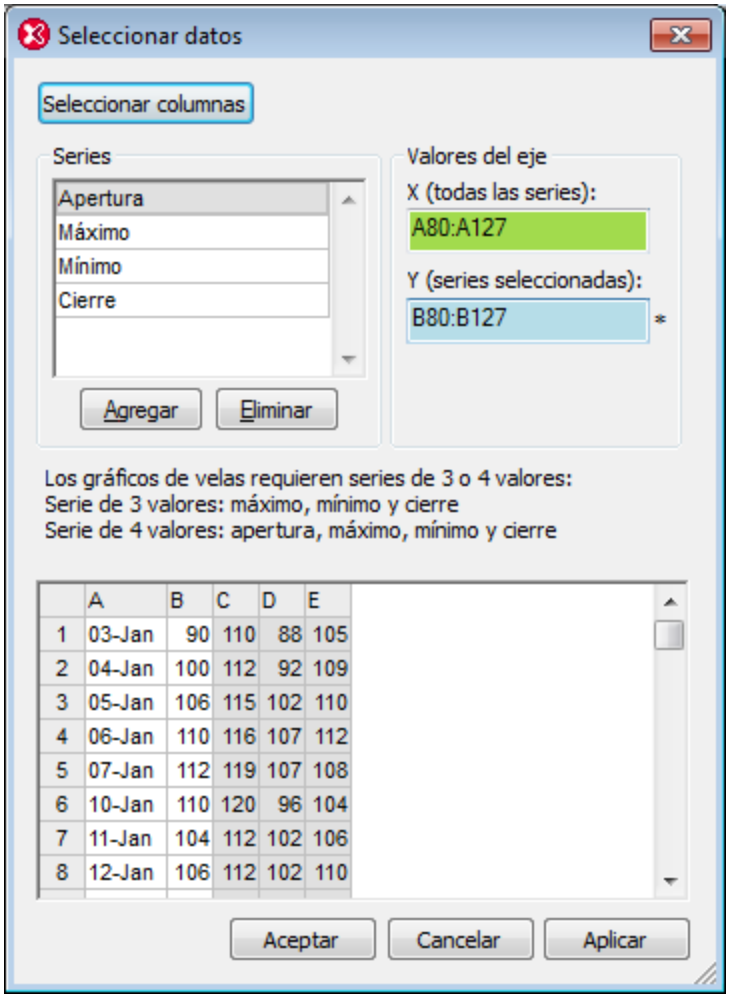
© 2016-2022 Altova GmbH

4. Para las series del eje Y, marque las casillas `Open`, `High`, `Low` y `Close` del panel "Columnas de la serie" (es decir, los valores de apertura, máximo, mínimo y cierre).
5. Haga clic en **Aceptar**.
6. Pulse el botón **Cambiar de tipo** de la ventana de gráficos y cambie el tipo de gráfico a gráfico de velas.
7. Haga clic en **Aceptar**. Instantáneamente se crea un gráfico de velas como el que aparece en esta imagen:



Seleccionar un subconjunto de datos

Si desea visualizar un subconjunto de los datos del gráfico seleccionados en el cuadro de diálogo "Seleccionar columnas" (si, por ejemplo, desea visualizar un rango de fechas determinado), haga clic en el botón Seleccionar datos de la ventana de gráficos. A continuación aparece el cuadro de diálogo "Seleccionar datos" (imagen siguiente).



Si lo desea, puede cambiar el nombre de las series. En el panel "Valores del eje" introduzca el rango de celdas `A80:A127` en el cuadro del eje X (es decir, el rango de fechas) tal y como puede ver en la imagen anterior. A continuación haga clic en el cuadro del eje Y, seleccione la serie del panel "Series" e introduzca el rango de celdas para la serie. Repita este paso para cada una de las cuatro celdas. Por ejemplo, en la imagen anterior puede ver el rango de celdas del eje Y seleccionado para la serie Apertura (`Open`). Para más información sobre el cuadro de diálogo "Seleccionar datos", consulte el apartado Datos del gráfico 366.

 © 2016-2022 Altova GmbH

5.13 Firmas XML

Los archivos XML se pueden firmar de forma digital y estas firmas se pueden verificar. Si el archivo XML se cambia después de firmarse, la verificación fallará. XMLSpy ofrece funciones para crear y verificar firmas XML.

Las firmas XML en las vistas de XMLSpy

Las firmas XML se pueden crear en todos los tipos de archivos XML, incluidos los archivos de esquema XML, WSDL y XBRL. Por tanto, los comandos **XML | Crear firma XML** 413 y **XML** 416 **|** 413 **Verificar la firma XML** 416 están disponibles en todas las vistas de XMLSpy: vista Texto 145, vista Cuadrícula 169, vista Esquema 171, vista WSDL 251 y vista XBRL 264.

Funcionamiento

El proceso de creación y verificación de firmas XML funciona de la siguiente manera:

1. El archivo XML se firma o bien con una clave privada de un certificado, o bien con una contraseña. En XMLSpy puede crear una firma con el comando **XML | Crear firma XML** 413. La firma se obtiene al procesar: (i) el documento XML y (ii) la clave privada de un certificado o una contraseña.
2. La firma se puede incluir en el archivo XML o se puede guardar en un archivo distinto.
3. Para verificar la firma del archivo XML se usa la clave pública del certificado o la contraseña. El proceso de verificación funciona así: primero, se procesa (i) el documento XML y (ii) la clave pública del certificado o la contraseña que se usó para crear la firma y, en segundo lugar, se compara este resultado con la firma. Si el archivo XML se cambió después de firmarse, la verificación fallará. En XMLSpy puede verificar una firma con el comando **XML | Verificar la firma XML** 416.

Para más información sobre cómo crear y verificar firmas XML en XMLSpy consulte los apartados de esta sección:

- Crear firmas XML 413
- Verificar firmas XML 416

Uso de certificados en firmas XML

Para poder usarlos en firmas XML, los certificados deben tener una clave privada y una clave pública. La clave privada se usa para crear la firma XML, mientras que la clave pública se usa para verificar la firma XML.

En un caso de uso típico, la persona que envía el documento XML tiene acceso a la clave privada de un certificado y crea con ella la firma XML. El destinatario del documento tendrá acceso a la clave pública del certificado. Este acceso a la clave pública puede ser de dos tipos: (i) la persona que envía el documento envía la información sobre la clave pública con la firma o (ii) el destinatario del documento tiene acceso a una versión de clave pública del certificado utilizado por quien envió el documento.

Para más información sobre los certificados, consulte el apartado Trabajar con certificados 419.

Nota: la función para firmas XML de XMLSpy es compatible con certificados de tipo RSA-SHA1, DSA-SHA1 y SHA-256. Las firmas basadas en contraseñas usan el algoritmo HMAC-SHA256 (a partir de la versión 2018 de XMLSpy, *véase* *Crear firmas XML* 413).

Validez de los documentos XML

Si se incrusta una firma XML en el documento XML, se añade al documento un elemento `Signature` en el espacio de nombres `http://www.w3.org/2000/09/xmldsig#`. Para que el documento siga siendo válido con respecto al esquema, este debe contener las declaraciones de elementos correspondientes. XMLSpy incrusta las firmas en el documento de dos maneras:

- *Firmas envueltas:* el elemento `Signature` se crea como último elemento secundario del elemento raíz (o del elemento de documento).
- *Firmas envolventes:* el elemento `Signature` se crea como elemento raíz (o como elemento de documento) y el elemento de documento XML original se coloca dentro de un elemento secundario del elemento `Signature` llamado `Object`.

Si no quiere modificar el esquema del documento XML, la firma XML se puede crear en un archivo externo. Para más información, consulte la descripción de este proceso en el apartado Crear firmas XML 413.

A continuación ofrecemos dos extractos de esquemas XML que muestran cómo se puede permitir el uso del elemento `Signature` de una firma envuelta. Puede usar estos ejemplos como referencia cuando necesite modificar sus esquemas.

En el primer ejemplo, el esquema de la firma XML se importa al esquema del usuario. El esquema de la firma XML está ubicado en la dirección web: http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:xsig="http://www.w3.org/2000/09/xmldsig#"
           elementFormDefault="qualified"
           attributeFormDefault="unqualified">
   <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
              schemaLocation="http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd"/>
   <xs:element name="Root">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="FirstChildOfRoot"/>
            <xs:element ref="SecondChildOfRoot" minOccurs="0"/>
            <xs:element ref="ThirdChildOfRoot" minOccurs="0"/>
            <xs:element ref="xsig:Signature" minOccurs="0"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   ...
</xs:schema>
```

Otra opción (*ver el ejemplo que sigue*) es añadir un elemento comodín genérico que coincida con cualquier elemento de otros espacios de nombres. Si se fija el atributo `processContents` en `lax`, el validador pasará por alto este elemento (porque no se encuentra una declaración de elemento equivalente). Consecuentemente, el usuario no necesita hacer referencia al esquema de las firmas XML. El inconveniente de esta opción, sin embargo, es que es posible añadir cualquier elemento (no sólo el elemento `Signature`) en la posición indicada del documento XML sin que el documento deje de ser válido.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           elementFormDefault="qualified"
```


© 2016-2022 Altova GmbH

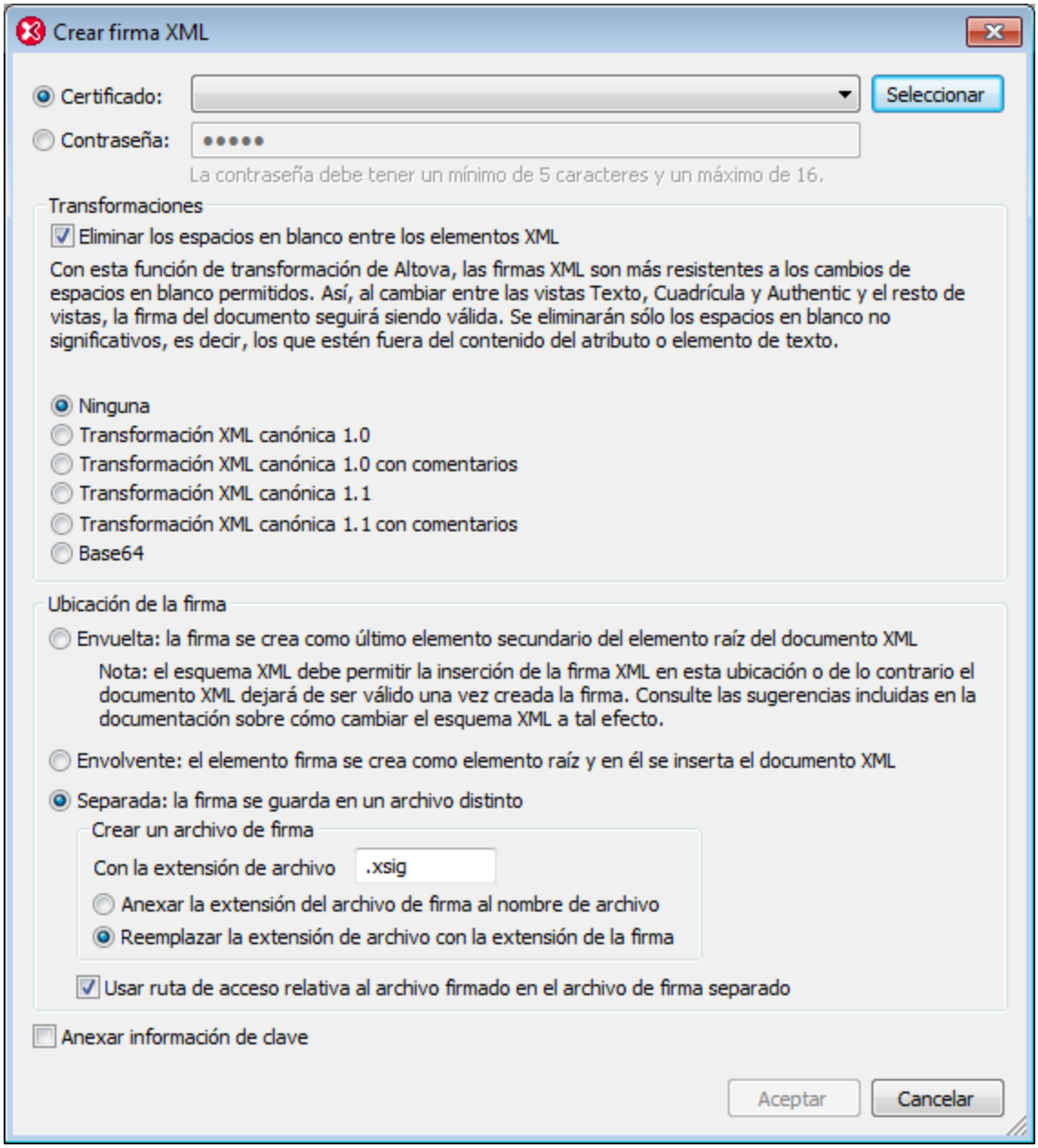
```
          attributeFormDefault="unqualified">
   <xs:element name="Root">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="selection"/>
            <xs:element ref="newsitems" minOccurs="0"/>
            <xs:element ref="team" minOccurs="0"/>
            <xs:any namespace="##other" minOccurs="0" processContents="lax"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   ...
</xs:schema>
```

Especificación del W3C

Para más información sobre las firmas XML, consulte la especificación del W3C para firmas digitales, disponible en https://www.w3.org/TR/xmldsig-core1/.

5.13.1 Crear firmas XML

Para crear una firma XML para un documento XML, abra el documento para el que desea crear la firma. Después haga clic en el comando de menú **XML | Crear firma XML**. A continuación se abre el cuadro de diálogo "Crear firma XML" (*ver imagen siguiente*), cuyas opciones se explican a continuación.



Método de autenticación: certificado o contraseña

La firma puede estar basada en un certificado o en una contraseña. Seleccione el botón de opción del método que desea usar.

- *Certificado:* haga clic en el botón **Seleccionar** y busque el certificado que desea usar. El certificado debe tener una clave privada. La firma se genera usando la clave privada del certificado. Para verificar la firma, es necesario tener acceso al certificado (o a una versión del certificado de clave pública). La clave pública del certificado se usa para verificar la firma. Para más información sobre los certificados, consulte el apartado Trabajar con certificados 419.

 © 2016-2022 Altova GmbH

- *Contraseña:* introduzca una contraseña de como mínimo cinco caracteres (y 16 caracteres como máximo). Esta contraseña será necesaria para verificar la firma.

Nota: a partir de la versión 2018 de XMLSpy se pasó de firmar documentos basándose en contraseñas con el algoritmo HMAC-SHA1 a firmarlos usando el algoritmo HMAC-SHA256. Las versiones de XMLSpy anteriores a la versión 2018 no podrán verificar firmas que usen contraseñas generadas con XMLSpy 2018.

Transformaciones

Los datos XML se transforman y el resultado de la transformación se utiliza para crear la firma. Puede especificar el algoritmo de canonización que se debe aplicar a los datos XML del archivo (el contenido de `SignedInfo`) antes de realizar los cálculos de la firma. Tenga en cuenta que hay diferencias entre usar un tipo de transformación u otro:

- *Transformación canónica XML con o sin comentarios:* si se incluyen comentarios para el cálculo de la firma, cualquier cambio en los comentarios de los datos XML darán lugar a un error de verificación. Aparte de esto, puede cambiar los comentarios o añadir comentarios nuevos en el documento XML después de firmarlo y la firma seguirá siendo auténtica.
- *Base64:* se considera que el elemento raíz (o de documento) del documento XML tiene codificación Base64 y se lee en formato binario. Si el elemento raíz no tiene codificación Base64, se genera un error o se interpreta que el elemento está vacío.
- *Ninguna*: no se realiza ninguna transformación y los datos XML del archivo binario guardado en el disco se pasan directamente para crear la firma. Si después se realizan cambios en los datos, se produce un error de verificación de la firma. No obstante, si marca la casilla *Eliminar los espacios en blanco,* se eliminan todos los espacios en blanco y los cambios en los espacios en blanco se omiten. La principal diferencia entre no usar ninguna transformación y usar una transformación canónica es que la canonización da lugar a una secuencia de datos XML, en la que algunas diferencias (como diferencias en el orden de los atributos) están normalizadas. Como consecuencia, una transformación canónica normalizará determinados cambios, como por ejemplo los cambios en el orden de los atributos (y la verificación se realizará correctamente). Por el contrario, si no se realiza ninguna transformación, ese tipo de cambios se tendrá en cuenta y se producirá un error de verificación. Recuerde que si la firma se incrusta en el documento (ya sea como firma envuelta o envolvente), entonces se realiza una canonización por defecto. De modo que los datos XML se usarán tal y como están (es decir, sin transformación) cuando la firma esté en un archivo separado, cuando se seleccione la opción *Ninguna* y cuando esté desactivada la casilla *Eliminar los espacios en blanco.*

Posición de la firma

Puede colocar la firma dentro del archivo XML o guardarla en un archivo separado. Hay varias opciones:

- *Firma envuelta:* se crea el elemento `Signature` como último elemento secundario del elemento (de documento) raíz.
- *Firma envolvente:* se crea el elemento `Signature` como elemento (de documento) raíz y el documento XML se inserta como elemento secundario de este.
- *Firma separada:* la firma XML se crea en un archivo separado. En este caso puede indicar la extensión de archivo del archivo de firma y si el nombre del archivo se crea (i) uniendo la extensión de archivo al nombre del archivo XML (p. ej. `test.xml.xsig`) o (ii) usando el nombre del archivo XML pero con la extensión del archivo de firma en lugar de la extensión XML (p. ej. `test.xsig`). También puede indicar si en el archivo de firma la referencia al archivo XML es una ruta de acceso relativa o absoluta.

Nota: las firmas XML para archivos de esquema XML (`.xsd`) se pueden crear en la vista Esquema como archivos de firma separados (no como firma incrustada). Las firmas XML para archivos XBRL se

pueden crear en la vista XBRL como archivos de firma separados (no como firma incrustada). Las firmas XML para archivos WSDL se pueden crear en la vista WSDL como archivos de firma separados o como firmas envueltas en el archivo WSDL.

Nota: si la firma XML se crea como archivo separado, entonces se crea una referencia en el archivo de firma que asocia el archivo XML con el archivo de firma. Como consecuencia, cuando la firma se guarde en un archivo separado, debe asegurarse de que el archivo de firma sea el documento activo antes de iniciar la verificación de la firma.

Anexar información de clave

La opción *Anexar información de clave* está disponible si la firma está basada en un certificado. Si la firma está basada en una contraseña, esta opción se deshabilita.

Si selecciona esta opción, la información de clave pública se coloca dentro de la firma. De lo contrario, la información de clave no se incluye en la firma. El incluir la información de clave tiene la ventaja de que el certificado propiamente dicho (en concreto, su información de clave pública) no será necesario para el proceso de verificación (puesto que la información de clave está presente en la firma).

5.13.2 Verificar firmas XML

La firma XML se verifica correctamente si no se realizaron cambios en el archivo XML desde que se firmó. Si se realizaron cambios, la verificación fallará. En XMLSpy las firmas XML se pueden verificar en varias circunstancias:

- El archivo XML contiene una firma basada en un certificado y la información de clave está incluida en la firma 416
- El archivo XML contiene una firma basada en un certificado y la información de clave no está incluida en la firma 416
- La firma está basada en un certificado y está en un archivo externo que incluye la información de clave 417
- La firma está basada en un certificado y está en un archivo externo que no incluye la información de clave 418
- El archivo XML contiene una firma basada en una contraseña 418
- La firma está basada en una contraseña y está en un archivo externo 418

El archivo XML contiene una firma basada en un certificado y la información de clave está incluida en la firma

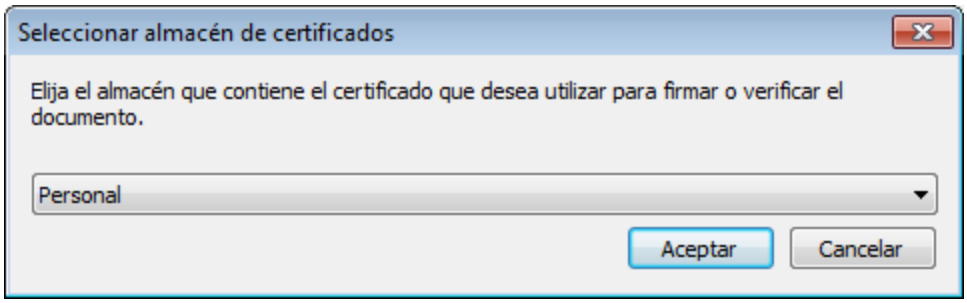
Para verificar una firma XML de estas características, abra el archivo XML en XMLSpy y asegúrese de que este archivo es el documento activo. Al hacer clic en el comando **XML | Verificar la firma XML** se ejecuta el proceso de verificación y el resultado aparece en la ventana de mensajes (aparece un mensaje que confirma si la verificación se realizó correctamente o si se produjo un error de verificación).

El archivo XML contiene una firma basada en un certificado y la información de clave no está incluida en la firma

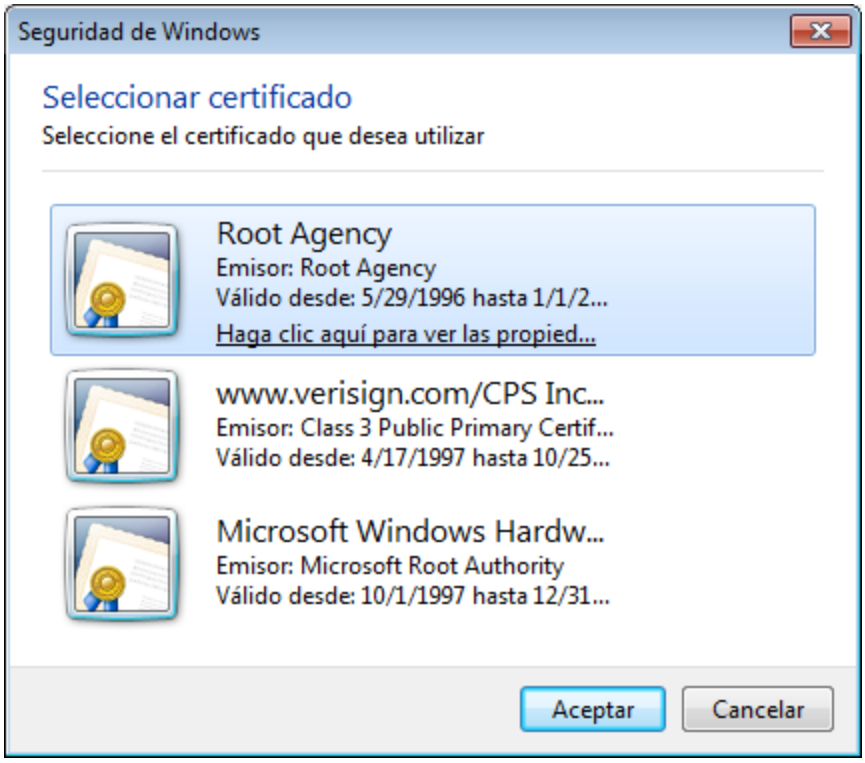
Si una firma basada en un certificado no incluye la información de clave, XMLSpy solicita el certificado donde se puede leer la información de clave necesaria para verificar la firma. Para verificar la firma es necesario que el archivo XML sea el archivo activo en XMLSpy. Al hacer clic en el comando **XML | Verificar la firma XML**,

© 2016-2022 Altova GmbH

aparece un diálogo donde debe seleccionar el almacén de certificados [419] donde está almacenado el certificado (*ver imagen siguiente*).



Después de seleccionar el almacén de certificados [419], haga clic en **Aceptar**. Aparece un cuadro de diálogo con todos los certificados del almacén seleccionado (*ver imagen siguiente*). Seleccione el certificado que desea usar en el proceso de verificación y haga clic en **Aceptar**.



A continuación se ejecuta el proceso de verificación y el resultado aparece en la ventana de mensajes.

La firma está basada en un certificado y está en un archivo externo que incluye la información de clave

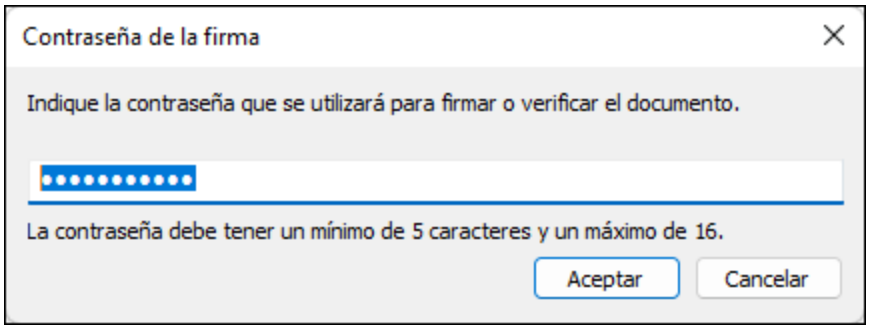
Si una firma XML basada en un certificado está en un archivo externo, la firma se verifica con el archivo de firma que esté activo en XMLSpy. Al hacer clic en el comando **XML | Verificar la firma XML** se ejecuta el proceso de verificación y el resultado aparece en la ventana de mensajes (aparece un mensaje que confirma si la verificación se realizó correctamente o si se produjo un error de verificación).

La firma está basada en un certificado y está en un archivo externo que no incluye la información de clave

Si una firma XML basada en un certificado está en un archivo externo, la firma se verifica con el archivo de firma que esté activo en XMLSpy. Al hacer clic en el comando **XML | Verificar la firma XML**, XMLSpy solicita el certificado donde se puede leer la información de clave necesaria para verificar la firma. Seleccione el certificado (siguiendo los pasos descritos en el subapartado *El archivo XML contiene una firma basada en un certificado y la información de clave no está incluida en la firma*416). El proceso de verificación se ejecuta y el resultado aparece en la ventana de mensajes (aparece un mensaje que confirma si la verificación se realizó correctamente o si se produjo un error de verificación).

El archivo XML contiene una firma basada en una contraseña

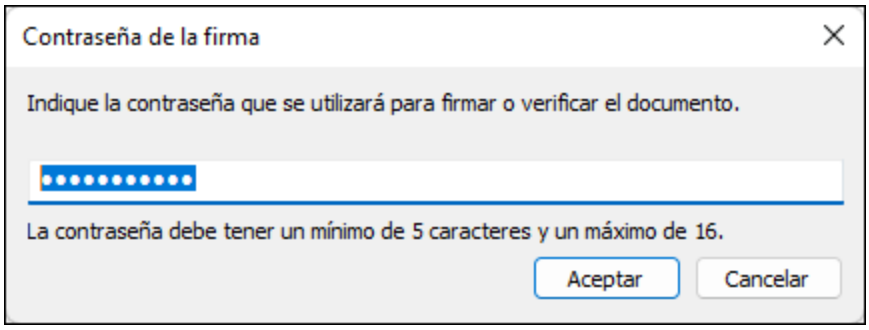
Si el archivo XML tiene una firma basada en una contraseña, la firma se verifica con el archivo XML que esté activo en XMLSpy. Al hacer clic en el comando **XML | Verificar la firma XML** aparece un cuadro de diálogo donde debe introducir la contraseña (*ver imagen siguiente*).



Introduzca la contraseña, que debe tener un mínimo de cinco caracteres y un máximo de 16, y haga clic en **Aceptar**. El proceso de verificación se ejecuta y el resultado aparece en la ventana de mensajes (aparece un mensaje que confirma si la verificación se realizó correctamente o si se produjo un error de verificación).

La firma está basada en una contraseña y está en un archivo externo

Si una firma XML basada en una contraseña está en un archivo externo, la firma se verifica con el archivo de firma que esté activo en XMLSpy. Al hacer clic en el comando **XML | Verificar la firma XML** aparece un cuadro de diálogo donde debe introducir la contraseña (*ver imagen siguiente*).



© 2016-2022 Altova GmbH

Introduzca la contraseña, que debe tener un mínimo de cinco caracteres y un máximo de 16, y haga clic en **Aceptar**. El proceso de verificación se ejecuta y el resultado aparece en la ventana de mensajes (aparece un mensaje que confirma si la verificación se realizó correctamente o si se produjo un error de verificación).

5.13.3 Trabajar con certificados

Los certificados de autorización se suelen utilizar para crear y verificar firmas XML. Este apartado contiene información sobre cómo se pueden obtener, importar y exportar certificados. Se divide en los subapartados siguientes:

- Obtener un certificado con un par de claves (privada/pública) 419
- Importar un certificado de clave privada y clave pública 419
- Los almacenes de certificados en equipos Windows 420
- Exportar un certificado de clave pública 421

Obtener un certificado con un par de claves (privada/pública)

Los certificados se pueden obtener de dos maneras distintas:

- *De una entidad de certificación:* la entidad de certificación verifica la identidad del propietario del certificado. Los certificados obtenidos de esta manera son distintos de los certificados autofirmados, que pueden crearse desde cero con una herramienta de creación de certificados.
- *Creando un certificado autofirmado:* aunque no están verificados por ninguna entidad, estos certificados suelen ofrecer suficiente seguridad. Hay varias herramientas de creación de certificados que puede usar (Microsoft Visual Studio, por ejemplo).

Si desea usar el certificado con firmas XML, es necesario que el certificado tenga un par de claves (privada/pública).

Nota: la función de creación y verificación de firmas XML de XMLSpy es compatible con certificados de tipo RSA-SHA1, DSA-SHA1 y SHA-256.

Importar un certificado de clave privada y clave pública

Tras obtener un certificado de clave privada y clave pública, debe importar el certificado al almacén de certificados de Windows. Siga estos pasos:

1. Haga doble clic en el archivo del certificado para abrir el Asistente para importación de certificados (*ver imagen siguiente*) y haga clic en **Siguiente**.



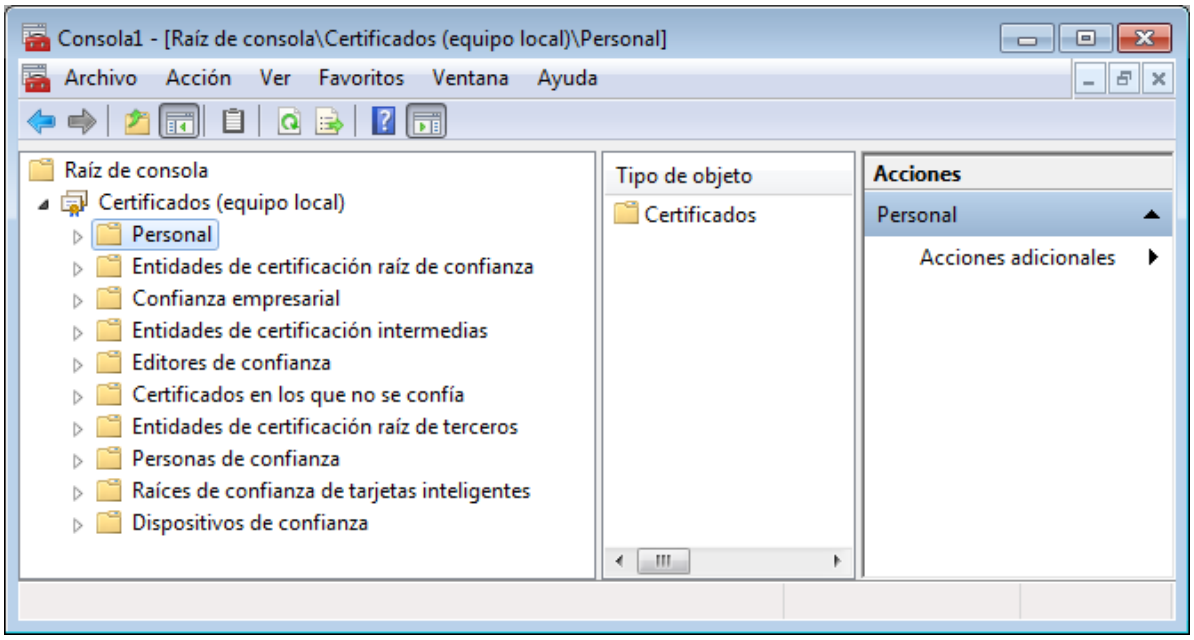
2. En la pantalla Archivo para importar, asegúrese de que el archivo del certificado está seleccionado y haga clic en **Siguiente**.
3. En la siguiente pantalla, escriba la contraseña para la clave privada. Tenga en cuenta que deberá recordar la contraseña si desea utilizar la clave privada para crear una firma XML. Cuando obtenga el certificado, recibirá la contraseña para la clave privada. Después de escribir la contraseña, haga clic en **Siguiente**.
4. En la siguiente pantalla puede configurar el asistente para que seleccione automáticamente el almacén donde se debe colocar el certificado (de acuerdo con el tipo de certificado) o puede seleccionar el almacén usted mismo. (Quizás sea mejor que seleccione el almacén usted mismo y así será más fácil recordar la ubicación del certificado). Haga clic en **Siguiente**.
5. Para terminar el proceso haga clic en **Finalizar**.

Almacenes de certificados en equipos Windows

Para tener acceso al almacén de certificados en un equipo Windows 7, siga estos pasos:

1. En el menú **Inicio**, escriba `mmc` en el cuadro de **Inicio** y pulse **Entrar**.
2. Aparece una ventana de consola (*ver imagen siguiente*).

© 2016-2022 Altova GmbH



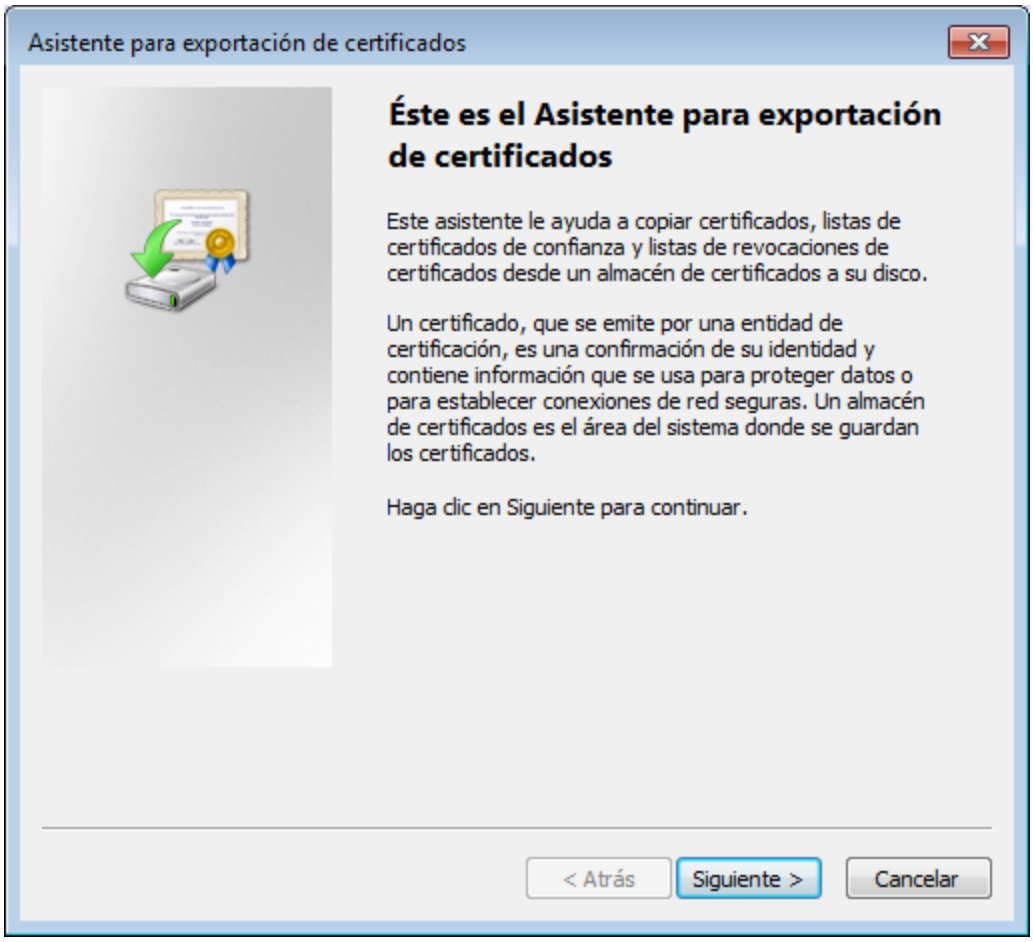
3. En la ventana de consola, seleccione el comando **Archivo | Agregar o quitar complemento**.
4. Aparece el cuadro de diálogo "Agregar o quitar complementos". Seleccione *Certificados* en este cuadro de diálogo y haga clic en **Agregar**.
5. Aparece el cuadro de diálogo "Complemento Certificados". Seleccione la opción *Cuenta de equipo* y haga clic en **Siguiente**.
6. En la siguiente pantalla del cuadro de diálogo, seleccione la opción *Equipo local* y después haga clic en **Finalizar**.
7. Haga clic en **Aceptar** para cerrar el cuadro de diálogo "Agregar o quitar complementos".
8. La "Raíz de consola" de la ventana de consola ahora contiene un componente llamado "Certificados" (*ver imagen anterior*). Este componente contiene los almacenes de certificados de su equipo.
9. Guarde la consola como archivo de Microsoft Management Console (archivo `.msc`). Para ello seleccione el comando **Archivo | Guardar** de la ventana de consola. Posteriormente podrá usar este archivo MSC para acceder a los almacenes de certificado de su equipo (con el comando **Archivo | Abrir** de una ventana de consola).

Exportar un certificado de clave pública

Si tiene un certificado con clave privada y pública, puede exportar este certificado sólo con una clave pública. Este certificado de clave pública se puede enviar después a otros usuarios, quienes podrán usarla para verificar firmas creadas con la clave privada del certificado.

Para exportar un certificado de clave pública desde un certificado de clave pública/privada, siga estos pasos:

1. Abra los almacenes de certificados en una ventana de consola. Para ello: (i) escriba `mmc` en el cuadro del menú **Inicio**. (ii) En la ventana de consola que aparece, seleccione **Archivo | Abrir** y seleccione el archivo MSC en el que se guardaron los almacenes de certificados (*ver el apartado anterior*).
2. Busque el certificado que desea exportar como certificado de clave pública y haga clic en él con el botón derecho.
3. Seleccione el comando **Todas las tareas | Exportar**. Aparece el Asistente para exportación de certificados (*ver imagen siguiente*).



4. Haga clic en **Siguiente**.
5. En la pantalla Exportar la clave privada, seleccione la opción *No exportar la clave privada* y haga clic en **Siguiente**.
6. En la pantalla Formato de archivo de exportación, seleccione el formato que desea usar (si no está seguro, utilice la opción predeterminada *DER*) y haga clic en **Siguiente**.
7. En la pantalla Archivo que se va a exportar, busque la ubicación donde desea guardar el archivo y escriba el nombre del archivo (sin la extensión de archivo, porque se anexará automáticamente). Después haga clic en **Siguiente**.
8. Haga clic en **Finalizar** para terminar la exportación.

Como resultado se crea un certificado de clave pública en la ubicación especificada en el proceso anterior. Puede enviar este certificado de clave pública a los destinatarios de archivos XML firmados con la clave privada correspondiente. El destinatario puede importar este certificado de clave pública a un almacén de certificados de su equipo y usar la clave pública de este certificado para verificar las firmas XML.

© 2016-2022 Altova GmbH

5.14 Características adicionales

Mientras trabaja con archivos XML puede disfrutar de estas características adicionales:

- Codificación 423
- Generar archivos DTD y esquemas XML 423
- Buscar y reemplazar 423
- Evaluar expresiones XPath 424
- Importar y exportar archivos de texto 424

Codificación

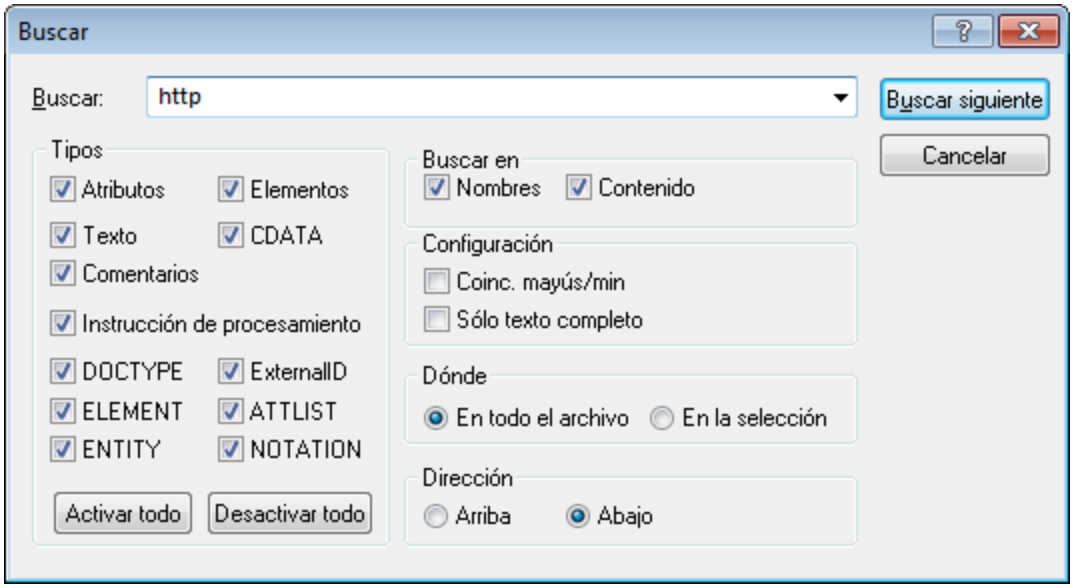
La codificación de los archivos XML (y de otros tipos de documentos) se puede configurar con el comando **Archivo | Codificación** 1268. La codificación predeterminada de los archivos XML y no XML se puede configurar en la sección Herramientas | Opciones | Codificación 1589.

Generar archivos DTD y esquemas XML

Si desea crear un esquema que describa la estructura de un documento XML, utilice el comando de menú **DTD/Esquema | Generar DTD o esquema** 1358. Aparece el cuadro de diálogo "Generar DTD o esquema", donde puede elegir si se genera una DTD o un esquema XML y otras opciones propias de los esquemas XML (p. ej. si se deben generar enumeraciones a partir de los valores incluidos en el documento XML).

Buscar y reemplazar

Las funciones Buscar 1289 y Reemplazar 1293 (menú **Edición**) ofrecen potentes posibilidades. El término de búsqueda se puede configurar según varias opciones (p. ej. coincidencia con mayúsculas y minúsculas, coincidencia con palabras enteras, etc.) y también se puede buscar como expresión regular. El intervalo de búsqueda se puede limitar a parte del documento o a determinados tipos de nodos (*ver imagen siguiente*).



La imagen anterior muestra el cuadro de diálogo "Buscar" de la vista Cuadrícula. El cuadro de diálogo y sus opciones varían dependiendo de qué vista esté activa. Para ver la función de búsqueda y reemplazo de la vista Texto consulte este apartado 157.

Evaluar expresiones XPath

Las expresiones XPath, que se escriben en la ventana XPath/XQuery, se pueden evaluar con respecto al documento XML activo. Los resultados de la evaluación aparecen en la ventana XPath/XQuery y al hacer clic en un nodo de los resultados, dicho nodo se resalta en el documento en la vista principal. Recuerde que puede activar la ventana XPath con el comando **XML | Evaluar XPath**.

Importar y exportar archivos de texto

Los datos de texto se pueden importar desde otros formatos de aplicación o exportar a otros formatos. Los comandos correspondientes a estas funciones están en el menú **Conversión** 1452.

© 2016-2022 Altova GmbH

6 DTD y esquemas XML

Sitio web de Altova: Editor XSD

Esta sección explica a grandes rasgos cómo trabajar con archivos DTD[442] y esquemas XML[444]. También describe SchemaAgent[463] y la potente función de Búsqueda en esquemas[475]. Además de ofrecer prácticas funciones de edición, XMLSpy incluye potentes funciones relacionadas con archivos DTD y esquemas XML:

Mecanismo de catalogación

La compatibilidad con el mecanismo de catalogación[456] OASIS permite el redireccionamiento de identificadores URI a direcciones locales, lo cual facilita el uso en varias estaciones de trabajo diferentes.

Reglas de esquema

A un esquema XML se le puede asignar un conjunto de restricciones adicionales definidas por el usuario. XMLSpy incluye un editor de reglas de esquema con el que puede crear y editar conjuntos de reglas de esquema para esquemas XML.

Subconjuntos de esquema

En la vista Esquema puede crear un esquema con los componentes de otro esquema de mayor tamaño. Estos subconjuntos de esquema más pequeños se pueden incluir después en un esquema más grande. La operación inversa, conocida cono nivelación de esquemas, coloca los componentes de archivos incluidos directamente en el esquema más grande. Para más información sobre cómo generar subconjuntos de esquemas y cómo nivelar esquemas, consulte el apartado Subconjuntos de esquemas[445].

Convertir DTD en esquema XML y viceversa

Con los comandos del menú **DTD/Esquema**[1355] puede convertir un documento DTD en un esquema XML y viceversa y ambos tipos de documentos se pueden nivelar. Cuando se nivela un documento DTD, los componentes de los módulos incluidos/importados se guardan en el archivo primario directamente y los componentes no utilizados se eliminan.

Generar archivo XML de muestra

Con el comando de menú **DTD/Esquema | Generar archivo XML de muestra**[1365] puede generar un documento XML de base, que se basa en el archivo DTD o esquema XML activo. Esta función es muy práctica y permite crear un archivo XML basado en el esquema activo con gran rapidez.

Ir a la definición

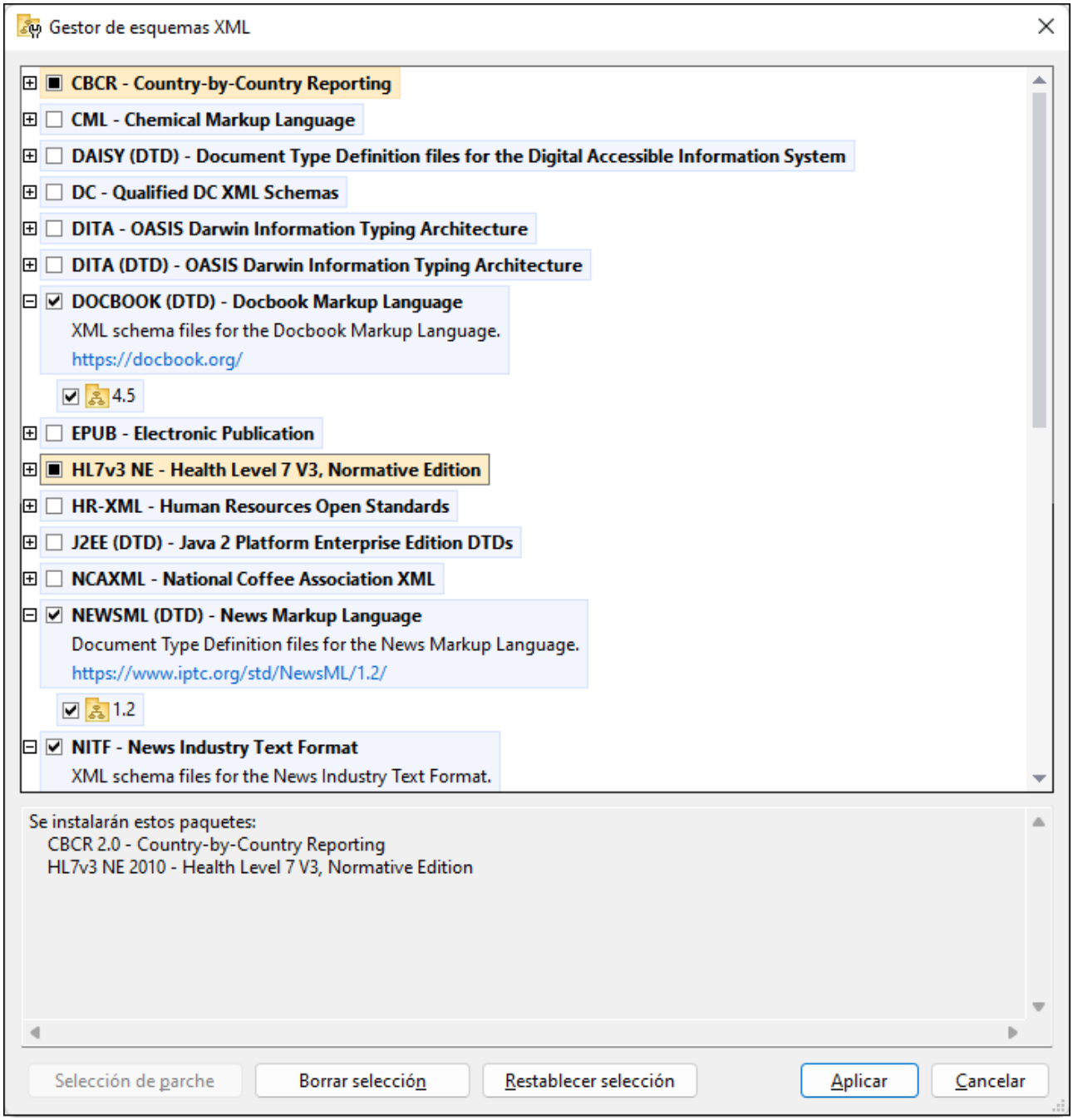
Cuando el cursor se sitúa dentro de un nodo de un documento XML, al hacer clic en el comando **DTD/Esquema | Ir a la definición**[1357], se abre el archivo de esquema y en él se resalta la definición del nodo XML seleccionado.

6.1 Gestor de esquemas

El Gestor de esquemas XML es una herramienta que ofrece una forma centralizada de instalar y administrar esquemas XML (DTDs para XML y esquemas XML) para usarlos en todas las aplicaciones de Altova compatibles con XML, incluido XMLSpy.

- En Windows, el gestor tiene una interfaz gráfica del usuario (*imagen siguiente*) a la que también puede acceder desde la línea de comandos. (Las aplicaciones de escritorio de Altova solo están disponibles para Windows; *consulte la lista siguiente*.)
- En Linux y macOS el Gestor de esquemas solo está disponible en la línea de comandos. (Las aplicaciones de escritorio de Altova están disponibles para Windows, Linux y macOS; *consulte la lista siguiente*.)

© 2016-2022 Altova GmbH



Aplicaciones de Altova que funcionan con el Gestor de esquemas

Aplicaciones de escritorio (solo para Windows)	Aplicaciones de servidor (Windows, Linux, macOS)
XMLSpy (todas las ediciones)	RaptorXML Server, RaptorXML+XBRL Server
MapForce (todas las ediciones)	StyleVision Server

StyleVision (todas las ediciones)	

Instalación y desinstalación del Gestor de esquemas

El Gestor de esquemas se instala automáticamente al instalar cualquiera de las aplicaciones de Altova compatibles con XML o el Altova Mission Kit Enterprise Edition (*véase la tabla de más arriba*).

También se elimina automáticamente si desinstala todas las aplicaciones de Altova compatibles con XML o el Altova Mission Kit Enterprise Edition del equipo.

Características de Gestor de esquemas

El Gestor de esquemas permite:

- Ver los esquemas XML que hay instaladas en su equipo y comprobar si hay versiones nuevas para descargar.
- Descargar las versiones más recientes de los esquemas XML independientemente del ciclo de versiones de Altova. Altova guarda todos los esquemas en un sistema de almacenamiento en línea al que tiene acceso el Gestor de esquemas y desde donde puede descargarlas tan pronto como estén disponibles.
- Instalar o desinstalas cualquiera de las múltiples versiones de un esquema en concreto (o todas ellas, si las necesita).
- Un solo esquema XML representa un "paquete", pero puede tener dependencias en otros esquemas. Al instalar o desinstalar un esquema, se detectan e instalan o desinstalan también automáticamente todas sus dependencias. La interfaz gráfica del usuario (o la línea de comandos, en su caso) le informa cuando se añaden o eliminan esquemas.
- Los esquemas XML administradas con el Gestor de esquemas pueden usar el catálogo XML, que permite resolver referencias a URI en documentos de instancia o esquema desde archivos locales, en vez de a través de Internet.
- Todos los esquemas principales están incluidos en Gestor de esquemas y se actualizan de forma periódica a la versión más reciente. De esta forma puede administrar todos los esquemas desde un punto común y tenerlos siempre listos para las aplicaciones de Altova que las usan.
- Los cambios que se realizan en el Gestor de esquemas afectan a todos los productos de Altova que estén instalados en ese equipo.

Funcionamiento

Altova mantiene un almacenamiento en línea donde guarda todos los esquemas XML de los productos de Altova. Este almacenamiento se actualiza de forma periódica, por ejemplo, poco después de que las organizaciones correspondientes publiquen las versiones nuevas de los esquemas respectivos. Al ejecutar Gestor de esquemas desde la interfaz gráfica del usuario aparece información sobre los esquemas más recientes disponibles en un cuadro de diálogo en el que puede visualizarlos, instalarlos, actualizarlos o desinstalarlos.

También puede instalar los esquemas de otra manera. En el sitio web de Altova (https://www.altova.com/schema-manager) puede seleccionar el esquema y los esquemas dependientes de este que quiere instalar. El sitio web prepara un archivo de tipo `.altova_xmlschemas` que puede descargar y que contiene la información sobre los esquemas seleccionados. Al hacer doble clic en este archivo o pasarlo a Gestor de esquemas desde la línea de comandos como argumento del comando `install` 437, Gestor de esquemas instala los esquemas que contiene.

© 2016-2022 Altova GmbH

Memoria caché local: seguimiento de esquemas
Independientemente de cómo se instalen los esquemas, toda la información sobre los esquemas instalados se almacena en una ubicación centralizada de su equipo, el directorio caché. El directorio caché local está en:

Windows	`C:\ProgramData\Altova\pkgs\.cache`
Linux	`/var/opt/Altova/pkgs\.cache`
macOS	`/var/Altova/pkgs`

El directorio caché local se actualiza automáticamente de vez en cuando para que el estado más actual del equipo corresponda con el del almacenamiento en línea. Más concretamente, el caché se actualiza:

- al ejecutar el Gestor de esquemas.
- al ejecutar XMLSpy por primera vez en un mismo día natural.
- si XMLSpy ya se está ejecutando, el directorio caché se actualiza cada 24 horas.
- también puede actualizar el caché local desde el almacenamiento en línea manualmente ejecutando el comando de actualización **update** 440 desde la línea de comandos.

Si instala o desinstala esquemas, el directorio caché local se actualiza automáticamente con información sobre los esquemas disponibles e instalados, además de con los propios archivos de esquema.

No modifique la memoria caché manualmente

El directorio caché local se mantiene automáticamente en base a los esquemas que instale o desinstale; no debe modificarlo ni eliminarlo manualmente. Si necesita restaurar el Gestor de esquemas a su estado original, ejecute el comando **reset** 438 desde la línea de comandos y después ejecute el comando **initialize** 436.

6.1.1 Ejecutar el gestor de esquemas

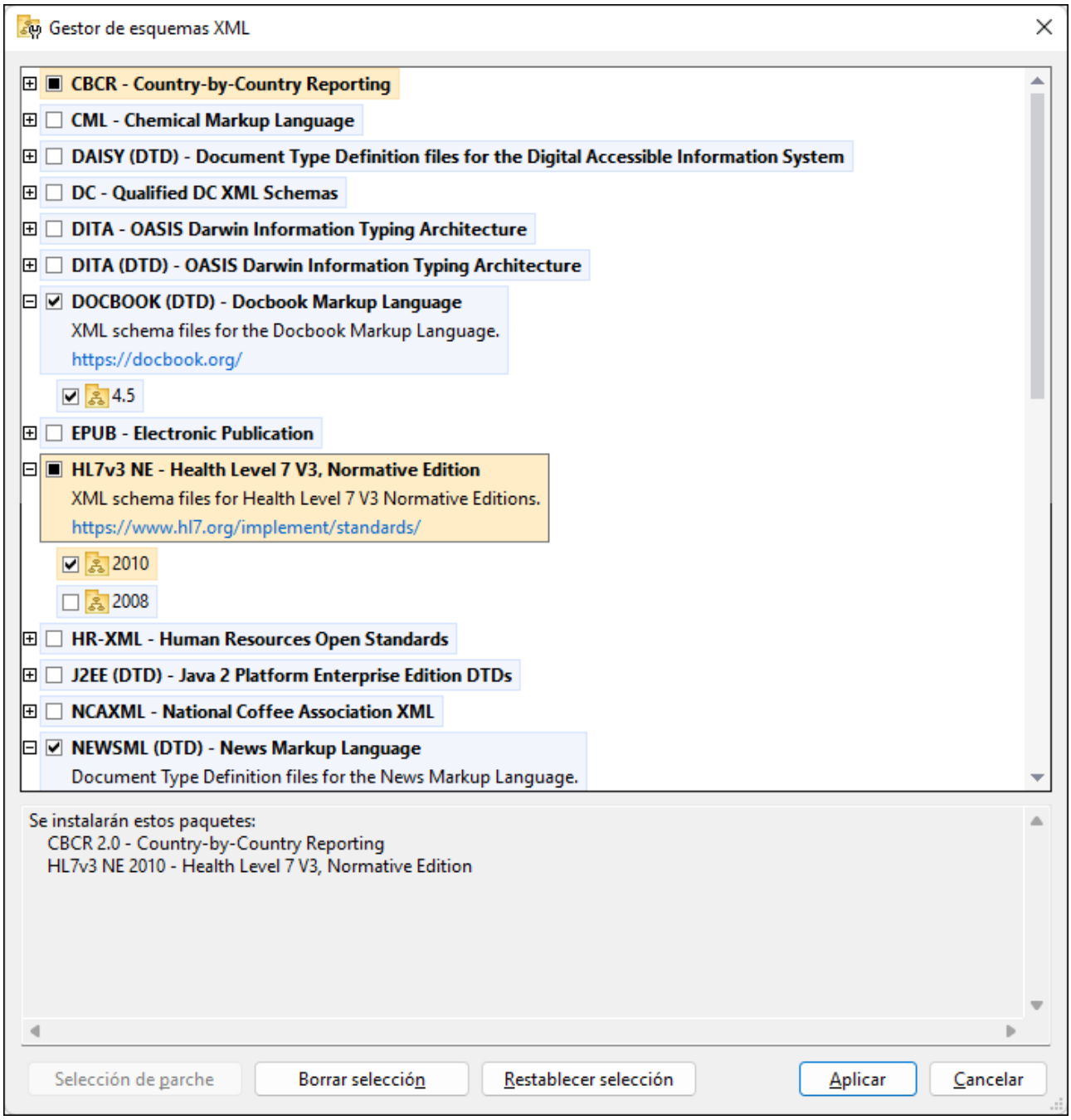
Interfaz gráfica del usuario

Hay varias formas de acceder a la IGU del Gestor de esquemas:

- *Durante la instalación de XMLSpy:* al final del proceso de instalación, seleccione la casilla *Invocar al Gestor de esquemas* para acceder directamente a la IGU del gestor de esquemas XML. Con él puede instalar esquemas durante el proceso de instalación de su aplicación de Altova.
- *Después de la instalación de XMLSpy:* una vez haya instalado la aplicación puede acceder al Gestor de esquemas en cualquier momento desde el comando de menú Herramientas | **Gestor de esquemas XML**.
- Mediante el archivo **`.altova_schemas`** que descargó del *sitio web de Altova:* haga doble clic en el archivo para ejecutar Gestor de esquemas, que instalará los esquemas que haya seleccionado.

Cuando se abra la IGU del Gestor de esquemas (*imagen siguiente*) podrá ver en ella los esquemas que ya se han instalado. Si quiere instalar más solo tiene que seleccionarlas, y al contrario si quiere desinstalar alguna. Una vez haya terminado, puede aplicar los cambios. los esquemas que se vayan a instalar o desinstalar

aparecerán resaltados y un mensaje le avisará de los cambios que está a punto de hacer en la ventana *Mensajes*, en la parte inferior de la ventana de Gestor de esquemas (*véase imagen*).



Interfaz de la línea de comandos

Para ejecutar el Gestor de esquemas desde una interfaz de la línea de comandos debe usar su archivo ejecutable, **`xmlschemamanager.exe`**.

© 2016-2022 Altova GmbH

Puede encontrar este archivo:

- *en Windows:* `C:\ProgramData\Altova\SharedBetweenVersions`
- *en Linux o macOS (solo para aplicaciones de servidor):* `%INSTALLDIR%/bin`, donde `%INSTALLDIR%` es el directorio de instalación del programa.

Puede usar cualquiera de los comandos de la referencia de la línea de comandos 435, a continuación.

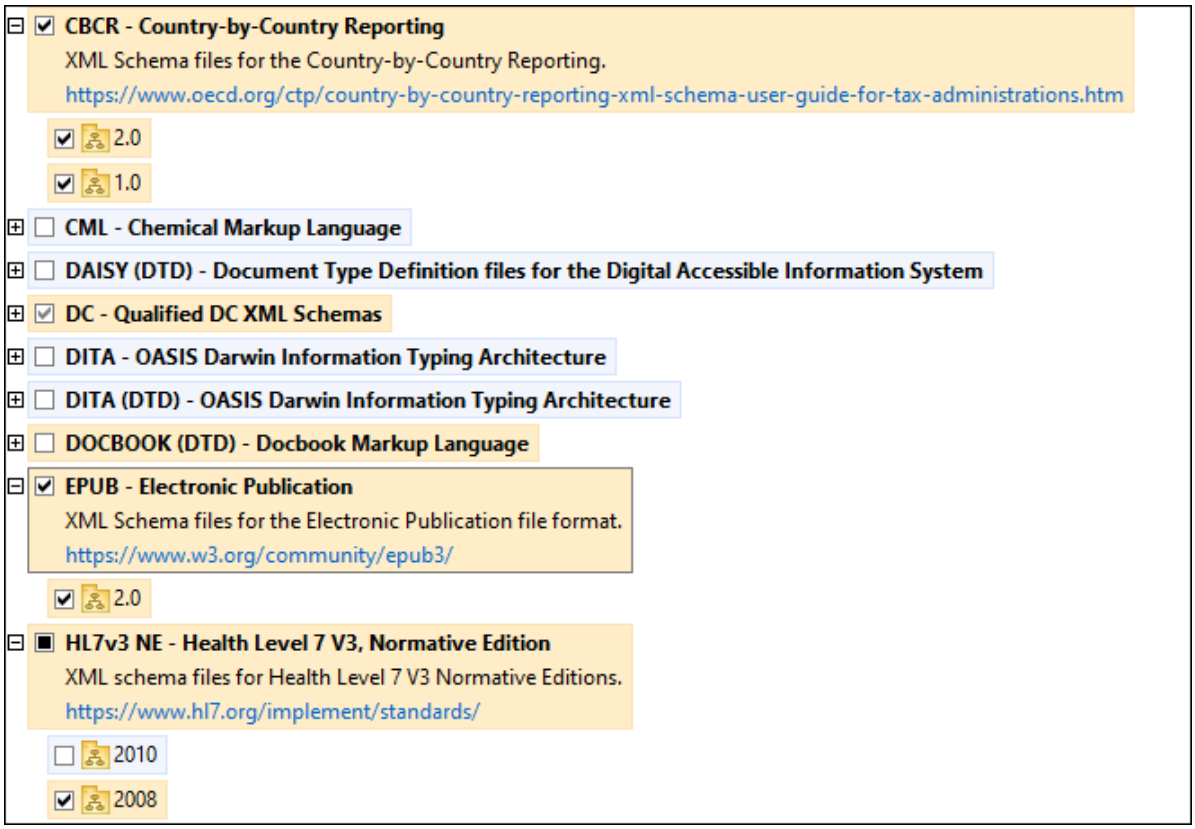
Para mostrar al ayuda de los comandos ejecute:

- *en Windows:* `xmlschemamanager.exe --help`
- *en Linux o macOS (solo para aplicaciones de servidor):* `sudo ./xmlschemamanager --help`

6.1.2 Categorías de estado

Gestor de esquemas diferencia los esquemas que administra entre:

- *Esquemas instalados:* estos aparecen en la IGU con sus casillas marcadas (*en la imagen siguiente las versiones marcadas de los esquemas EPUB y HL7v3 NE son las que están instaladas*). Si se seleccionan todas las versiones de un esquema, en la casilla del esquema aparece una marca de verificación. Si hay al menos un esquema sin seleccionar, en la en la casilla del esquema aparece un cuadrado negro. Para **desinstalar** un esquema debe desmarcar la casilla correspondiente (*en la imagen siguiente, el DTD DocBook está instalado y su casilla se ha desmarcado, es decir, se va a desinstalar*).
- *Esquemas disponibles no instalados:* estos aparecen en la IGU con las casillas correspondientes sin seleccionar. Para **instalar** esquemas, marque la casilla correspondiente.



- *Esquemas que pueden actualizarse:* son los que han sido revisados por sus emisores. Aparecen indicados en la IGU con el icono  (*imagen anterior*). Puede aplicar **parches** al esquema seleccionado con la revisión que esté disponible.

Puntos importantes

- En la imagen anterior se han marcado los esquemas CBCR. Las que tienen un fondo azul ya están instaladas. Las que tienen el fondo amarillo no están instaladas pero se han seleccionado para instalarlas. Observe que el esquema HL7v3 NE 2010 no está instalada ni se ha seleccionado para instalarlo.
- Al ejecutar el Gestor de esquemas desde la línea de comandos puede usar el comando `list` con distintas opciones para ver distintas categorías de esquemas:

`xmlschemamanager.exe list`	Muestra todas los esquemas instalados y disponibles; también indica qué esquemas se pueden actualizar
`xmlschemamanager.exe list -i`	Muestra solo los esquemas instalados; también indica qué esquemas se pueden actualizar
`xmlschemamanager.exe list -u`	Muestra qué esquemas se pueden actualizar

Nota: en Linux y macOS `use sudo ./xmlschemamanager list`

 © 2016-2022 Altova GmbH

6.1.3 Aplicar parches o instalar un esquema

Aplicar un parche a un esquema instalado

A veces los emisores de los esquemas XML generan parches. Cuando el Gestor de esquemas XML detecta que hay parches disponibles, estos aparecen en las listas de esquemas, desde donde puede instalarlos.

En la IGU

Los parches se indican con el icono . (*Consulte también el apartado anterior sobre categorías de esquemas*[431].) Si hay parches disponibles se habilita el botón **Seleccionar parches**. Haga clic en él para seleccionar y preparar los parches. En la IGU, el icono de los esquemas correspondientes cambia de  a  y el cuadro de diálogo le informa de qué parches se van a aplicar. Las listas del panel principal y las del panel Mensajes están ordenadas alfabéticamente, por lo que puede ver y revisar los esquemas antes de aplicar los parches. Una vez esté listo para instalar los parches seleccionados, haga clic en **Aplicar**.

En la línea de comandos

Para aplicar un parche desde lal ínea de comandos:

1. Ejecute el comando `list -u`[438]. Aparece una lista con los esquemas para las que hay parches disponibles.
1. Ejecute el comando `upgrade`[441] para instalar todos los parches.

Instalar un esquema disponible

Para instalar esquemas puede usar la IGU del Gestor de esquemas o enviar las instrucciones al Gestor de esquemas desde la línea de comandos.

Nota: si el esquema actual tiene dependencias en otros esquemas, también se instalan (o desinstalan, según el caso) los esquemas dependientes.

En la IGU

Para instalar esquemas con la IGU del Gestor de esquemas, seleccione los esquemas que quiere instalar y haga clic en **Aplicar**.

También puede seleccionar los esquemas que quiere instalar en el sitio web de Altova y generar desde allí un archivo **`.altova_schemas`**. Al hacer doble clic en este archivo se abre el Gestor de esquemas con los esquemas que indicó preseleccionados. Solo tiene que hacer clic en **Aplicar**.

En la línea de comandos

Para instalar esquemas desde la línea de comandos ejecute el comando **`install`**:

```
xmlschemamanager.exe install [opciones] Schema+
```

donde **`FILTER`** es el esquema (o los esquemas) que quiere instalar o un archivo **`.altova_schemas`**. Para hacer referencia a un esquema se usa un identificados con el formato **`<nombre>-<versión>`** que aparece junto a cada esquema que muestra el comando `list`[438]. Puede introducir tantos esquemas como quiera. Para más detalles consulte la descripción del comando `install`[437].

Nota: en Linux o macOS, use el comando **`sudo ./xmlschemamanager`**.

Instalar un esquema requerido
Si ejecuta un comando en XMLSpy y XMLSpy descubre que uno de los esquemas que necesita para ejecutar el comando falta o está incompleta, el Gestor de esquemas incluirá información sobre ese componente de esquema que falta. Entonces puede aplicar el parche indicado y/o instalar el esquema que falta.

Siempre puede ver todos los esquemas instalados previamente ejecutando el Gestor de esquemas desde **Herramientas | Gestor de esquemas**.

6.1.4 Desinstalar o restaurar esquemas

Desinstalar un esquema

Para desinstalar esquemas puede usar la IGU del Gestor de esquemas o enviar las instrucciones al Gestor de esquemas desde la línea de comandos.

Nota: si el esquema actual tiene dependencias en otros esquemas, también se instalan (o desinstalan, según el caso) los esquemas dependientes.

En la IGU
Para desinstalar esquemas con la IGU del Gestor de esquemas, seleccione los esquemas que quiere desinstalar y haga clic en **Aplicar**. Los esquemas seleccionadas y sus dependencias se desinstalarán.

Para desinstalar todos los esquemas haga clic en **Deseleccionar todas** y haga clic en **Aplicar**.

En la línea de comandos
Para desinstalar esquemas desde la línea de comandos ejecute el comando `uninstall` 439:

```
xmlschemamanager.exe uninstall [options] FILTER+
```

donde `FILTER` es el esquema (o los esquemas) que quiere desinstalar o un archivo `.altova_schemas`. Para hacer referencia a un esquema se usa un identificador con el formato `<nombre>-<versión>` que aparece junto a cada esquema que muestra el comando `list` 438. Puede introducir tantos esquemas como quiera. Para más detalles consulte la descripción del comando `uninstall` 439.

Nota: en Linux o macOS, use el comando `sudo ./xmlschemamanager`.

Restaurar el Gestor de esquemas

Puede restaurar el Gestor de esquemas, es decir, eliminar todos los esquemas instaladas, así como el directorio caché.

- En la IGU, haga clic en **Restaurar selección**.
- En la línea de comandos, use el comando `reset` 438.

© 2016-2022 Altova GmbH

Una vez haya ejecutado este comando, asegúrese de que ejecuta también el comando `initialize` [436] para recrear el directorio caché. También puede ejecutar el comando `reset` [438] con la opción `-i`.

Recuerde que `reset -i` [438] restaura la instalación original del producto, por lo que es recomendable ejecutar el comando `update` [440] después de restaurar el gestor. Puede ejecutar el comando `reset` [438] con las opciones `-i` o `-u`.

6.1.5 Interfaz de la línea de comandos (ILC)

Para llamar a Gestor de esquemas desde la línea de comandos necesita saber la ruta del ejecutable. Por defecto, el ejecutable del Gestor de esquemas se encuentra en:

```
C:\ProgramData\Altova\SharedBetweenVersions\XMLSchemaManager.exe
```

Nota: en los sistemas Linux y macOS una vez haya cambiado el directorio al que contiene el ejecutable, puede llamar al ejecutable con **`sudo ./xmlschemamanager`**.El prefijo **`./`** indica que el ejecutable está en el directorio actual. El prefijo **`sudo`** indica que el comando se debe ejecutar con derechos de administrador.

Sintaxis de la línea de comandos

La sintaxis general para usar la línea de comandos es:

```
<exec> -h | --help | --version | <command> [opciones] [argumentos]
```

En el código anterior la barra vertical **`|`** separa elementos que se excluyen mutuamente. Los corchetes **`[]`** indican elementos opcionales. Básicamente, puede teclear la ruta del ejecutable seguida por las opciones `--h`, `--help` o `--version`, o por un comando. Cada comando puede tener opciones y argumentos. Los comandos se describen en los apartados siguientes.

6.1.5.1 help

Este comando ofrece ayuda contextual sobre los comandos del ejecutable del Gestor de esquemas.

Sintaxis

```
<exec> help [command]
```

Donde `[command]` es un argumento opcional que indica cualquier nombre válido de comando.

Tenga en cuenta que:

- Puede invocar la ayuda tecleando un comando seguido por `--h` or `--help`, por ejemplo: **`<exec> list -h`**
- Puede invocar la ayuda general tecleando `--h` o `--help` directamente después del ejecutable, por ejemplo:

Ejemplo

Este comando muestra la ayuda del comando `list`:

```
xmlschemamanager help list
```

6.1.5.2 info

Este comando muestra información detallada sobre cada uno de los esquemas dados como argumento. Esta información incluye el título, la versión, la descripción, el editor y las referencias de las dependencias.

Sintaxis

```
<exec> info [options] Schema+
```

- El argumento **`Schema`** es el nombre de un esquema o parte del nombre de un esquema. (Para ver el ID de un paquete de esquemas y la información relativa a su estado de instalación use el comando `list`[438].)
- Use **`<exec> info -h`** para ver la ayuda sobre este comando en la línea de comandos..

Ejemplo

Este comando muestra información detallada sobre los esquemas **`DocBook-DTD`** y **`NITF`**:

```
xmlschemamanager info doc nitf
```

6.1.5.3 initialize

Este comando inicializa el entorno del Gestor de esquemas y crea un directorio caché donde se guardan todos los esquemas localmente. El Gestor de esquemas se inicializa automáticamente la primera vez que instale una aplicación de Altova compatible con él, por lo que normalmente no es necesario ejecutar este comando. Por lo general solo es necesario ejecutarlo después de haber ejecutado el comando `reset`.

Sintaxis

```
<exec> initialize | init [opciones]
```

Opciones
Estas son las opciones del comando **`initialize`**:

`--help, --h`	Muestra la ayuda sobre este comando en la línea de comandos.
`--silent, --s`	Muestra solamente los mensajes de error. El valor predeterminado es **`false`**.

© 2016-2022 Altova GmbH

`--verbose, --v`	Muestra información suplementaria durante la ejecución. El valor predeterminado es **`false`**.

Ejemplo

Este comando inicializa el Gestor de esquemas:

```
xmlschemamanager initialize
```

6.1.5.4 install

Este comando instala una o más esquemas.

Sintaxis

```
<exec> install [options] Schema+
```

Para indicar varios esquemas, repita el argumento `Schema` tantas veces como sea necesario.

El argumento de `Schema` puede ser:

1. Un identificador de esquema en el formato **`<name>-<version>`**, por ejemplo: **`cbcr-2.10`**). Para ver todos los identificadores de esquemas y sus versiones ejecute el comando `list` 438. También puede usar el nombre de el esquema abreviado, si este es único, por ejemplo **`docbook`**. Si usa una abreviación del nombre se desinstalan todos los esquemas que contengan esa abreviación.
2. La ruta de acceso a un archivo **`.altova_schemas`** descargado desde el sitio web de Altova. Para más información sobre estos archivos consulte la *Introducción al Gestor de esquemas: funcionamiento* 426.

Opciones

Estas son las opciones del comando **`install`**:

`--help, --h`	Muestra la ayuda sobre este comando en la línea de comandos.
`--silent, --s`	Muestra solamente los mensajes de error. El valor predeterminado es **`false`**.
`--verbose, --v`	Muestra información suplementaria durante la ejecución. El valor predeterminado es **`false`**.

Ejemplo

Este comando instala el esquema CBCR 2.0 (Country-By-Country Reporting) y el DTD DocBook más reciente:

```
xmlschemamanager install cbcr-2.0 docbook
```

6.1.5.5 list

Use este comando para ver los esquemas del Gestor de esquemas; tiene varias opciones:

- lista de todos los esquemas disponibles
- lista de esquemas específicos
- lista de los esquemas instalados
- lista de los esquemas que se pueden actualizar.

Sintaxis

```
<exec> list | ls [options] Schema?
```

Si no se indica ningún argumento **Schema** la lista incluye todos los esquemas. De lo contrario la lista incluye los esquemas indicados en las opciones (*véase el ejemplo de más abajo*). Recuerde que puede usar el argumento **Schema** tantas veces como quiera.

Opciones
Estas son las opciones del comando **list**:

`--help, --h`	Muestra la ayuda sobre este comando en la línea de comandos.
`--installed, --i`	Muestra solamente los esquemas instaladas. El valor predeterminado es **false**.
`--upgradeable, --u`	Muestra solamente los esquemas para las que hay disponible una versión más reciente (parches). El valor predeterminado es **false**.

Ejemplos

- Para ver todos los esquemas disponibles ejecute: **xmlschemamanager list**
- Para ver solamente los esquemas instaladas ejecute: **xmlschemamanager list -i**
- Para ver todos los esquemas cuyos nombres contienen "doc" o "nitf" ejecute: **xmlschemamanager list doc nitf**

6.1.5.6 reset

Este comando elimina todos los esquemas instalados, así como el directorio de caché. Este comando elimina todos los esquemas instalados y su información. Una vez haya ejecutado este comando, asegúrese de que ejecuta el comando initialize 436 para volver a crear el directorio de caché. También puede ejecutar el comando `reset` con la opción `-i`. Tenga en cuenta que `reset -i` restaura la instalación original del producto, por lo que se recomienda ejecutar también el comando update 440 después de una restauración. También puede ejecutar el comando `reset` con las opciones `-i` y `-u`.

Sintaxis

```
<exec> reset [opciones]
```


© 2016-2022 Altova GmbH

Opciones
Estas son las opciones del comando **reset**:

`--help, --h`	Muestra la ayuda sobre este comando en la línea de comandos.
`--init, --i`	Inicializa el entorno del Gestor de esquemas XML después de una restauración. El valor predeterminado es **false**.
`--silent, --s`	Muestra solamente los mensajes de error. El valor predeterminado es **false**.
`--update, --u`	Inicializa y actualiza el entorno del Gestor de esquemas XML después de una restauración. El valor predeterminado es **false**.
`--verbose, --v`	Muestra información suplementaria durante la ejecución. El valor predeterminado es **false**.

Ejemplos

- Para restaurar el Gestor de esquemas, ejecute: **xmlschemamanager reset**
- Para restaurar el Gestor de esquemas e inicializarlo, ejecute: **xmlschemamanager reset -i**
- Para restaurar el Gestor de esquemas, inicializarlo y actualizar la lista de esquemas, ejecute: **xmlschemamanager reset -i -u**

6.1.5.7 uninstall

Este comando desinstala una o más esquemas. Por defecto, cualquier esquema a la que haga referencia el esquema actual también se desinstala. Para desinstalar solamente el esquema actual y mantener aquellas a las que se hace referencia, use la opción `--k`.

Sintaxis

```
<exec> uninstall [opciones] Schema+
```

Para indicar varios esquemas, repita `FILTER` tantas veces como sea necesario.

El argumento de **Schema** puede ser:

- Un identificador de esquema en el formato **<name>-<version>**, por ejemplo: **eba-2.10**). Para ver todos los identificadores de esquemas y sus versiones ejecute el comando `list -i` 438. También puede usar el nombre del esquema abreviado, si este es único, por ejemplo **eba**. Si usa una abreviación del nombre se desinstalan todos los esquemas que contengan esa abreviación.
- La ruta de acceso a un archivo **.altova_taxonomies** descargado desde el sitio web de Altova. Para más información sobre estos archivos consulte la *Introducción al Gestor de esquemas: funcionamiento* 426.

Opciones
Estas son las opciones del comando **uninstall**:

--help, --h	Muestra la ayuda sobre este comando en la línea de comandos.
--keep-references, --k	Si usa esta opción, los esquemas referenciados no se desinstalan. El valor predeterminado es **false**.
--silent, --s	Muestra solamente los mensajes de error. El valor predeterminado es **false**.
--verbose, --v	Muestra información suplementaria durante la ejecución. El valor predeterminado es **false**.

Ejemplo
Este comando desinstala los esquemas CBCR 2.0 y EPUB 2.0:

```
xmlschemamanager uninstall cbcr-2.0 epub-2.0
```

Este comando desinstala el esquema **eba-2.10** pero no los esquemas a los que hace referencia:

```
xmlschemamanager uninstall --k cbcr-2.0
```

6.1.5.8 update

Este comando consulta la lista de esquemas disponibles en el almacenamiento en línea y actualiza el directorio de caché local. Esta información se actualiza de forma implícita, por lo que no es necesario ejecutar este comando a no ser que haya ejecutado reset 438 e initialize 436.

Sintaxis
```
<exec> update [opciones]
```

Opciones
Estas son las opciones del comando update:

--help, --h	Muestra la ayuda sobre este comando en la línea de comandos.
--silent, --s	Muestra solamente los mensajes de error. El valor predeterminado es **false**.
--verbose, --v	Muestra información suplementaria durante la ejecución. El valor predeterminado es **false**.

Ejemplo
Este comando actualiza la lista de esquemas:

```
xmlschemamanager update
```


© 2016-2022 Altova GmbH

6.1.5.9 upgrade

Este comando actualiza todos los esquemas aptos para la versión *parche* más reciente disponible. Puede identificar cuáles lo son con el comando `list -u`.

Nota: el comando `upgrade` eliminaría un esquema obsoleto si no hay ninguna versión disponible.

Sintaxis

```
<exec> upgrade [opciones]
```

Opciones
Estas son las opciones del comando `upgrade`:

`--help, --h`	Muestra la ayuda sobre este comando en la línea de comandos.
`--silent, --s`	Muestra solamente los mensajes de error. El valor predeterminado es **`false`**.
`--verbose, --v`	Muestra información suplementaria durante la ejecución. El valor predeterminado es **`false`**.

6.2 Archivos DTD

En la vista Texto los documentos DTD se presentan con color de sintaxis. Puede configurar la vista predeterminada en la que se deben abrir los documentos XSLT en la sección Tipos de archivo[1587] del cuadro de diálogo "Opciones".

Vista Texto

En la vista Texto los documentos DTD se presentan con color de sintaxis. A continuación aparece un fragmento de una DTD de muestra:

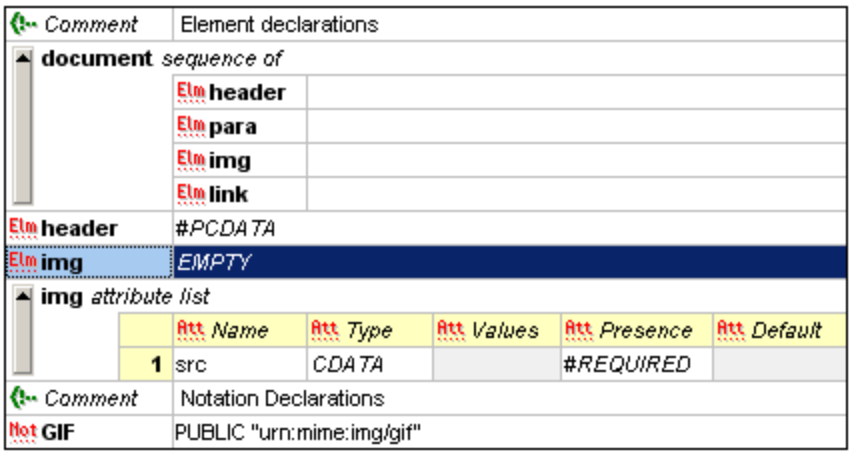
```
<!-- declaraciones de elementos -->
<!ELEMENT document (header, para, img, link)>
<!ELEMENT header (#PCDATA)>
<!ELEMENT img EMPTY>
  <!ATTLIST img
    src CDATA #REQUIRED
  >

<!-- declaraciones de notaciones -->
<!NOTATION GIF PUBLIC "urn:mime:img/gif">
```

La sangría se indica con guías de sangría (pulsando la tecla de tabulación). En el cuadro de diálogo Configurar la vista Texto[149] puede indicar el número de tabulaciones que se debe usar para la sangría.

Vista Cuadrícula

En la vista Texto los documentos DTD se presentan en forma de tabla. A continuación aparece la misma DTD del ejemplo anterior, ahora en la vista Cuadrícula.



Editar la estructura DTD

- Cuando el cursor está dentro de una celda puede insertar o anexar nodos o añadir un nodo secundario con el menú contextual o el menú **XML**.
- Haga clic en el icono de tipo de nodo, en la parte superior izquierda de la celda, para cambiar el tipo de nodo.

© 2016-2022 Altova GmbH

- Cambie el modelo de contenido (*secuencia*, *mixto*, *vacío*, etc.) y el modificador de ocurrencias (*exactamente 1*, *1 o más*, etc.) de un nodo haciendo clic en el icono correspondiente y eligiendo la opción que prefiera.
- También puede usar las funciones de arrastrar y colocar para mover nodos dentro del documento, así como copiar y pegar para mover nodos a otras ubicaciones.

Editar valores DTD

- Para editar valores como nombres de elemento o atributo y comentarios, haga doble clic en la celda y edítela.

Barra de herramientas de la vista Cuadrícula
La barra de herramientas Vista Cuadrícula permite acceder al cuadro de diálogo de opciones de configuración de la vista y contiene comandos como el acercamiento o el ajuste automático de línea.

Funciones de XMLSpy relacionadas con documentos DTD

XMLSpy ofrece estas prácticas funciones:

- *Convertir DTD en esquema XML:* con el **comando DTD/Esquema | Convertir DTD en esquema** 1360 puede convertir documentos DTD en esquemas XML.
- *Generar archivo XML de muestra a partir de la DTD:* con el **comando DTD/Esquema | Generar archivo XML** 1365 de muestra puede generar un documento XML a partir de la DTD activa.

6.3 Esquemas XML

Los documentos de esquema XML se pueden editar en las vistas Texto, Cuadrícula y Esquema. La vista predeterminada en la que se abren los esquemas XML se puede especificar en la sección Tipos de archivo[1587] del cuadro de diálogo "Opciones". Mientras edita un esquema XML puede cambiar de una vista a otra, dependiendo de qué necesite en cada momento. Los documentos de esquema XML suelen tener la extensión de archivo `.xsd` o `.xs`.

Edición en la vista Texto

En la vista Texto los esquemas XML se editan como si fuesen documentos XML. Esta vista ofrece las mismas funciones de edición disponibles para los documentos XML[290] cuando se trabaja con esquemas XML. Al igual que ocurre con los documentos XML que tienen asociado un esquema, los ayudantes de entrada de la vista Texto[164] muestran los diferentes componentes que se pueden añadir en el punto de inserción del cursor.

Edición en la vista Cuadrícula

En la vista Texto los esquemas XML se editan como si fuesen documentos XML. Esta vista ofrece las mismas funciones de edición disponibles para los documentos XML[169] cuando se trabaja con esquemas XML. Al seleccionar un componente en la vista Cuadrícula, los ayudantes de entrada[299] muestran los diferentes componentes que se pueden añadir en el punto de inserción del cursor.

Edición en la vista Esquema

La vista Esquema es una interfaz gráfica para diseñar esquemas. Mientras crea o edita un esquema XML en la vista Esquema, XMLSpy genera en segundo plano un documento de texto equivalente. Para más información sobre cómo crear y editar esquemas XML en la vista Esquema, consulte el apartado Vistas de edición | Vista Esquema[171].

Sitio web de Altova: Editor XSD

Funciones de XMLSpy relacionadas con esquemas XML

XMLSpy también ofrece estas prácticas funciones:

- *Convertir esquemas XML en DTD* con el **comando DTD/Esquema | Convertir Esquema en esquema**[1363] puede convertir documentos DTD DTD XML.
- *Generar archivo XML de muestra a partir de la XML:* con el **comando DTD/Esquema | Generar archivo XML**[1365] de muestra puede generar un documento XML a partir de la DTD activa. También puede indicar valores de muestra para los elementos y atributos del archivo XML de muestra.
- En la vista Esquema puede crear firmas XML[411] en archivos de firma externos para sus archivos de esquema XML (`.xsd`). Para más información sobre cómo funcionan las firmas, consulte la sección Firmas XML[411].

© 2016-2022 Altova GmbH

6.4 Subconjuntos de esquema

Puede crear un archivo de esquema con algunos componentes de su esquema XML. Este esquema más pequeño se denomina *subconjunto de esquema*. La ventaja de usar subconjuntos de esquemas para formar el esquema mayor (mediante inclusiones) es que los archivos más pequeños se pueden gestionar más fácilmente que un solo esquema de gran tamaño.

A continuación describimos un posible caso de uso de la función de subconjuntos de esquemas en la vista Esquema:

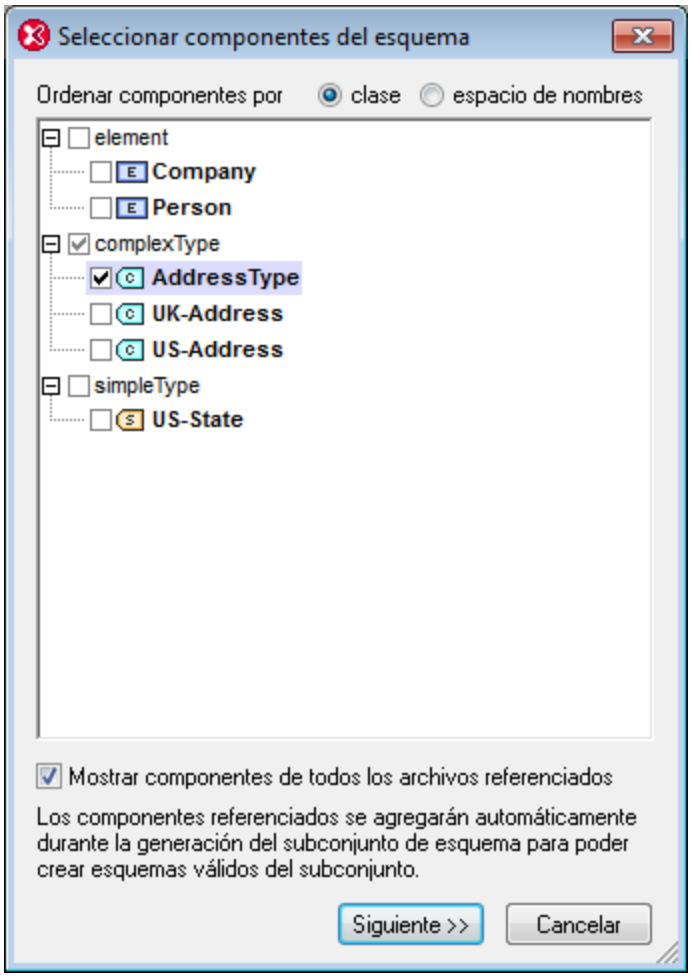
1. Cree un subconjunto de esquema que contenga como mínimo un componente del esquema activo. Más abajo explicamos cómo hacerlo. 445
2. Si es necesario, cree otros subconjuntos de esquema.
3. Incluya los subconjuntos de esquema recién creados para formar el esquema mayor. Para incluir los subconjuntos de esquema en el esquema principal, anexe o inserte un componente `Include` en la ventana de vista general del esquema 177 y seleccione el archivo de subconjunto de esquema que creó en los pasos anteriores.
4. En el esquema original elimine los componentes del esquema que ahora aparecen dos veces en los subconjuntos incluidos.

En la vista Esquema también puede realizar el proceso inverso, es decir, nivelar los subconjuntos de esquema incluidos de modo que: (i) los componentes de los subconjuntos de esquema se añadan directamente al esquema principal y (ii) los subconjuntos de esquema incluidos se eliminen del esquema principal. Más abajo explicamos cómo 447 se nivela un esquema.

Crear subconjuntos de esquema

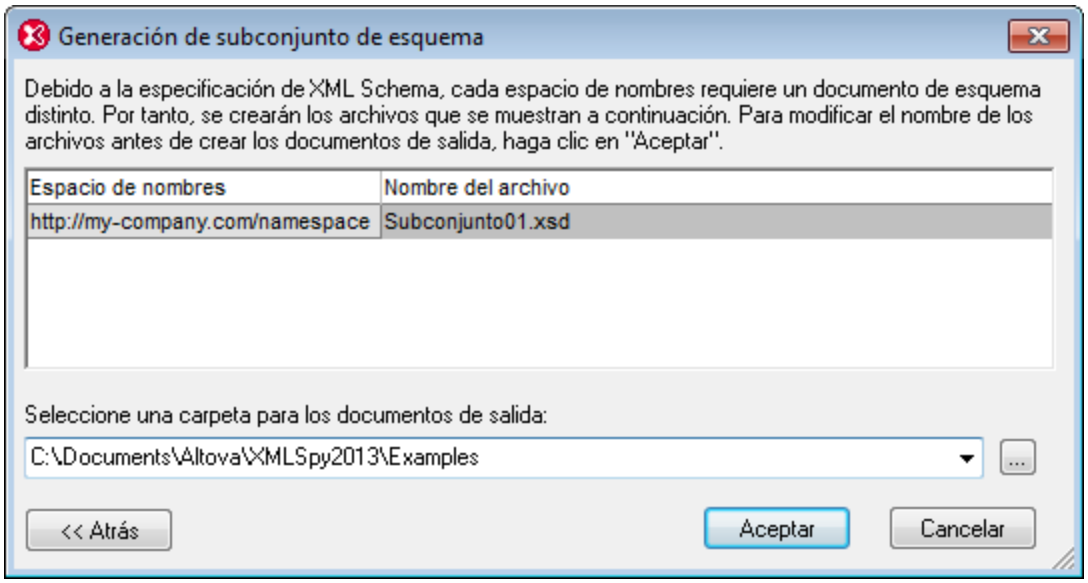
Para crear un subconjunto de esquema siga estos pasos:

1. Abra el esquema XML en la vista Esquema y seleccione el comando **Diseño de esquemas | Crear subconjunto de esquema**. Aparece el cuadro de diálogo "Seleccionar componentes del esquema" (*ver imagen siguiente*).
2. En el cuadro de diálogo marque los componentes que desea usar para crear un subconjunto de esquemas y después haga clic en **Siguiente**. (Observe que el cuadro de diálogo incluye una casilla para mostrar también los componentes de todos los archivos referenciados.)



3. Aparece el cuadro de diálogo "Generación de subconjunto de esquema" (*ver imagen siguiente*). En este cuadro de diálogo escriba el nombre que desea dar al paquete de subconjuntos de esquema. También debe indicar en qué carpeta se deben guardar los archivos de subconjunto de esquemas. Si uno de los componentes es un componente importado en el esquema original, el paquete de subconjuntos de esquema puede tener varios archivos. Por cada espacio de nombres del subconjunto de esquemas se crea un archivo de esquema distinto. Los nombres de los archivos que aparecen por defecto en el cuadro de diálogo son los nombres de los archivos originales. Si desea guardar los archivos en la carpeta donde están los archivos originales, es necesario utilizar nombres de archivo nuevos porque los archivos originales no se pueden sobrescribir.

© 2016-2022 Altova GmbH

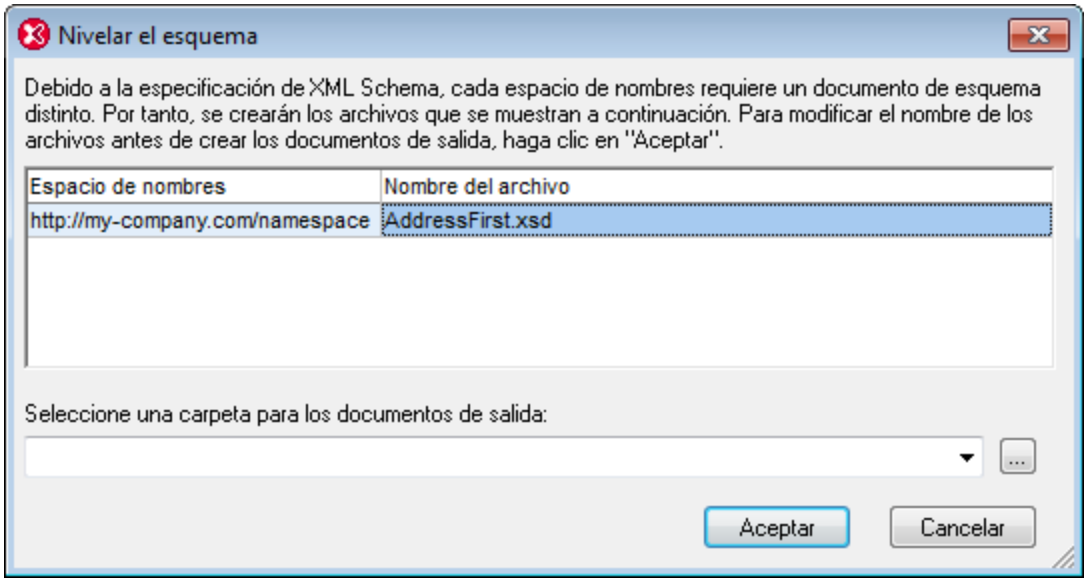


4. Al hacer clic en **Aceptar** se abre el archivo de subconjunto de esquema cuyo espacio de nombres coincida con el del archivo que está activo en la vista Esquema. Al mismo tiempo se crean los demás archivos del paquete y se guardan en la carpeta indicada (pero no se abren en la vista Esquema).

Nivelar un esquema

El proceso de nivelar un esquema activo en la vista Esquema consiste en: (i) añadir los componentes de todos los esquemas incluidos como componentes globales del esquema activo y (ii) eliminar los esquemas incluidos.

Para nivelar el esquema activo, seleccione el comando **Diseño de esquemas | Nivelar el esquema**. Aparece el cuadro de diálogo "Nivelar el esquema" (*imagen siguiente*), que incluye una lista con varios archivos, cuyos nombres coinciden con los espacios de nombres que se crearán en el esquema nivelado. Estos nombres predeterminados son los nombres de los archivos originales. Si desea guardar estos archivos en la misma carpeta que el archivo activo, debe cambiar el nombre de los archivos porque los archivos originales no se pueden sobrescribir. En el cuadro de diálogo puede indicar en qué carpeta se deben guardar el esquema nivelado y sus archivos asociados.



Al hacer clic en **Aceptar** el archivo de esquema nivelado se abre en la vista Esquema.

© 2016-2022 Altova GmbH

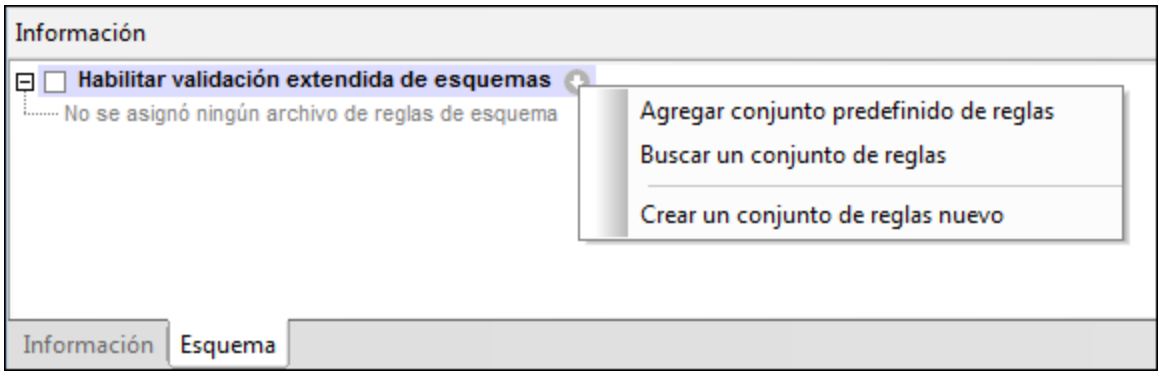
6.5 Reglas de esquema

Un conjunto de reglas de esquema es un conjunto de reglas que se puede usar para validar un esquema XML. Por ejemplo, una regla puede exigir que los nombres de los atributos, de acuerdo con lo definido en el esquema XML, empiecen con minúsculas o que un complexType determinado sólo puede extenderse desde un tipo determinado.

Un conjunto de reglas de esquema se guarda en un archivo de conjunto de reglas, que es un archivo XML (`.xml`). XMLSpy incluye un editor de reglas de esquema 451 con el que puede editar las reglas de esquema de forma gráfica. En XMLSpy puede asignar archivos de reglas de esquema 449 a un esquema XML. El esquema XML se valida en la vista Esquema con los conjuntos de reglas asignados (comando **XML | Validar (F8)**).

6.5.1 Administrar conjuntos de reglas

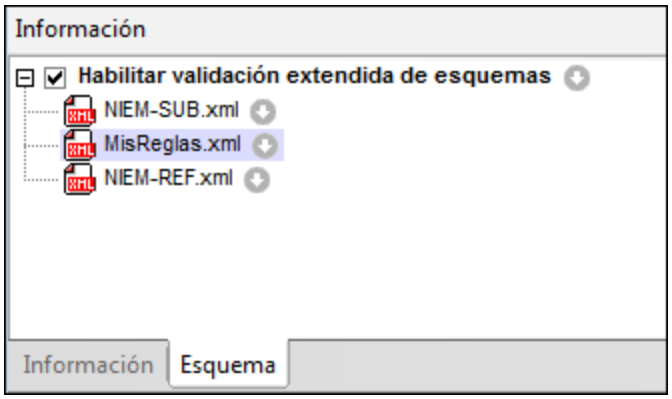
Puede asignar archivos de conjunto de reglas de esquema (archivos `.xml`) al esquema XML activo (archivo `.xsd`). Para ello utilice la pestaña Esquema de la ventana Información (*ver imagen siguiente*).



Agregar conjuntos de reglas para validación extendida de esquemas

Para agregar un archivo de conjunto de reglas de esquema, haga clic en el botón . Aparece un menú (*ver imagen anterior*) en el que puede seleccionar cómo desea añadir los conjuntos de reglas de esquema al esquema XML. Tiene tres opciones:

- *Agregar conjunto predefinido de reglas:* puede seleccionar un conjunto de reglas de una lista predeterminada de XMLSpy. Estos archivos de conjunto de reglas están guardados en la carpeta `Extended Schema Validation` de la carpeta de aplicación de XMLSpy. Todos los archivos de conjunto de reglas que se guarden en esta carpeta aparecerán en el cuadro de diálogo "Conjunto predefinido de reglas" para que pueda añadirlos.
- *Buscar un conjunto de reglas*: puede buscar un archivo ya existente de conjunto de reglas.
- *Crear un conjunto de reglas nuevo*: abre el editor de reglas de esquema, donde puede editar las reglas de esquema en un archivo de conjunto de reglas de esquema. Para más información sobre el editor de reglas de esquema, consulte el apartado Definir un conjunto de reglas 451 de esta sección. Después de guardar el archivo de conjunto de reglas de esquema, este se añade a la lista de conjuntos de reglas del esquema XML activo (*ver imagen siguiente*).



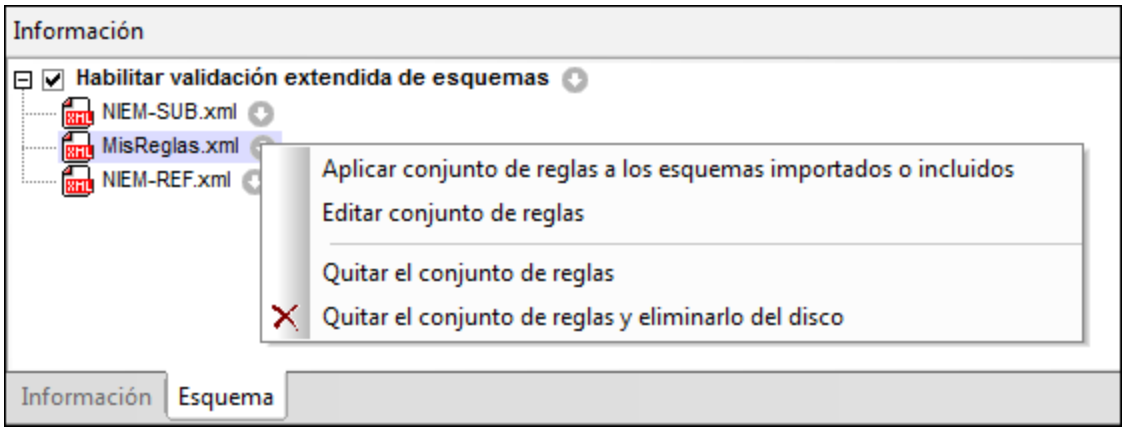
Puede añadir todos los conjuntos de reglas de esquema que desee (*ver imagen anterior*). Cuando se asignan varios conjuntos de reglas a un esquema XML, se usan todas las reglas de dichos conjuntos al validar el esquema XML en la vista Esquema (**XML | Validar**).

Habilitar y deshabilitar la validación extendida de esquemas

La validación extendida de esquemas se puede habilitar o deshabilitar con solo hacer clic en la casilla *Habilitar validación extendida de esquemas*.

Editar y quitar conjuntos de reglas

Puede gestionar cada uno de los conjuntos de reglas asignados a un esquema XML. Para ello, haga clic en el botón  del conjunto de reglas pertinente y elija una de las opciones del menú contextual (*ver imagen siguiente*).



Estas son las opciones del menú contextual:

- *Aplicar conjunto de reglas a los esquemas importados o incluidos:* si se aplica un conjunto de reglas, las reglas del conjunto se usarán para validar todos los esquemas que importa o incluye el esquema principal.
- *Editar conjunto de reglas:* abre el conjunto de reglas de esquema en el editor de reglas de esquema.
- *Quitar el conjunto de reglas:* quita el conjunto de reglas de la lista de conjuntos de reglas añadidos.

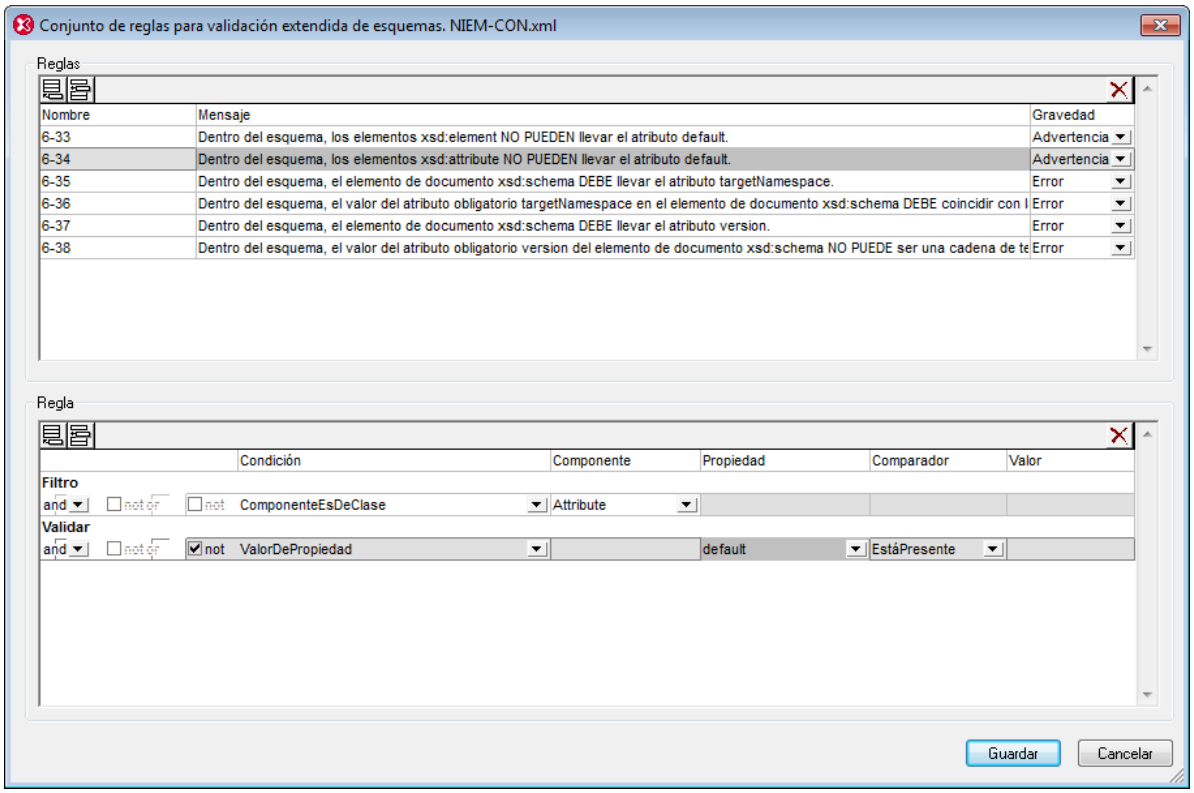
© 2016-2022 Altova GmbH

- *Quitar el conjunto de reglas y eliminarlo del disco:* este comando está disponible sólo para los conjuntos de reglas definidos por el usuario. Además de quitar el conjunto de reglas de la lista de conjuntos de reglas añadidos, este comando también elimina del disco el conjunto de reglas.

6.5.2 Definir un conjunto de reglas

Los conjuntos de reglas se pueden abrir y editar en el editor de reglas de esquema (*ver imagen siguiente*). En el editor puede crear, editar y eliminar reglas de esquemas en el archivo de conjunto de reglas de esquema. Para abrir un conjunto de reglas en el editor de reglas de esquema, siga estos pasos:

1. Seleccione el conjunto de reglas de la lista de conjuntos de reglas de la ventana Información.
2. Haga clic en el botón  del conjunto de reglas.
3. En el menú contextual que aparece seleccione la opción *Editar conjunto de reglas*.



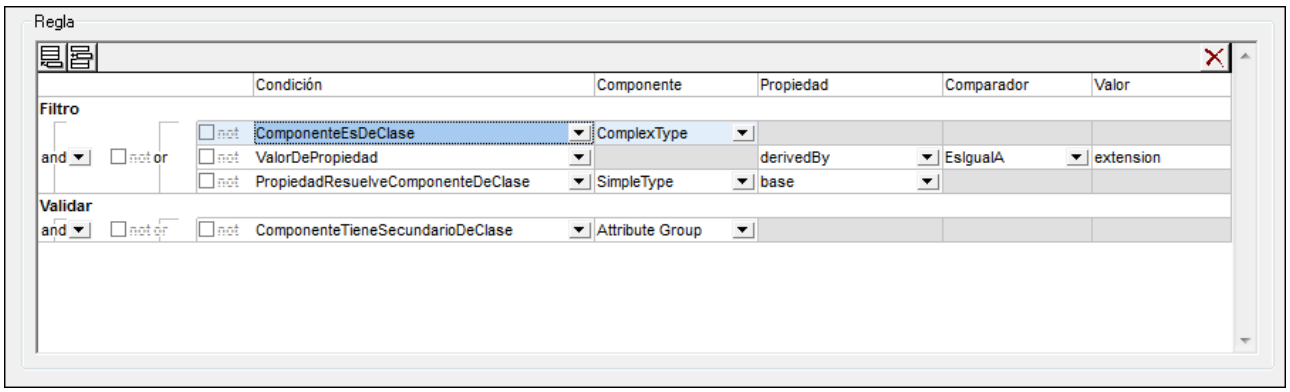
A continuación aparece el cuadro de diálogo del editor de reglas de esquema, que se divide en dos paneles:

- El panel "Reglas" (situado en la mitad superior del editor), donde puede añadir y eliminar reglas. Para anexar o insertar una regla nueva, haga clic en el botón correspondiente (**Anexar** o **Insertar**), situados en la esquina superior izquierda del panel. Para eliminar una regla, seleccione la regla que desea eliminar y haga clic en el botón **Eliminar**, situado en la esquina superior derecha del panel. Cada regla de este panel tiene un nombre, un mensaje descriptivo y un nivel de gravedad (si la regla no se cumple, el validador puede devolver un mensaje de error o de advertencia).
- El panel "Regla" (situado en la mitad inferior del editor), donde aparecen los detalles de la regla seleccionada en el panel Reglas. En este panel se pueden editar los detalles de las reglas. Para más información, consulte el apartado Definir una regla 452 que aparece a continuación.

Cuando termine de editar el archivo de conjunto de reglas, haga clic en **Guardar** para guardar las reglas en el archivo de conjunto de reglas.

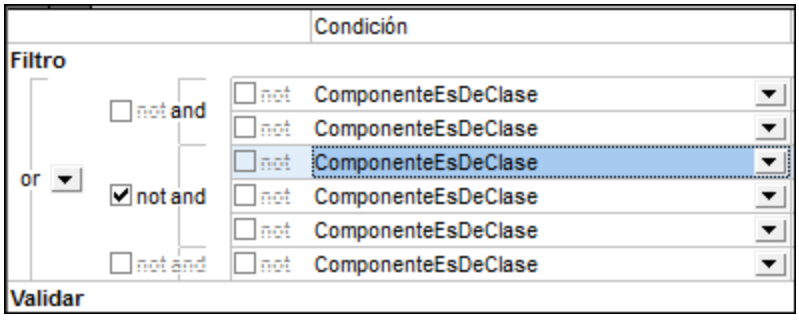
Definir una regla

Seleccione la regla del panel superior "Reglas" que desea definir o editar. En el panel inferior "Regla" puede editar la definición de la regla. La imagen siguiente muestra una regla que se podría definir de la siguiente manera: *Si un complexType es una extensión de un simpleType, entonces debe tener un elemento secundario de clase AttributeGroup.*



Condiciones de tipo "Validar" y condiciones de tipo "Filtro"

- Cada regla tiene un conjunto de condiciones de tipo "Validar" y un conjunto de condiciones de tipo "Filtro" (*ver la primera columna de la imagen anterior*).
- El conjunto de condiciones "Filtro" debe dar un valor de verdadero para que se puede evaluar la condición "Validar".
- Cada conjunto de condiciones (de tipo "Validar" o de tipo "Filtro") está formado por grupos de condiciones. Cada grupo contiene como mínimo una condición. En la imagen anterior, el conjunto de condiciones de tipo "Validar" contiene un grupo de condiciones formado por una sola condición. Por su parte, el conjunto de condiciones de tipo "Filtro" contiene tres grupos de condiciones, cada uno con una sola condición. En la imagen siguiente, el conjunto de condiciones de tipo "Filtro" contiene tres grupos de condiciones: el primero tiene dos condiciones, el segundo tiene tres y el tercero solo una.

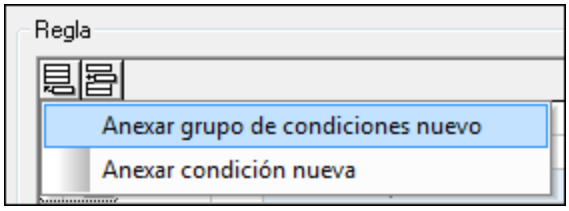


- Para negar una condición, marque la casilla *Not* de la condición (situada a la izquierda de la condición).
- Dentro de cada grupo de condiciones, los conectores lógicos `and` y `or` indican respectivamente si todas las condiciones del grupo o solo una deben dar un valor de verdadero para que el grupo entero

 © 2016-2022 Altova GmbH

dé un valor de verdadero. En la interfaz gráfica, estos operadores lógicos son los situados más a la derecha (en la tercera columna empezando por la izquierda).

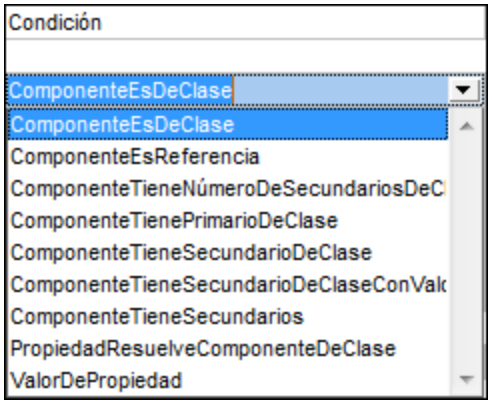
- Para negar un grupo de condiciones, marque la casilla *Not* del grupo de condición (situada a la izquierda del operador lógico del grupo de condiciones).
- Los conectores lógicos `and` y `or` situados más a la izquierda (junto al borde del cuadro de diálogo) indican respectivamente si todos los grupos de condiciones del conjunto o solo un grupo deben dar un valor de verdadero para que el conjunto entero (el conjunto "Filtro" por ejemplo) dé un valor de verdadero.
- Para cambiar los conectores lógicos seleccione la opción pertinente en el cuadro combinado situado más a la izquierda (el conector del grupo de condiciones). El valor de los conectores lógicos situados más a la derecha (los conectores de las condiciones del grupo de condiciones) recibirán el valor opuesto al del conector lógico situado más a la izquierda (el conector de todo el grupo de condiciones).
- Para anexar o insertar un grupo de condiciones o una condición en relación a otro grupo o a otra condición, seleccione el grupo o la condición pertinente y haga clic en el botón **Anexar** o **Insertar** (en la esquina superior izquierda del panel). Aparece un menú contextual con dos opciones (*ver imagen siguiente*).



Clases de condiciones

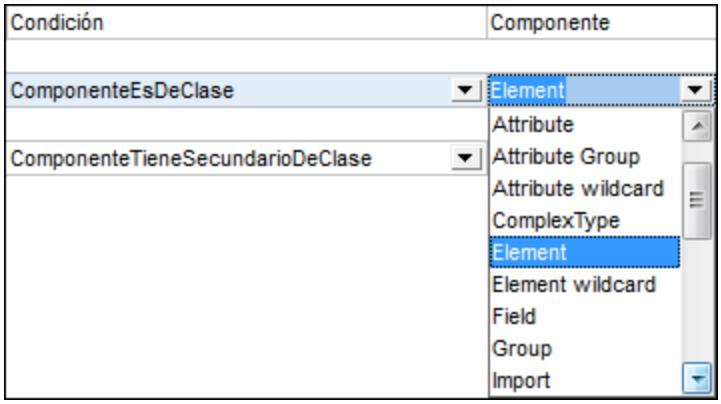
Hay tres tipos de condiciones (*véase también la imagen que aparece más abajo*):

- Condición basada en la clase de componente (las condiciones de la lista desplegable que empiezan con la palabra "Componente". *Ver imagen siguiente*)
- Condición basada en la clase de propiedad (valor de la propiedad)
- Condición basada en la clase de componente y de propiedad (las condiciones de la lista cuyo nombre incluye las palabras "Propiedad" y "Componente")



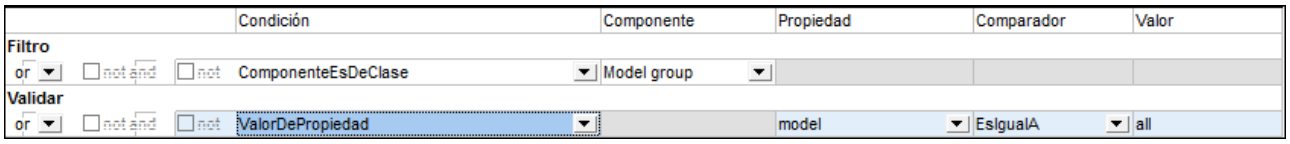
Para seleccionar el tipo de condición, haga clic en la punta de flecha de la columna **Condición** y seleccione una condición de la lista desplegable (*ver imagen anterior*). A continuación describimos los tres tipos de condiciones.

Condiciones basadas en la clase de componente
En las condiciones basadas en la clase de componente (las condiciones de la lista que empiezan con la palabra "Componente"), es obligatorio indicar la clase de componente en la columna adyacente *Componente* (*ver imagen siguiente*). El componente se selecciona en la lista desplegable de la columna Componente. Puesto que en las condiciones de este tipo no es necesario definir más campos (Propiedad, Comparador, Valor) el resto de los campos de la condición aparecen atenuados.



En la imagen anterior la condición de tipo "Filtro" especifica que la regla sólo afecta a componentes de clase *Element*. Si además la condición de tipo "Validar" especifica que el componente debe tener un secundario de clase *Notation*, entonces la regla completa se podría enunciar de la manera siguiente: "Los componentes *Element* deben tener un secundario de clase *Notation*". Si en la condición de tipo "Validar" se marcase la casilla `NOT`, entonces la regla se podría enunciar así: "Los componentes *Element* no pueden tener un secundario de clase *Notation*".

Condiciones basadas en la clase de propiedad
La condición basada en la clase de propiedad es la condición *ValorDePropiedad* (*ver imagen siguiente*). Este tipo de condición especifica la naturaleza de una propiedad. Por tanto, es obligatorio rellenar las columnas *Propiedad* y *Comparador* y, de forma opcional, la columna *Valor*. No es necesario rellenar la columna *Componente*, que aparece atenuada. La lista desplegable de la columna *Propiedad* no sólo incluye atributos XML (como `default` y `maxOccurs`) sino también propiedades lógicas de los componentes (como `derivedBy`).



La imagen anterior muestra una regla en la que la propiedad `model` tiene un valor igual a `All` y está negada (porque está marcada la casilla *Not*). En combinación con la condición de filtro del componente de clase *Model group*, esta regla se podría resumir en este enunciado: "El esquema XML no puede contener elementos `xsd:all`".

Nota: Es importante tener en cuenta estos aspectos:

- Cuando use el comparador `EsQNameIgualA`, el valor para este comparador se debe escribir de esta manera: `{URI}nombreLocal`. Un ejemplo sería: `{http://www.w3.org/2001/XMLSchema}NOTATION`
- La propiedad `default` puede estar presente y estar vacía (`<element name default=""/>`) o puede faltar (`<element name/>`).

 © 2016-2022 Altova GmbH

Condiciones basadas en la clase de componente y de propiedad
Las condiciones que combinan la clase de componente y de propiedad son:

- `ComponenteTieneSecundarioDeClaseConValorDePropiedad`: especifica la clase de componente de un elemento secundario y el valor de la propiedad.
- `PropiedadResuelveComponenteDeClase`: se especifica una propiedad que se resuelve como una clase de componente. Las columnas *Comparador* y *Valor* se dejan vacías.

Negar una condición
Para negar una condición basta con marcar la casilla *Not* situada justo a la izquierda de la condición (es decir, la casilla *Not* que está más a la derecha). Para negar un grupo de condiciones es necesario marcar la casilla *Not* situada a la izquierda del conector lógico de las condiciones del grupo de condiciones.

6.6 Catálogos en XMLSpy

XMLSpy es compatible con un subconjunto del mecanismo de catalogación XML OASIS. El mecanismo de catalogación permite a XMLSpy recuperar de carpetas locales del usuario los esquemas (y hojas de estilos y otros archivos) usados con frecuencia. Esto incrementa la velocidad global de procesamiento, permite al usuario trabajar sin conexión (es decir, sin estar conectado a una red) y mejora la portabilidad de los documentos (porque los identificadores URI se tienen que cambiar sólo en los archivos de catálogo).

A continuación describimos cómo funciona el mecanismo de catalogación en XMLSpy.

- Cómo funcionan los catálogos 456
- Estructura de los catálogos en XMLSpy 457
- Personalizar catálogos 459
- Memoria insuficiente 460

Para más información consulte la especificación XML Catalogs.

6.6.1 Funcionamiento de los catálogos

Los catálogos se pueden usar para redirigir los esquemas DTD y XML. El concepto tras los mecanismos es el mismo en los dos casos, pero los detalles son distintos; los explicamos a continuación.

DTDs

Los catálogos se suelen usar para redirigir una llamada a un DTD o un URI local. Para ello se asignan, en el archivo de catálogo, identificadores públicos o del sistema al URI local necesario. De este modo, cuando se lee la declaración **DOCTYPE** en un archivo XML, el identificador de sistema o público localiza el recurso local necesario con ayuda de la asignación del archivo de catálogo.

Para los esquemas más utilizados el identificador **PUBLIC** suele estar predefinido y, por tanto, sólo hace falta que el URI del archivo de catálogo asigne el identificador **PUBLIC** a la copia local correcta. Cuando se analiza el documento XML, se lee el identificador **PUBLIC** del documento. Si se encuentra este identificador en un archivo de catálogo, se buscará la URL correspondiente del archivo de catálogo y se leerá el esquema desde esta ubicación. Por ejemplo, imaginemos que abrimos este archivo SVG en XMLSpy:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="20" height="20" xml:space="preserve">
  <g style="fill:red; stroke:#000000">
     <rect x="0" y="0" width="15" height="15"/>
     <rect x="5" y="5" width="15" height="15"/>
  </g>
</svg>
```

Se busca en el catálogo el identificador **PUBLIC** de este archivo SVG. Imaginemos que el archivo de catálogo contiene esta entrada:

© 2016-2022 Altova GmbH

```
<catalog>
  ...
    <public publicId="-//W3C//DTD SVG 1.1//EN" uri="schemas/svg/svg11.dtd"/>
  ...
</catalog>
```

En este caso hay una coincidencia del identificador **PUBLIC**. La consulta del SVG DTD se redirige al URL **schemas/svg/svg11.dtd** (que es relativo al archivo de catálogo). Este es un archivo local y se usa como DTD para el archivo SCG. Si en el catálogo no hay una asignación para el identificador **Public**, entonces se usa la URL del documento XML (en el archivo SVG del ejemplo anterior la URL de Internet es: **http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd**).

Esquemas XML

En XMLSpy también puede usar catálogos para redireccionar a un **esquema XML**. En el archivo de instancia XML, la referencia al esquema sucederá en el atributo **xsi:schemaLocation** del elemento de documento de nivel superior del documento XML. Por ejemplo:

```
xsi:schemaLocation="http://www.xmlspy.com/schemas/orgchart OrgChart.xsd"
```

El valor del atributo **xsi:schemaLocation** tiene dos partes: un espacio de nombres (verde) y un URI (resaltado). La parte del espacio de nombres se usa en el catálogo para la asignación con el recurso alternativo. Por ejemplo, esta entrada de catálogo redirige la referencia del esquema anterior a un esquema en una ubicación alternativa.

```
<uri name="http://www.xmlspy.com/schemas/orgchart" uri="C:\MySchemas\OrgChart.xsd"/>
```

Por lo general, la parte URI del valor del atributo **xsi:schemaLocation** es una ruta a la ubicación real del esquema. Sin embargo, si se hace referencia al esquema a través de un catálogo, no es necesario que la parte URI apunte a un esquema XML real, aunque el esquema debe existir para que el atributo **xsi:schemaLocation** siga siendo válido desde el punto de vista léxico. Por ejemplo, el valor `foo` sería suficiente para que la parte URI del valor del atributo sea válida.

6.6.2 Estructura de los catálogos en XMLSpy

Al iniciarse, XMLSpy carga un archivo llamado **RootCatalog.xml** (*cuya estructura aparece a continuación*), que contiene una lista de los archivos de catálogo que se buscarán. El usuario puede modificar esta lista y añadir tantos archivos de catálogo como desee, escribiendo cada archivo en un elemento **nextCatalog**. La aplicación busca cada uno de estos archivos de catálogo y sus URI se resuelven de acuerdo con sus asignaciones.

Extracto de RootCatalog.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog"
   xmlns:spy="http://www.altova.com/catalog_ext"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:entity:xmlns:xml:catalog Catalog.xsd">
  <nextCatalog catalog="%PersonalFolder%/Altova/%AppAndVersionName%/CustomCatalog.xml"/>
  <!-- Include all catalogs under common schemas folder on the first directory level -->
  <nextCatalog spy:recurseFrom="%CommonSchemasFolder%" catalog="catalog.xml"
```

```
spy:depth="1"/>
  <nextCatalog spy:recurseFrom="%ApplicationWritableDataFolder%/pkgs/.cache"
catalog="remapping.xml" spy:depth="0"/>
  <nextCatalog catalog="CoreCatalog.xml"/>
</catalog>
```

El extracto anterior hace referencia a un catálogo personalizado (**CustomCatalog.xml**) y a un conjunto de catálogos que localizan los esquemas usados a menudo (como W3C XML Schemas y el esquema SVG).

- **CustomCatalog.xml** se encuentra en su carpeta personal (a la que puede acceder con la variable **%PersonalFolder%**). Es un archivo base en el que el usuario puede crear asignaciones propias. Puede añadir asignaciones a **CustomCatalog.xml** para cualquier esquema que necesite que no aparezca en los archivos de catálogo de la carpeta Common Schemas. Debe usar los elementos compatibles del mecanismo de catálogo OASIS (*véase más abajo*).
- La carpeta Common Schemas Folder (a la que se accede con la variable **%CommonSchemasFolder%**) contiene un conjunto de esquemas de uso habitual. Dentro de cada una de estas carpetas hay un archivo **catalog.xml** que asigna identificadores públicos y/o del sistema a URIs que apuntan a copias locales de los esquemas correspondientes.
- Los esquemas relacionados con XBRL y varias taxonomías XBRL son grandes y se instalan localmente a petición con ayuda del Gestor de taxonomías de Altova. Los esquemas y las taxonomías individuales se asignan en el catálogo **remapping.xml**, que se encuentra en la subcarpeta **pkgs/.cache** de la carpeta Program Data (a la que puede acceder con la variable **%ApplicationWritableDatFolder%**).
- **CoreCatalog.xml** está en la carpeta de la aplicación XMLSpy y se usa para localizar esquemas y hojas de estilos que usan los procesos específicos de XMLSpy, como los archivos SPS de StyleVision, que se usan para generar documentos XML para la Vista Authentic de Altova.

Variables de ubicación

Las variables que se usan en **RootCatalog.xml** (*véase el extracto más arriba*) tienen estos valores:

%PersonalFolder%	La carpeta personal del usuario, por ejemplo C:\Users\<name>\Documents.
%CommonSchemasFolder%	C:\ProgramData\Altova\Common2023\Schemas
%ApplicationWritableDataFolder%	C:\ProgramData\Altova

Ubicación de los archivos de catálogo y los esquemas

Estas son las ubicaciones de los distintos archivos de los catálogos.

- **RootCatalog.xml** y **CoreCatalog.xml** se instalan en la carpeta de aplicación de XMLSpy.
- **CustomCatalog.xml** está ubicado en su carpeta **MisDocumentos\Altova\XMLSpy**.
- Cada archivo **catalog.xml** está en una carpeta de esquema y estas carpetas están dentro de la carpeta común de esquemas.

© 2016-2022 Altova GmbH

6.6.3 Personalizar catálogos

Cuando cree entradas en el archivo **CustomCatalog.xml** (o en cualquier otro archivo de catálogo que sea leído por XMLSpy), utilice únicamente los elementos que aparecen a continuación de la especificación de catálogos OASIS. En la lista que aparece más adelante explicamos los valores de los atributos de cada elemento. Si desea consultar una descripción más detallada, visite la página de la especificación XML Catalogs. Tenga en cuenta que cada uno de los elementos del subconjunto pueden llevar el atributo **xml:base**, que se usa para especificar el URI base del elemento.

- `<public publicId="PublicID of Resource" uri="URL of local file"/>`
- `<system systemId="SystemID of Resource" uri="URL of local file"/>`
- `<uri name="filename" uri="URL of file identified by filename"/>`
- `<rewriteURI uriStartString="StartString of URI to rewrite" rewritePrefix="String to replace StartString"/>`
- `<rewriteSystem systemIdStartString="StartString of SystemID" rewritePrefix="Replacement string to locate resource locally"/>`

Tenga en cuenta que:

- Cuando no exista un identificador público, como es el caso de casi todas las hojas de estilos, el identificador de sistema se puede asignar directamente a una URL con el elemento **system**.
- Un URI se puede asignar a otro URI con el elemento `uri`.
- Los elementos `rewriteURI` y `rewriteSystem` sirven para volver a escribir la parte inicial de un URI o identificador de sistema respectivamente. Gracias a ello se puede sustituir el principio de la ruta de acceso de un archivo y, por consiguiente, se puede apuntar a otro directorio. Para más información sobre estos elementos, consulte la especificación XML Catalogs.

A partir de su versión de 2014 XMLSpy cumple escrupulosamente con la especificación de catálogos XML XML Catalogs specification (OASIS Standard V1.1, 7 October 2005) Esta especificación separa estrictamente las consultas por identificador externo (las realizadas con un ID público o de sistema) de las búsquedas por URI (los URI que no son ID públicos ni de sistema). Por tanto, los URIs de espacios de nombres deben considerarse simplemente como URIs (no como IDs públicos o del sistema) y deben usarse como búsquedas por URI y no como consultas por identificador externo. En las versiones de XMLSpy previas a la de 2014 los URIs de espacios de nombres de esquemas se traducían con asignaciones **<public>**. Sin embargo, a partir de la versión 2014 es necesario utilizar asignaciones **<uri>**.

Antes de la versión v2014: `<public publicID="http://www.MyMapping.com/ref" uri="file:///C:/MyDocs/Catalog/test.xsd"/>`
A partir de la versión 2014: `<uri name="http://www.MyMapping.com/ref" uri="file:///C:/MyDocs/Catalog/test.xsd"/>`

Cómo encuentra XMLSpy un esquema de referencia

Para hacer referencia a un esquema desde un documento XML se usa el atributo **xsi:scemaLocation** (*más abajo*). El valor del atributo **xsi:schemaLocation** tiene dos partes: un espacio de nombres (verde) y un URI (resaltado).

```
xsi:schemaLocation="http://www.xmlspy.com/schemas/orgchart OrgChart.xsd"
```

A continuación damos los pasos que sigue XMLSpy de forma secuencial para encontrar un esquema de referencia. El esquema se carga en el primer paso que se ejecuta correctamente.

1. Consulte en el catálogo la parte del URI del valor **xsi:schemaLocation**. Si se encuentra una asignación, y esto incluye las asignaciones **rewriteURI**, use el URI resultante para cargar el esquema.
2. Consulte en el catálogo la parte del espacio de nombres del valor **xsi:schemaLocation**. Si se encuentra una asignación, y esto incluye las asignaciones **rewriteURI**, use el URI resultante para cargar el esquema.
3. Use la parte URI del valor **xsi:schemaLocation** para cargar el esquema.

Extensiones de archivo y edición inteligente basada en un esquema

Mediante los archivos de catálogo también puede especificar a qué tipo de documentos (según su extensión de archivo) se les puede aplicar las funciones de edición inteligente de XMLSpy de acuerdo con las reglas del esquema que usted indique. Por ejemplo, si crea una extensión de archivo personalizada `.myhtml` para archivos (HTML) que deben ser válidos conforme al HTML DTD, entonces puede habilitar la edición inteligente para los archivos que tengan esas extensiones; para ello añada el elemento de texto siguiente a **CustomCatalog.xml** como elemento secundario del elemento **<catalog>**.

```
<catalog>
  ...
    <spy:fileExtHelper ext="myhtml" uri="schemas/xhtml/xhtml1-transitional.dtd"/>
  ...
</catalog>
```

Así se habilita la función de edición inteligente (finalización automática, ayudantes de entrada, etc.) para archivos **.myhtml** en XMLSpy en base a la DTD transicional XHTML 1.0. Consulte el archivo **catalog.xml** de la carpeta **%AltovaCommonSchemasFolder%\Schemas\xhtml**, que contiene entradas similares a este ejemplo.

Especificaciones de XML Schema

La información de la especificación XML Schema está integrada en XMLSpy y esta información interna se usa para validar los documentos de esquema XML (`.xsd`). Por tanto, en los documentos de esquema XML no se deberían hacer referencia a ningún esquema que defina la especificación XML Schema.

El archivo **catalog.xml** de la carpeta `%AltovaCommonSchemasFolder%\Schemas\schema` incluye referencias a las DTD que implementan especificaciones antiguas de XML Schema. Rogamos no valide sus documentos de esquema XML con estos esquemas. Los archivos referenciados se incluyen con el único objetivo de aportar información a XMLSpy para sus ayudantes de entrada, en caso de que el usuario quiera crear documentos basados en estas recomendaciones.

6.6.4 Variables de entorno

En el elemento `nextCatalog` puede utilizar algunas variables de entorno Shell para indicar la ruta de acceso a las ubicaciones del sistema (*ver el fragmento anterior del archivo RootCatalog.xml*). Estas son las variables de entorno Shell compatibles:

`%PersonalFolder%`	Ruta de acceso completa de la carpeta personal del usuario actual, por ejemplo `C:\Usuarios\<nombre>\Documentos`

© 2016-2022 Altova GmbH

%CommonSchemasFolder%	C:\ProgramData\Altova\Common2023\Schemas
%ApplicationWritableDataFolder%	C:\ProgramData\Altova
%AltovaCommonFolder%	C:\Archivos de programa\Altova\Common2023
%DesktopFolder%	Ruta de acceso completa de la carpeta Escritorio del usuario actual.
%ProgramMenuFolder%	Ruta de acceso completa de la carpeta del menú Programas del usuario actual.
%StartMenuFolder%	Ruta de acceso completa de la carpeta del menú Inicio del usuario actual.
%StartUpFolder%	Ruta de acceso completa de la carpeta Inicio del usuario actual.
%TemplateFolder%	Ruta de acceso completa de la carpeta de plantillas del usuario actual.
%AdminToolsFolder%	Ruta de acceso completa del directorio del sistema de archivos que almacena las herramientas administrativas del usuario actual.
%AppDataFolder%	Ruta de acceso completa de la carpeta Datos de programa del usuario actual.
%CommonAppDataFolder%	Ruta de acceso completa al directorio del sistema de archivos que sirve como repositorio de datos para aplicaciones locales (no roaming).
%FavoritesFolder%	Ruta de acceso completa de la carpeta Favoritos del usuario actual.
%PersonalFolder%	Ruta de acceso completa de la carpeta personal del usuario actual.
%SendToFolder%	Ruta de acceso completa de la carpeta SendTo del usuario actual.
%FontsFolder%	Ruta de acceso completa de la carpeta Fuentes del sistema.
%ProgramFilesFolder%	Ruta de acceso completa de la carpeta Archivos de programa del usuario actual.
%CommonFilesFolder%	Ruta de acceso completa de la carpeta Common files del usuario actual.
%WindowsFolder%	Ruta de acceso completa de la carpeta Windows del usuario actual.
%SystemFolder%	Ruta de acceso completa de la carpeta System del usuario actual.
%LocalAppDataFolder%	Ruta de acceso completa al directorio del sistema de archivos que sirve como repositorio de datos para aplicaciones locales (no roaming).
%MyPicturesFolder	Ruta de acceso completa a la carpeta Mis imágenes.

%

© 2016-2022 Altova GmbH

6.7 Trabajar con SchemaAgent

XMLSpy puede configurarse para trabajar con la aplicación SchemaAgent de Altova.

Tecnología SchemaAgent

La tecnología SchemaAgent permite al usuario generar y editar relaciones entre varios esquemas. La tecnología está compuesta por:

- La aplicación servidor SchemaAgent Server, que almacena y sirve la información sobre las relaciones existentes entre los esquemas de las rutas de búsqueda (carpetas de la red) indicadas por el usuario.
- La aplicación cliente Altova SchemaAgent, que utiliza la información sobre el esquema disponible en el servidor de SchemaAgent al que está conectada para (i) generar relaciones entre estos esquemas y (ii) administrar estos esquemas (cambiarlos de nombre, moverlos de sitio, eliminar esquemas, etc.).

Hay dos tipos de servidor SchemaAgent:

- Altova SchemaAgent Server, que se puede instalar en una red, desde la que se puede acceder a él.
- Altova SchemaAgent, que es el producto cliente de SchemaAgent. Incluye una versión más ligera del servidor, llamada LocalServer, que sólo se puede usar en el equipo en el que está instalado SchemaAgent.

XMLSpy utiliza la tecnología SchemaAgent para editar esquemas en la vista Esquema usando la información sobre otros esquemas que obtiene desde un servidor de SchemaAgent. En esta configuración, XMLSpy está conectado a un servidor de SchemaAgent y, en interacción con el SchemaAgent Client, envía solicitudes a SchemaAgent Server. Cuando XMLSpy se configura para trabajar con SchemaAgent, los ayudantes de entrada de la vista Esquema no sólo incluyen componentes del esquema activo en la vista Esquema, sino que además incluyen componentes de otros esquemas ubicados en las rutas de búsqueda del servidor de SchemaAgent al que está conectado. De este modo, el usuario tiene acceso directo a estos componentes. En la vista Esquema puede visualizar el modelo de contenido de cualquier componente de otro esquema y volver a utilizar estos componentes tal y como están o con algunos cambios. También podrá crear relaciones entre los esquemas en XMLSpy para separar y combinar sus componentes y para gestionar esquemas complejos.

Instalación de SchemaAgent y SchemaAgent Server

Para más información sobre la instalación de SchemaAgent y SchemaAgent Server y cómo configurar las rutas de búsqueda en los servidores, consulte la Guía y referencia del usuario de SchemaAgent.

Configurar XMLSpy como cliente de SchemaAgent

Siga estos pasos para que XMLSpy funcione como cliente de SchemaAgent:

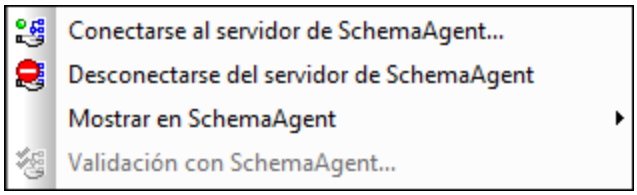
- Descargue SchemaAgent del sitio web de Altova. Ahora también puede usar el LocalServer de SchemaAgent para servir esquemas. Para más información sobre cómo configurar rutas de búsqueda en LocalServer, consulte el Guía y referencia del usuario de SchemaAgent.
  Nota importante: para usar SchemaAgent es necesario tener una licencia válida, que debe adquirirse una vez expire el plazo de prueba gratuito Recuerde que el conjunto de aplicaciones Altova MissionKit incluye la aplicación SchemaAgent con una clave de licencia. (La aplicación SchemaAgent Server no está incluida en el conjunto de aplicaciones Altova MissionKit.)
- También puede descargar e instalar la aplicación en red SchemaAgent Server del sitio web de Altova.

- Defina las rutas de búsqueda para el servidor de SchemaAgent (es decir, configure SchemaAgent Server). Para más información sobre cómo hacer esto, consulte el Guía y referencia del usuario de SchemaAgent. (Una ruta de búsqueda es una ruta de acceso a una carpeta que contiene los esquemas XML y que se usará para las relaciones entre esquemas.)
- Inicie una conexión a un servidor de SchemaAgent desde XMLSpy.

Nota importante: todos los productos de SchemaAgent y todos los productos de Altova relacionados con SchemaAgent (incluido XMLSpy) a partir de la versión 2005 Release 3 **no son compatibles** con las versiones anteriores de SchemaAgent ni de los productos de Altova relacionados con SchemaAgent.

Comandos de SchemaAgent en XMLSpy

En XMLSpy las funciones de SchemaAgent sólo están disponibles en la vista Esquema. A estas funciones se accede con los comandos del menú Diseño de esquemas (*ver imagen siguiente*) y desde los ayudantes de entrada de la vista Esquema.



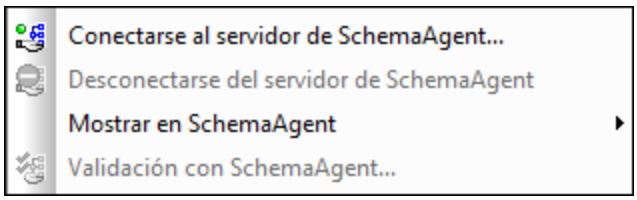
Los comandos de menú ofrecen funciones administrativas generales. Los ayudantes de entrada (y el resto de mecanismos estándar de la interfaz gráfica, como las operaciones de arrastrar y colocar) sirven para editar los esquemas.

Esta sección describe cómo usar las funciones de SchemaAgent disponibles en la vista Esquema.

6.7.1 Conectarse a SchemaAgent Server

Nota importante: es necesario tener instalado el SchemaAgent Clientpara poder iniciar una conexión.

Antes de conectarse a SchemaAgent Server, el único comando de SchemaAgent que está habilitado en el menú **Diseño de esquemas** es el comando **Conectarse al servidor de SchemaAgent** (*ver imagen siguiente*). El resto de comandos se habilitan en cuanto se completa la conexión a SchemaAgent Server. El resto de comandos se habilitan en cuanto se completa la conexión a SchemaAgent Server.

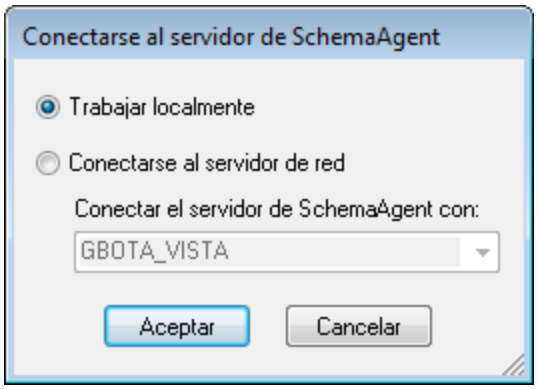


Instrucciones para conectarse a un servidor de SchemaAgent

Siga estos pasos para conectarse a un servidor de SchemaAgent:

© 2016-2022 Altova GmbH

Haga clic en el icono "Conectarse a SchemaAgent"  de la barra de herramientas (**Diseño de esquemas | Conectarse al servidor de SchemaAgent**). Aparece el cuadro de diálogo "Conectarse al servidor de SchemaAgent" (*ver imagen siguiente*):



En este diálogo puede elegir si usa el servidor local (el servidor de SchemaAgent que viene con Altova SchemaAgent) o un servidor de red (el producto Altova SchemaAgent Server, que se puede descargar gratis). Si selecciona la opción Trabajar localmente, el servidor local de SchemaAgent se iniciará nada más hacer clic en **Aceptar** y se establecerá una conexión con este servidor. Si elige la opción **Conectarse al servidor de red**, SchemaAgent Server tiene que estar funcionando para poder establecer la conexión.

Nota sobre servidores que funcionan con Windows XP SP2
Si el nombre del servidor SchemaAgent Server aparece en el cuadro de diálogo "Conectarse al servidor de SchemaAgent" pero no se puede conectar al servidor, es posible que su servidor no esté participando en el proceso de resolución de nombres de su red. La resolución de nombres está bloqueada por las opciones de configuración del Firewall de Windows XP SP2.

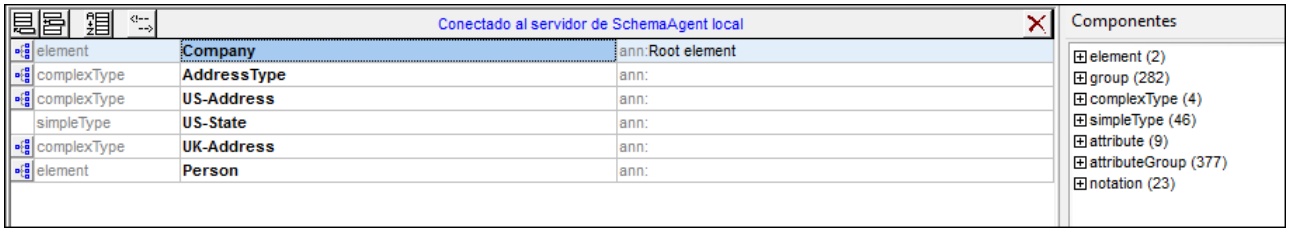
Para conectarse a un servidor de este tipo, tiene dos opciones:

- puede cambiar la configuración del servidor para permitir el proceso de resolución de nombres o
- puede escribir la dirección IP del servidor en el campo de edición del cuadro de diálogo "Conectarse al servidor de SchemaAgent".

Sólo tendrá que hacer esto una vez, porque el SchemaAgent Client almacena la cadena de conexión de la última conexión establecida.

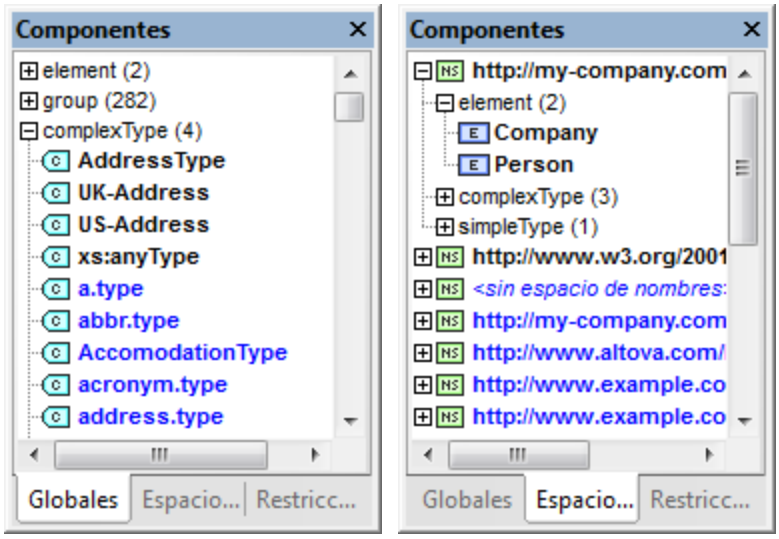
Cambios en la vista Esquema después de conectarse al servidor de SchemaAgent

Tras establecer la conexión con el servidor de SchemaAgent, la vista Esquema tendrá este aspecto:



Nota importante:

- en la parte superior de la vista de componentes globales aparece el texto "Conectado al servidor de SchemaAgent", que especifica con qué servidor se estableció la conexión.
- ahora tiene acceso a todos los esquemas y construcciones de esquema disponibles en la ruta de búsqueda del servidor. Las construcciones de esquema de SchemaAgent, como los element, complexType y simpleType globales, aparecen en negrita y en azul después de las construcciones del esquema activo (**en negrita y en negro**).



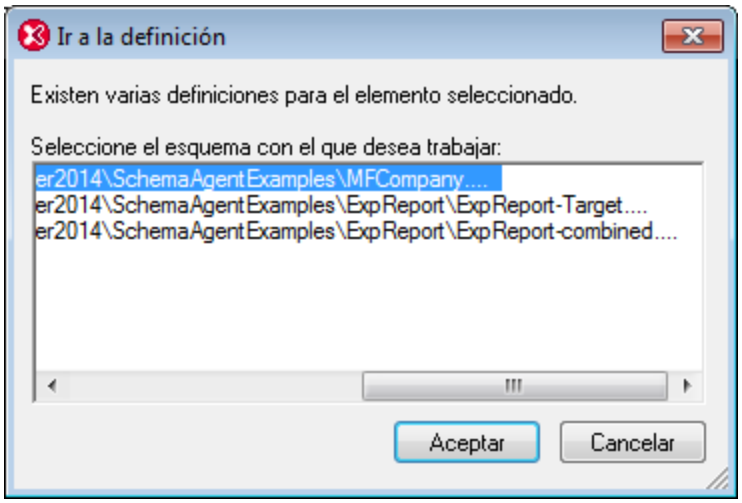
En el ayudante de entrada Componentes de la vista Esquema las construcciones de esquema están organizadas según la clase de componente (en la pestaña Globales), su espacio de nombres o sus restricciones de identidad.

6.7.2 Abrir esquemas encontrados en la ruta de búsqueda

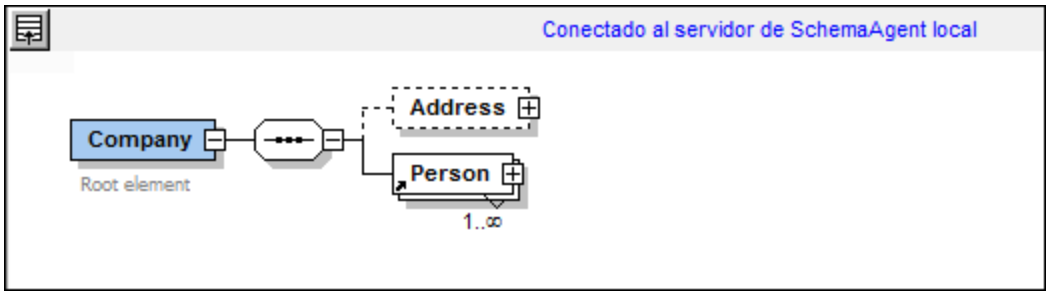
En este apartado explicamos cómo se abre un esquema situado en una ruta de búsqueda definida en SchemaAgent Server. Para ello usamos como esquema activo el archivo de ejemplo `DB2schema.xsd`, que se encuentra en la carpeta `..\Tutorial`. Después de abrir el esquema en la vista Esquema, asegúrese de que está activa la pestaña `Globales` del ayudante de entrada Componentes.

En el ayudante de entrada Componentes, desplácese hasta encontrar la entrada `Company` de color azul y haga doble clic en ella. Se abre el cuadro de diálogo "Ir a la definición".

© 2016-2022 Altova GmbH



Haga clic en la entrada `MFCompany.xsd` y haga clic en **Aceptar** para confirmar. A continuación se abre el esquema `MFCompany.xsd` y se presenta el modelo de contenido del elemento `Company`.



Nota: al hacer doble clic en una construcción de esquema de SchemaAgent, como un element, un complexType o un simpleType, se abre el esquema asociado (así como otros esquemas incluidos) en XMLSpy.

6.7.3 Relaciones IIR

La especificación XML Schema ofrece instrucciones Import, Include y Redefine (IIR) para ayudar a modular los esquemas. Cada método tiene requisitos diferentes en cuanto a los espacios de nombres. El SchemaAgent Client y XMLSpy verifican estos requisitos automáticamente cuando el usuario intenta crear instrucciones IIR.

Instrucciones Import, Include y Redefine (IIR)

Las construcciones de esquema se pueden "insertar" de varias maneras:

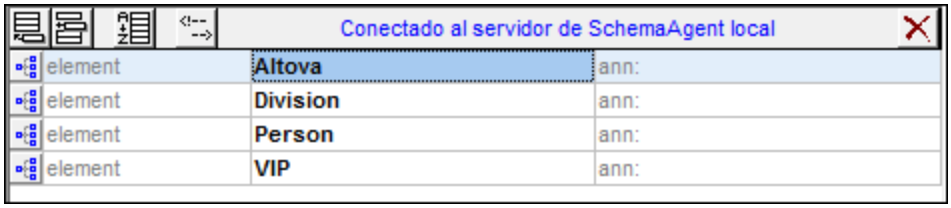
- Los Element globales se pueden arrastrar directamente desde el ayudante de entrada Componentes hasta el modelo de contenido de un componente del esquema (en la vista Esquema).
- Algunos componentes, como los complexType y simpleType, se pueden seleccionar en un cuadro de lista que se abre automáticamente al definir elementos o atributos nuevos, etc.
- Algunos componentes, como los complexType, se pueden seleccionar en el ayudante de entrada Detalles al crear o actualizar este tipo de construcciones.

Incorporar componentes de esquema

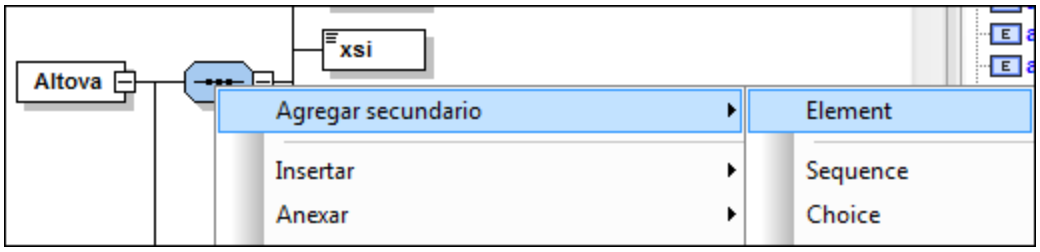
Para este ejemplo utilizamos como esquema activo el archivo `DB2schema.xsd`, que está en la carpeta `..\Tutorial`. La pestaña `Globales` del ayudante de entrada Componentes está activa.

Para usar construcciones de esquema de esquemas de SchemaAgent Server:

Compruebe que está conectado a un servidor de SchemaAgent (consulte el apartado Conectarse a un servidor de SchemaAgent 464).
Abra el archivo `DB2Schema.xsd` y guárdelo con el nombre `Altova-office`.

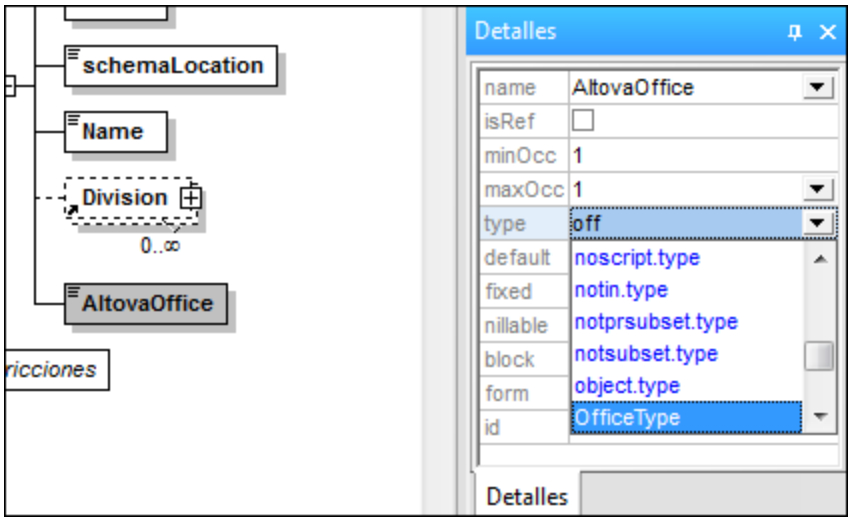


Haga clic en el icono del elemento `Altova` en la vista general del esquema para ver su modelo de contenido.
Haga clic con el botón derecho en el compositor de secuencia del elemento `Altova` y seleccione la opción **Agregar secundario | Element**. Automáticamente se abre un cuadro de lista con todos los elementos globales ubicados en la ruta del servidor. Para insertar un elemento de la lista, basta con hacer clic en el elemento.

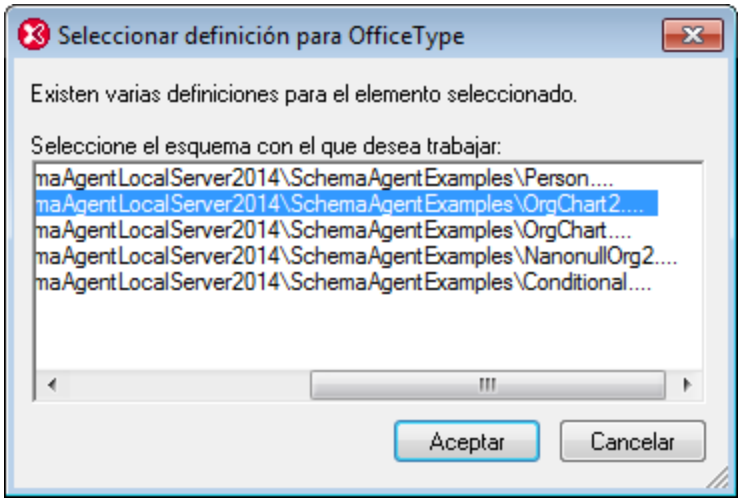


Escriba `Altova-office`, como nombre de este elemento nuevo, y pulse la tecla **Entrar**.
En el ayudante de entrada Detalles, haga clic en el cuadro combinado `type` y seleccione la opción `OfficeType`.

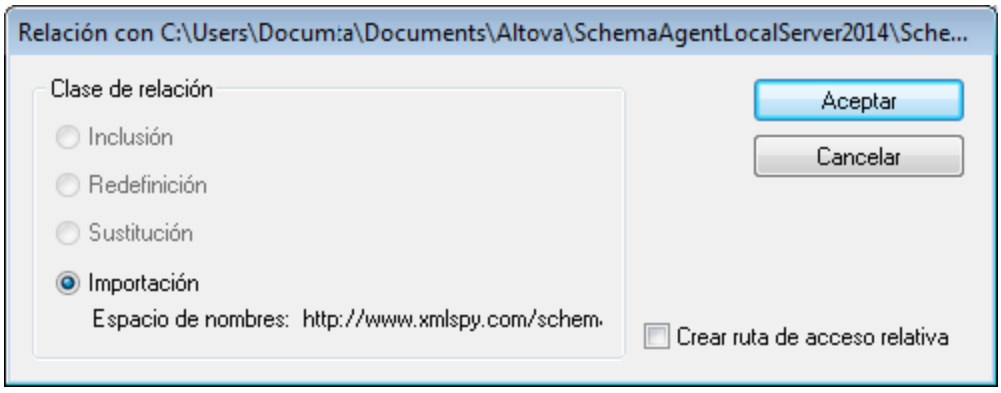
© 2016-2022 Altova GmbH



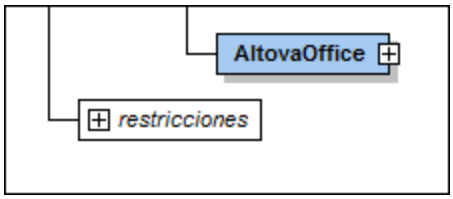
Aparece el cuadro de diálogo "Seleccionar definición para OfficeType".



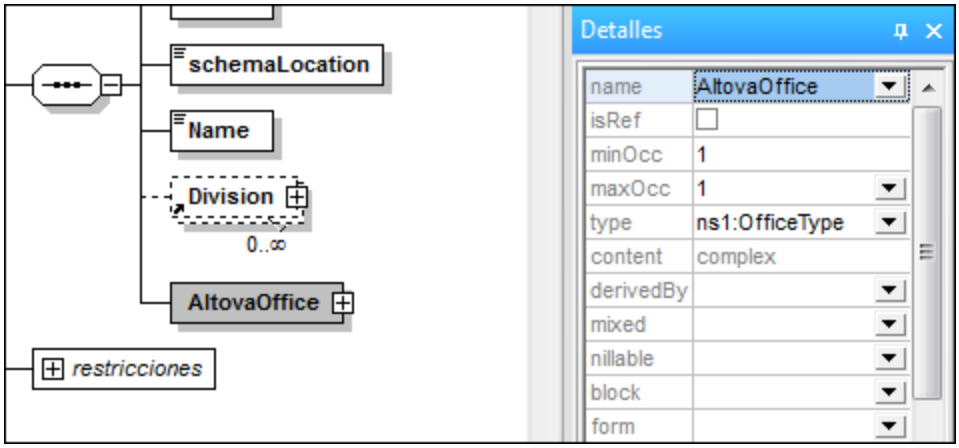
Seleccione el esquema `OrgChart2.xsd` y haga clic en **Aceptar**.



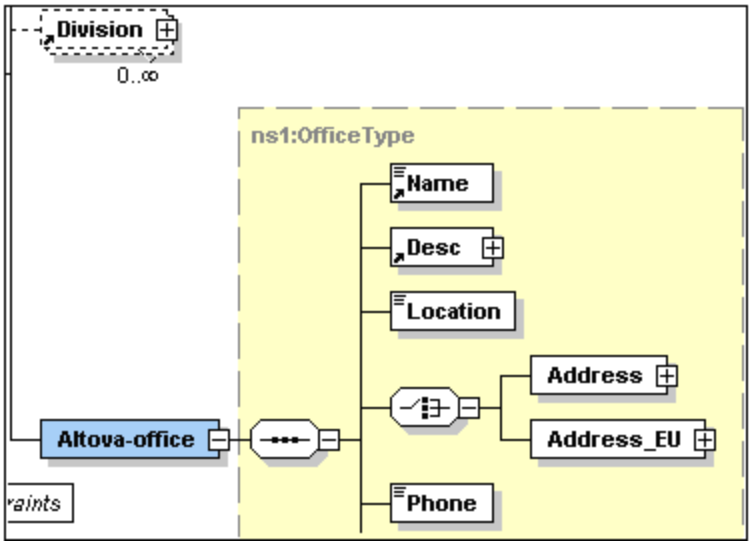
8. Haga clic en **Aceptar**. El comando Importación estaba seleccionado por defecto. En el elemento `Altova-office` aparece un icono +.



Nota: el campo `type` del ayudante de entrada Detalles ha cambiado. Ahora muestra el tipo `ns1:OfficeType` porque el archivo de esquema `OrgChart.xsd` se importó y los esquemas no pueden tener el mismo espacio de nombres de destino. Además se añadió un comando Import al esquema. Además se añadió un comando Import al esquema.



Haga clic en el icono + del elemento Altova-office para ver el modelo de contenido de `OfficeType`.



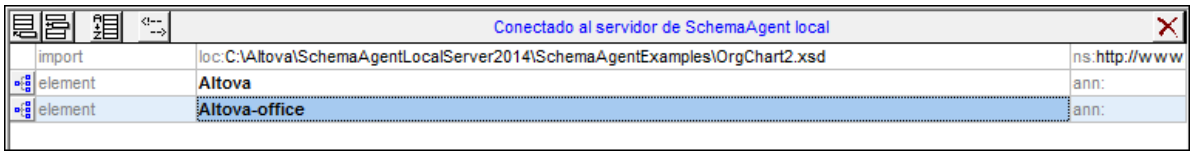
Pulse la tecla **F8** para validar el esquema. En la ventana de mensajes debería aparecer el mensaje "El esquema es válido".

© 2016-2022 Altova GmbH

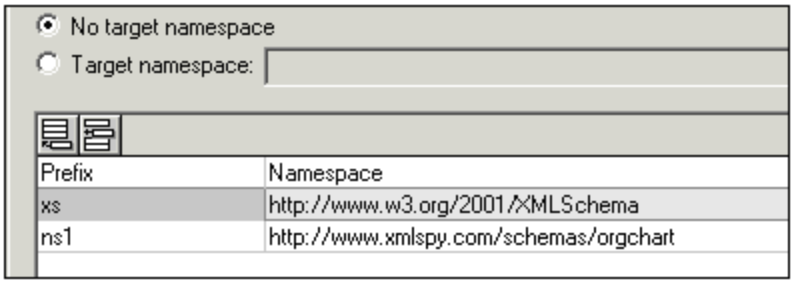
Limpiar el esquema

Elimine el elemento `Division` del modelo de contenido.

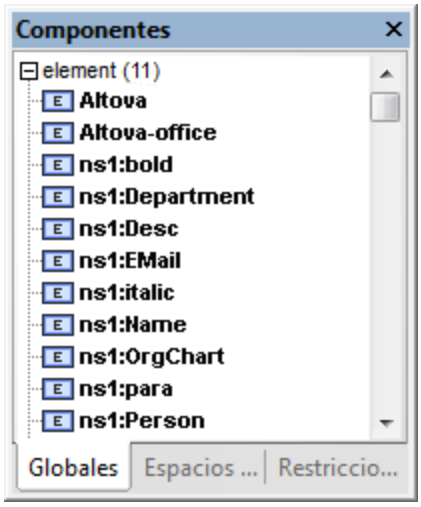
Haga clic en el icono "Mostrar nodos globales" para volver a la vista general del esquema.
Elimine los elementos globales: `Division`, `Person` y `VIP`.



Seleccione el comando de menú **Diseño de esquemas | Configuración del esquema** para ver la nueva configuración de los espacios de nombres



El prefijo `ns1` se añade automáticamente al espacio de nombres `www.xmlspy.com/schemas/orgchart`. Seleccione el comando de menú Diseño de esquemas | Configuración del esquema para ver la nueva configuración de los espacios de nombres.



Nota:

- Los cambios realizados con XMLSpy en los esquemas controlados por SchemaAgent Server actualizan automáticamente los esquemas de la ruta de búsqueda de SchemaAgent Server que hacían referencia al esquema en el que se realizaron los cambios.

- Es posible que aparezcan duplicados de las construcciones element, simpleType, etc. en los ayudantes de entrada (tanto en negro como en azul) si el esquema con el que está trabajando también está en la ruta de búsqueda de SchemaAgent Server.

6.7.4 Visualizar esquemas en SchemaAgent

Para trabajar con el esquema activo y con sus esquemas relacionados de SchemaAgent, seleccione el comando de menú **Diseño de esquemas | Mostrar en SchemaAgent | El archivo y todos los archivos de esquema relacionados** *(ver imagen siguiente).*
Puede elegir entre abrir en SchemaAgent sólo el esquema activo (comando **Sólo el archivo**) o el esquema más (i) todos los esquemas a los que se hace referencia directa, o (ii) todos los esquemas que hacen referencia directa o (iii) todos los esquemas relacionados directamente.

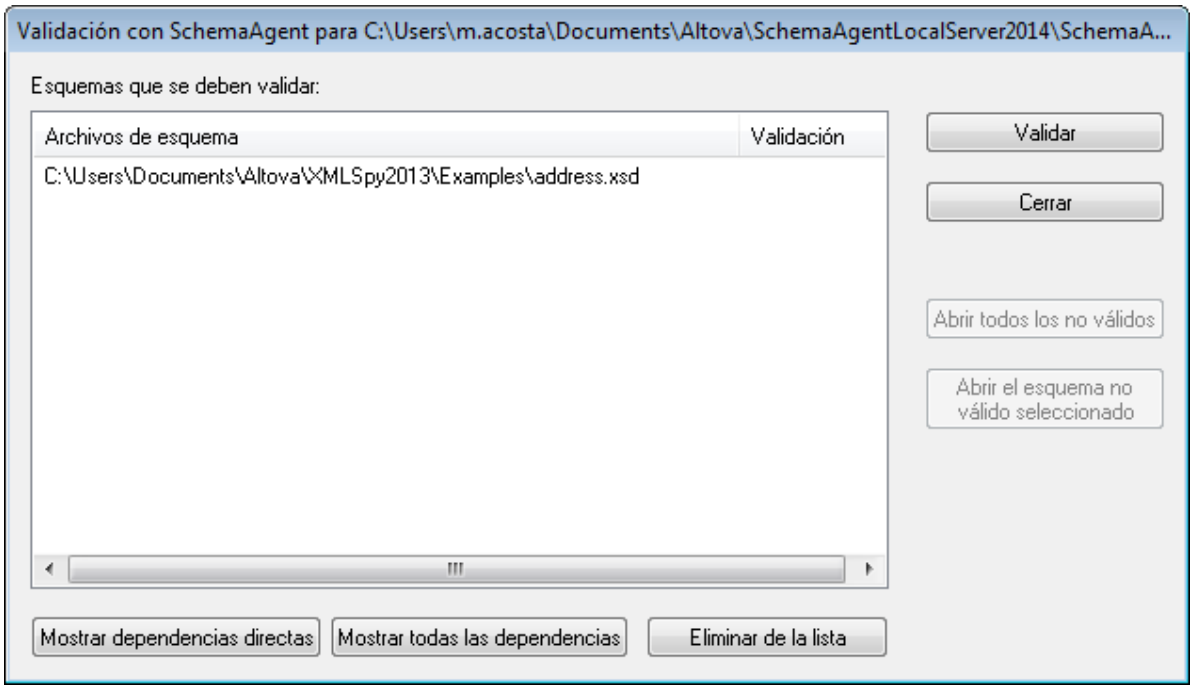
6.7.5 Validación con SchemaAgent

XMLSpy, en colaboración con SchemaAgent, permite validar no sólo el esquema activo, sino también los esquemas relacionados con el esquema activo. Esto recibe el nombre de *validación extendida*. SchemaAgent distingue entre dos tipos de esquemas relacionados a la hora de realizar la validación extendida de esquemas: (i) esquemas con dependencias directas (es decir, a los que se hace referencia directa o los que hacen referencia directa) y (ii) todos los esquemas con dependencias (es decir, los esquemas con dependencias directas y los esquemas que incluyen dependencias indirectas, es decir, el conjunto de esquemas relacionados con otro esquema mediante un esquema intermediario).

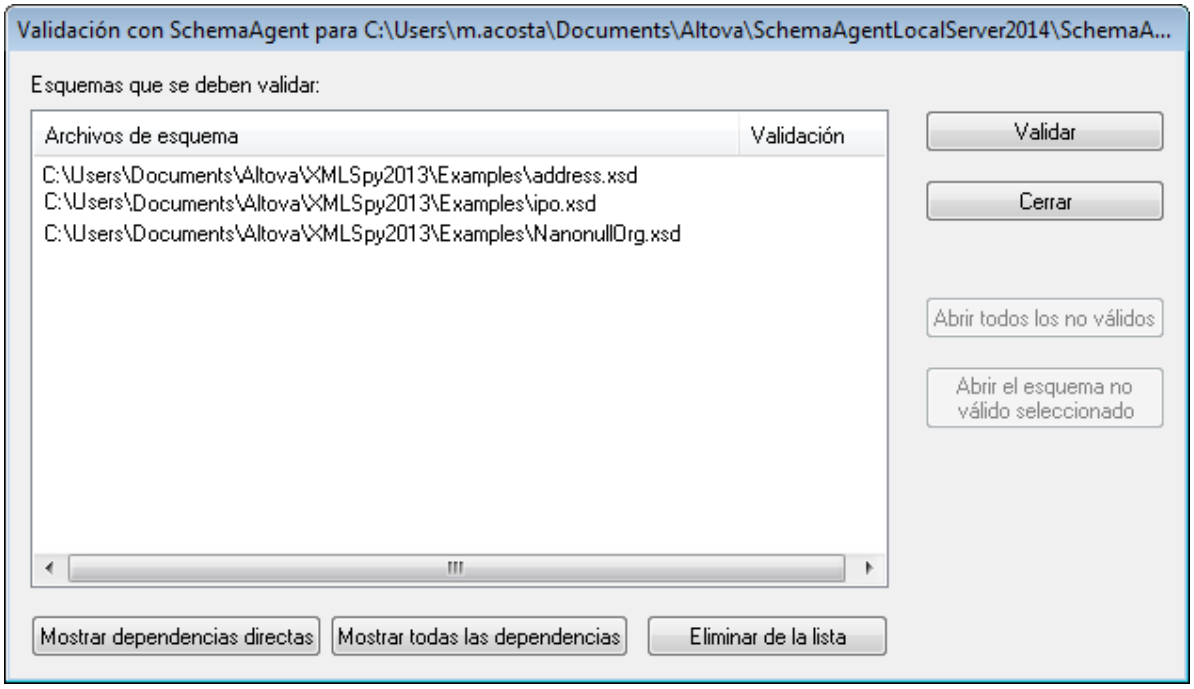
A continuación explicamos cómo usar la función de validación extendida de esquemas. Para ello asumimos que el archivo `address.xsd` es el esquema activo en la vista Esquema de XMLSpy. Asegúrese de que las rutas de búsqueda de SchemaAgent Server contienen el archivo activo y los archivos dependientes. De lo contrario no se habilitará el comando **Validación con SchemaAgent**. El proceso es el siguiente:

1. Haga clic en el icono **Validación con SchemaAgent**  de la barra de herramientas o seleccione el comando de menú **Diseño de esquemas | Validación con SchemaAgent**. A continuación aparece el cuadro de diálogo "Validación con SchemaAgent" (*imagen siguiente*), en el que puede elegir si se valida sólo el esquema activo o también los esquemas relacionados.

© 2016-2022 Altova GmbH

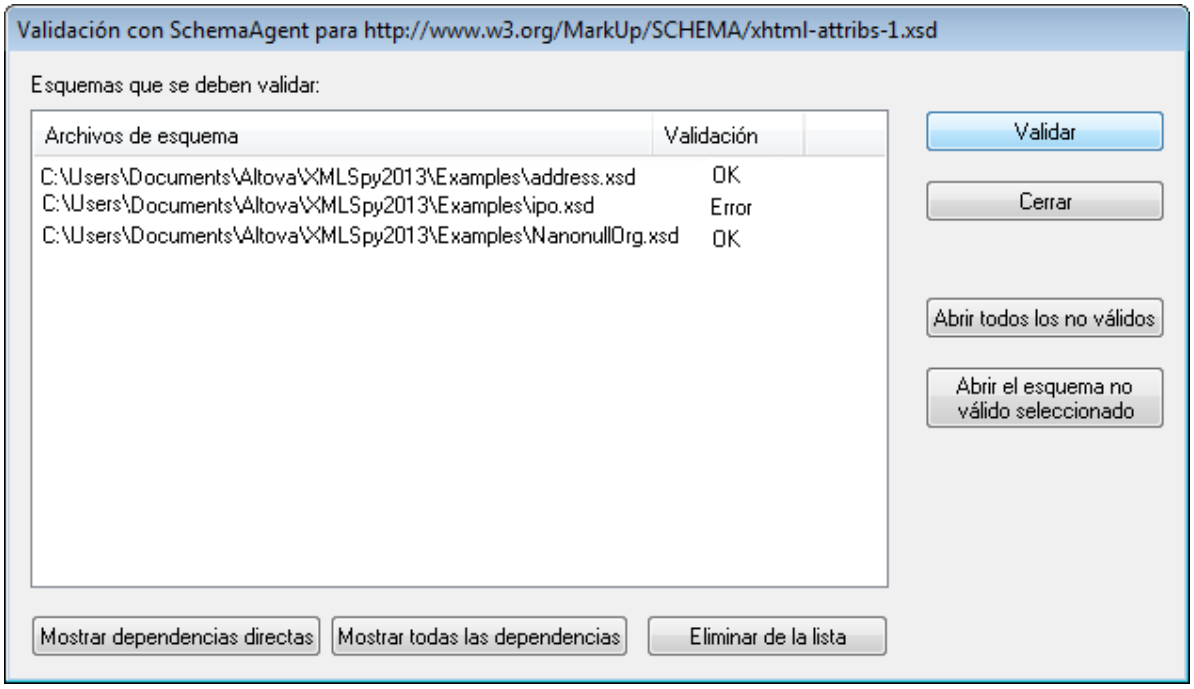


2. Para insertar esquemas en la lista, haga clic en el botón **Mostrar dependencias directas** (esquemas a los que se hace referencia directa) o **Mostrar todas las dependencias** (esquemas a los que se hace referencia directa o indirecta) según corresponda. Para este ejemplo, hacemos clic en el botón **Mostrar todas las dependencias** y se insertan en la lista todos los archivos a los que se hace referencia.



Si lo desea, puede eliminar esquemas de la lista (con el botón **Eliminar de la lista**).

3. Haga clic en el botón **Validar** para validar todos los esquemas de la lista.



La columna *Validación* indica si la validación se realizó correctamente o si se produjeron errores de validación.

Ahora puede abrir en XMLSpy todos los esquemas que no superaron la validación o los esquemas no válidos que seleccione en la lista.

© 2016-2022 Altova GmbH

6.8 Búsqueda en esquemas

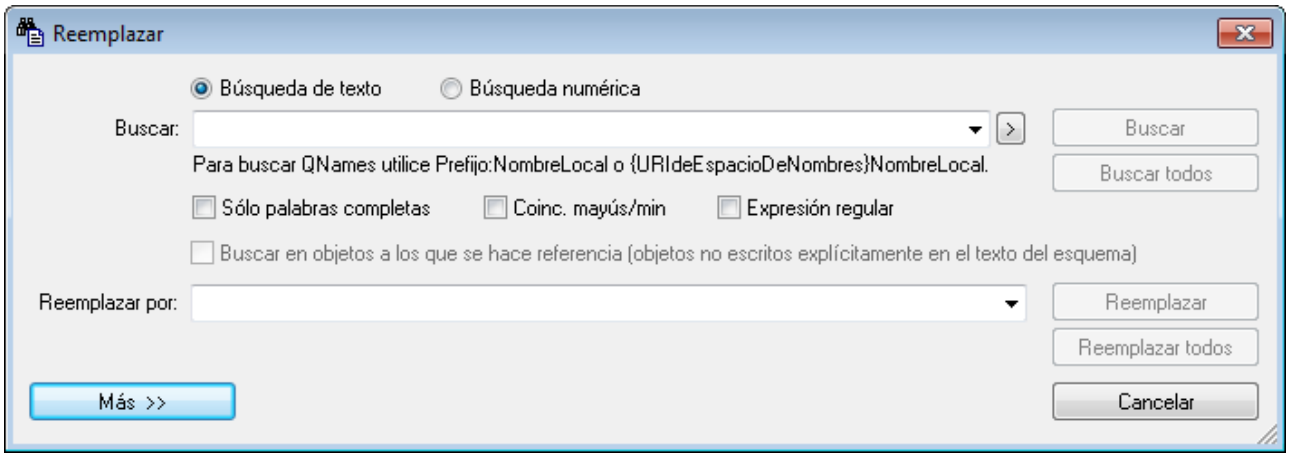
En la vista Esquema puede realizar búsquedas inteligentes en los esquemas con la función de búsqueda y reemplazo de XMLSpy.

La función de búsqueda y reemplazo de la vista Esquema se habilita cuando está activo un esquema en la vista Esquema. Puede acceder a esta función de dos maneras:

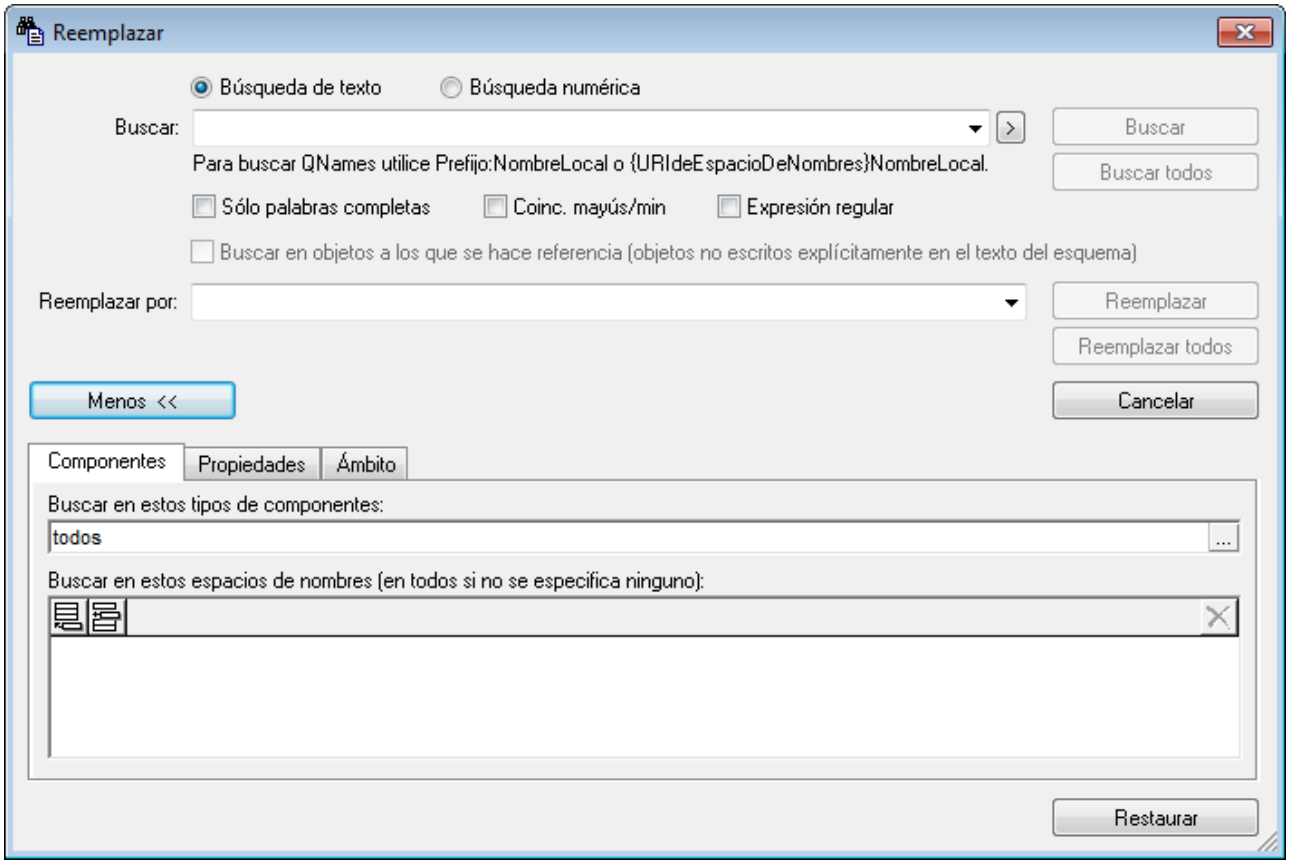
- Con los comandos **Edición | Buscar** y **Edición | Reemplazar.**
- Con los botones **Buscar** y **Reemplazar** de la ventana Buscar en esquemas.

Al hacer clic en estos comandos o botones se abre el cuadro de diálogo "Buscar" o "Reemplazar", dependiendo del botón utilizado. El cuadro de diálogo "Reemplazar" (*ver imagen siguiente*) se diferencia del cuadro de diálogo "Buscar" porque contiene un campo de entrada de texto para el término de reemplazo.

Este es el aspecto del cuadro de diálogo "Reemplazar":



Al hacer clic en el botón **Más** se expande el cuadro de diálogo, que incluye criterios de búsqueda adicionales (*ver imagen siguiente*).



Para usar este cuadro de diálogo:

- Escriba los términos de búsqueda y reemplazo[477] en los campos *Buscar* y *Reemplazar por*
- En la pestaña Componentes indique en qué componentes del esquema se debe buscar[478]
- Especifique las propiedades de los componentes en las que se debe buscar[480]. Esto ayuda a acotar la búsqueda
- Fije el ámbito de la búsqueda[483] en el documento o proyecto actual o indique en qué carpeta se debe realizar la búsqueda
- Ejecute el comando[484]
- Use la ventana Buscar en esquemas[486] para navegar hasta los componentes encontrados

El botón **Restaurar** situado en la parte inferior del cuadro de diálogo restaura las opciones de configuración predeterminadas, que son:

- Ningún término de búsqueda y ningún término de reemplazo
- Componentes: `todos`
- Espacios de nombres: ninguno
- Restricciones de propiedad: `en cualquier parte`
- Restricciones adicionales de propiedad: ninguna
- Ámbito: el archivo actual

Nota: en el campo *Reemplazar por* no se admite el uso de expresiones regulares.

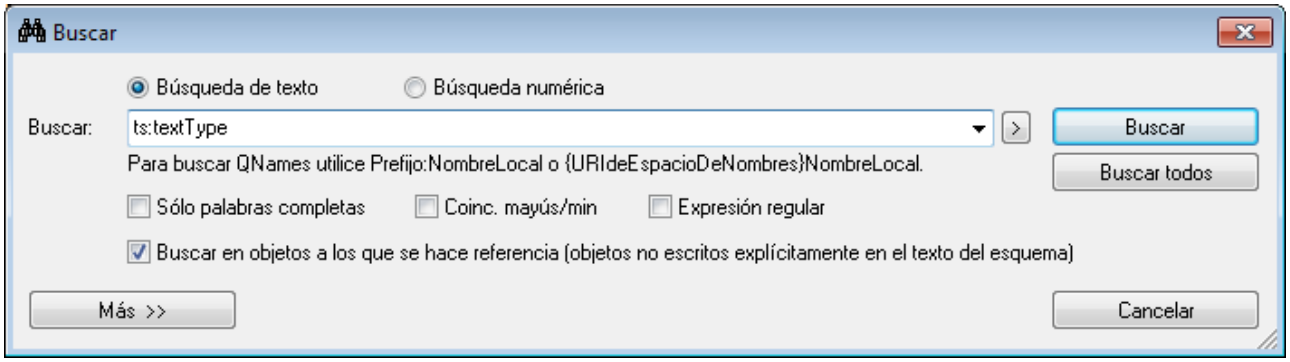
© 2016-2022 Altova GmbH

6.8.1 Término de búsqueda

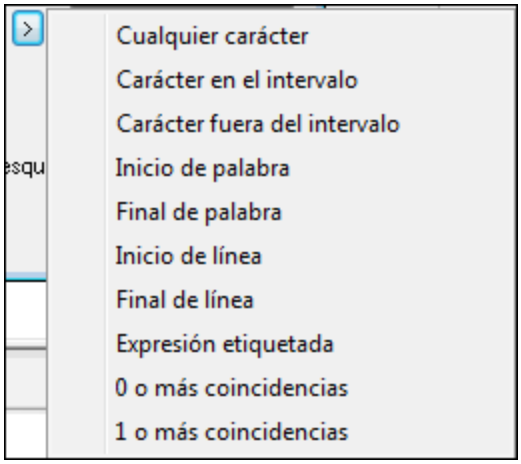
El término de búsqueda puede ser una cadena de texto (seleccione el botón de opción **Búsqueda de texto**) o un número (botón de opción **Búsqueda numérica**).

Búsqueda de texto

En las búsquedas de texto (*ver imagen siguiente*), el término de búsqueda puede ser: (i) una cadena de texto, (ii) un QName o (iii) una expresión regular. En las búsquedas de QNames, el espacio de nombres se determina en base al prefijo usado en el documento o al URI del espacio de nombres, que debe introducirse en ambos casos. En la imagen siguiente, el prefijo `ts:` es el prefijo usado en el documento para identificar un determinado espacio de nombres.



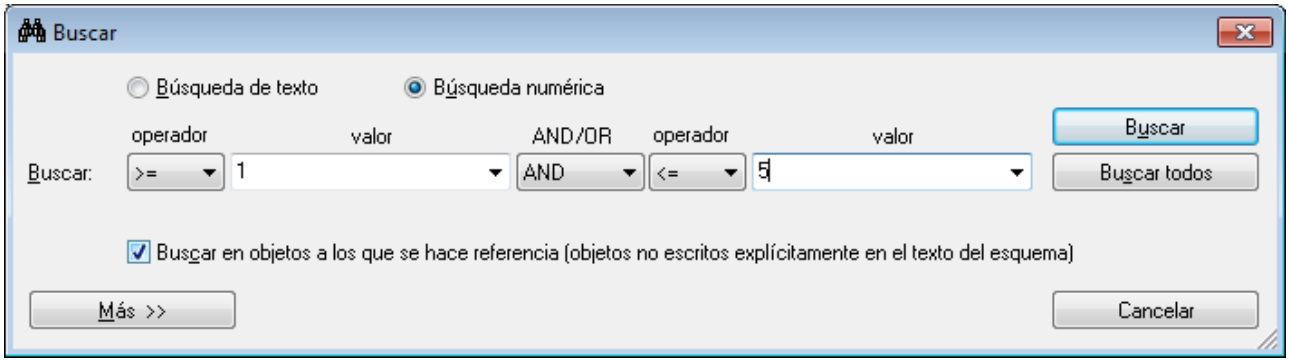
Para usar una expresión regular en la búsqueda, marque la casilla *Expresión regular* y después escriba la expresión regular. Si hace clic en la punta de flecha, situada a la derecha del campo *Buscar*, se abre el ayudante de entrada para expresiones regulares (*ver imagen siguiente*).



También puede indicar si el término de búsqueda debe coincidir con una palabra completa del documento y si debe coincidir en el uso de mayúsculas y minúsculas. Estas opciones se pueden configurar con las casillas de verificación situadas debajo del campo *Buscar*.

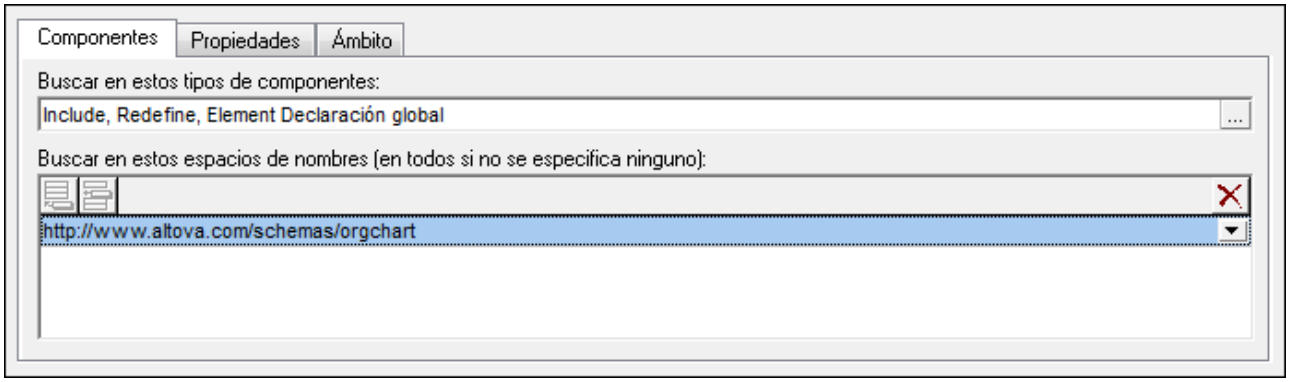
Si desea buscar en los objetos a los que se hace referencia (como una definición de complexType o en un elemento global), marque la casilla *Buscar en objetos a los que se hace referencia*. Esta opción está disponible en el cuadro de diálogo "Buscar", pero en el cuadro de diálogo "Reemplazar" está deshabilitada.

Búsqueda numérica



6.8.2 Componentes

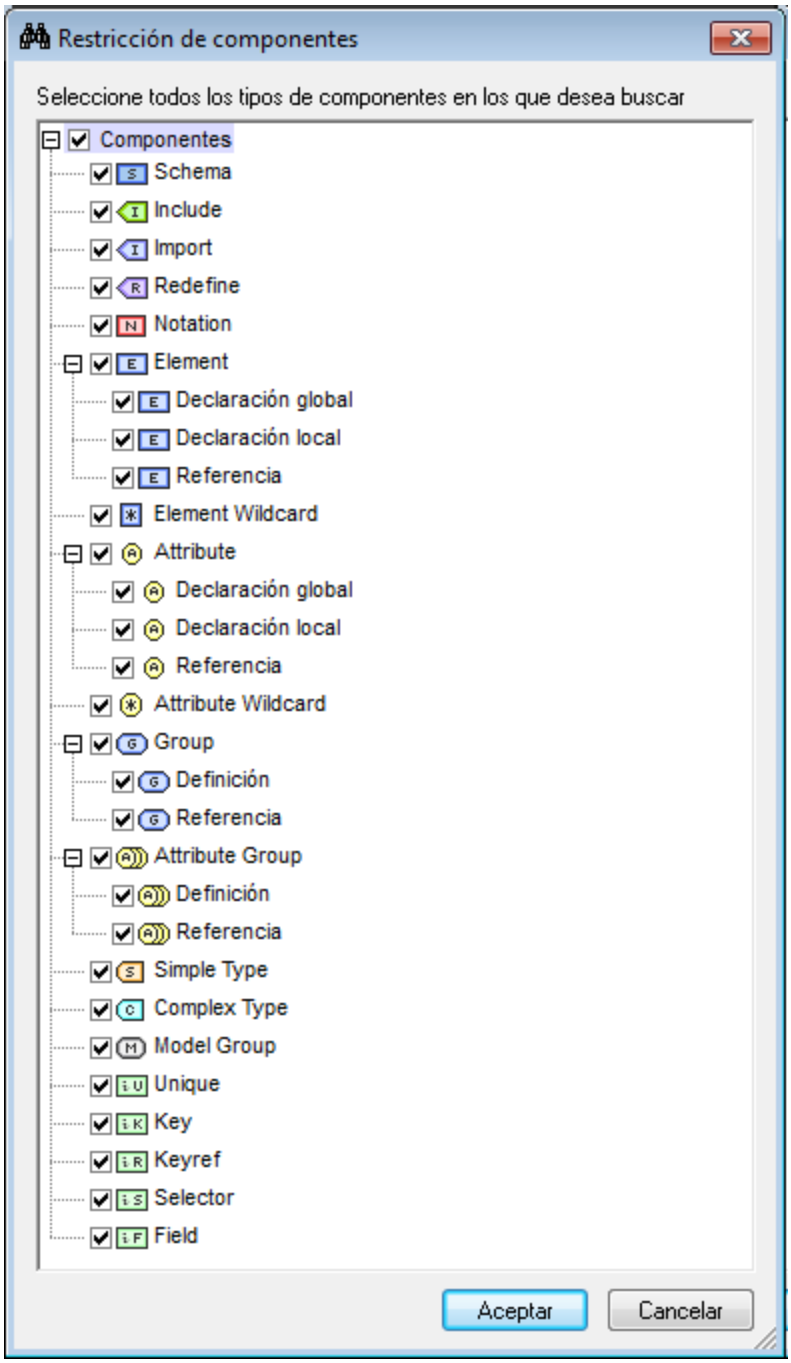
Puede limitar la búsqueda a ciertos tipos de componentes y a ciertos espacios de nombres de destinos. Estas opciones están disponibles en la pestaña Componentes. Al hacer clic en el botón **Más** se expande el cuadro de diálogo "Buscar" o "Reemplazar" y aparecen 3 pestañas. Estas pestañas sirven para acotar la búsqueda y una de ellas es la pestaña Componentes (*ver imagen siguiente*).



La pestaña Componentes se divide en dos partes: (i) una parte para seleccionar los tipos de componentes donde se debe realizar la búsqueda y (ii) otra parte para seleccionar los espacios de nombres de destino donde se debe realizar la búsqueda.

© 2016-2022 Altova GmbH

Seleccionar los componentes



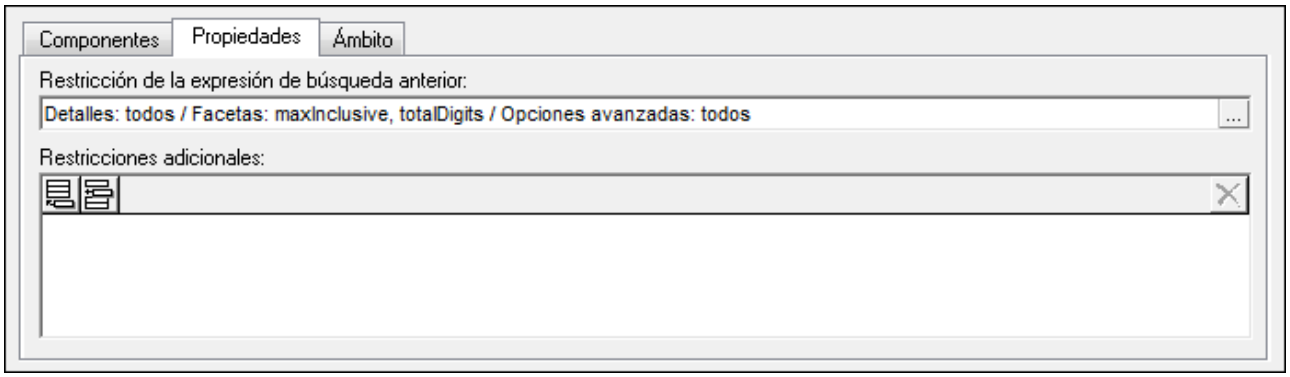
Nota: tenga en cuenta que cuando se abren los cuadros de diálogo "Buscar" y "Reemplazar", la pestaña Componentes aún incluye la selección de componentes de la última búsqueda realizada.

Seleccionar el espacio de nombres

Para seleccionar en qué espacios de nombres de destino se debe realizar la búsqueda, utilice los iconos **Añadir** o **Insertar** y escriba los espacios de nombres en los que desea buscar. Si no se indica ningún espacio de nombres de destino, entonces la búsqueda se realiza en todos los espacios de nombres de destino. Para eliminar un espacio de nombres de destino del panel, seleccione el espacio de nombres de destino y haga clic en el icono **Eliminar**.

6.8.3 Propiedades

Puede limitar la búsqueda a ciertas propiedades de los componentes (detalles y facetas) usando las opciones de la pestaña Propiedades. También se puede solicitar que el término de búsqueda coincida con el contenido de las propiedades. Haga clic en el botón **Más** para expandir el cuadro de diálogo "Buscar" o "Reemplazar" y después seleccione la pestaña Propiedades (*imagen siguiente*).

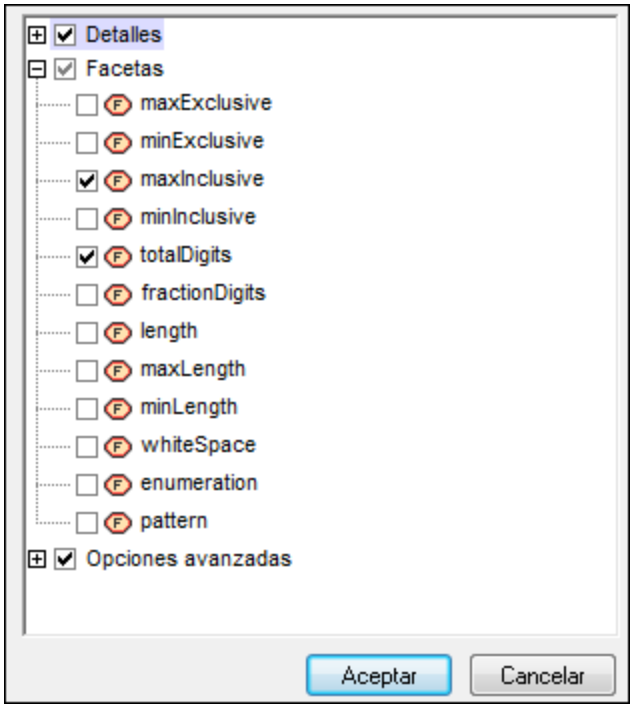


La Pestaña propiedades se divide en dos partes: (i) una parte para limitar el término de búsqueda principal (escrito en el cuadro de texto *Buscar*) y (ii) otra parte para añadir restricciones de contenido adicionales (que tienen su propio término de búsqueda). Para más información consulte el subapartado Restricciones adicionales 481 que aparece más adelante.

Selección de las propiedades

Para seleccionar las propiedades en las que se debe realizar la búsqueda, haga clic en el icono  situado a la derecha del campo de entrada texto (*ver imagen anterior*). Al hacer clic en este icono se abre el cuadro de diálogo "Restricciones de propiedad" (*ver imagen siguiente*), donde puede marcar en qué propiedades se debe buscar. Las propiedades están organizadas en tres grupos: (i) detalles, (ii) facetas y (iii) opciones avanzadas (como la propiedad `DerivedFrom`). Si marca las casillas `Detalles`, `Facetas` u `Opciones avanzadas`, se seleccionan todas las propiedades de ese grupo. Si desactiva las casillas `Detalles`, `Facetas` u `Opciones avanzadas`, se anula la selección de las propiedades de ese grupo. Por tanto, para seleccionar propiedades concretas, es necesario desactivar antes la casilla principal del grupo al que pertenezca la propiedad. Las propiedades seleccionadas se introducen en el campo de texto (*ver imagen anterior*).

© 2016-2022 Altova GmbH

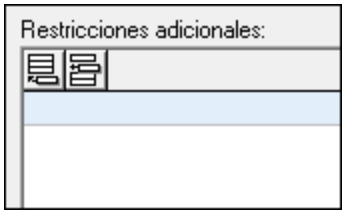


Nota: tenga en cuenta que cuando se abren los cuadros de diálogo "Buscar" y "Reemplazar", la pestaña Propiedades aún incluye la selección de propiedades de la última búsqueda realizada.

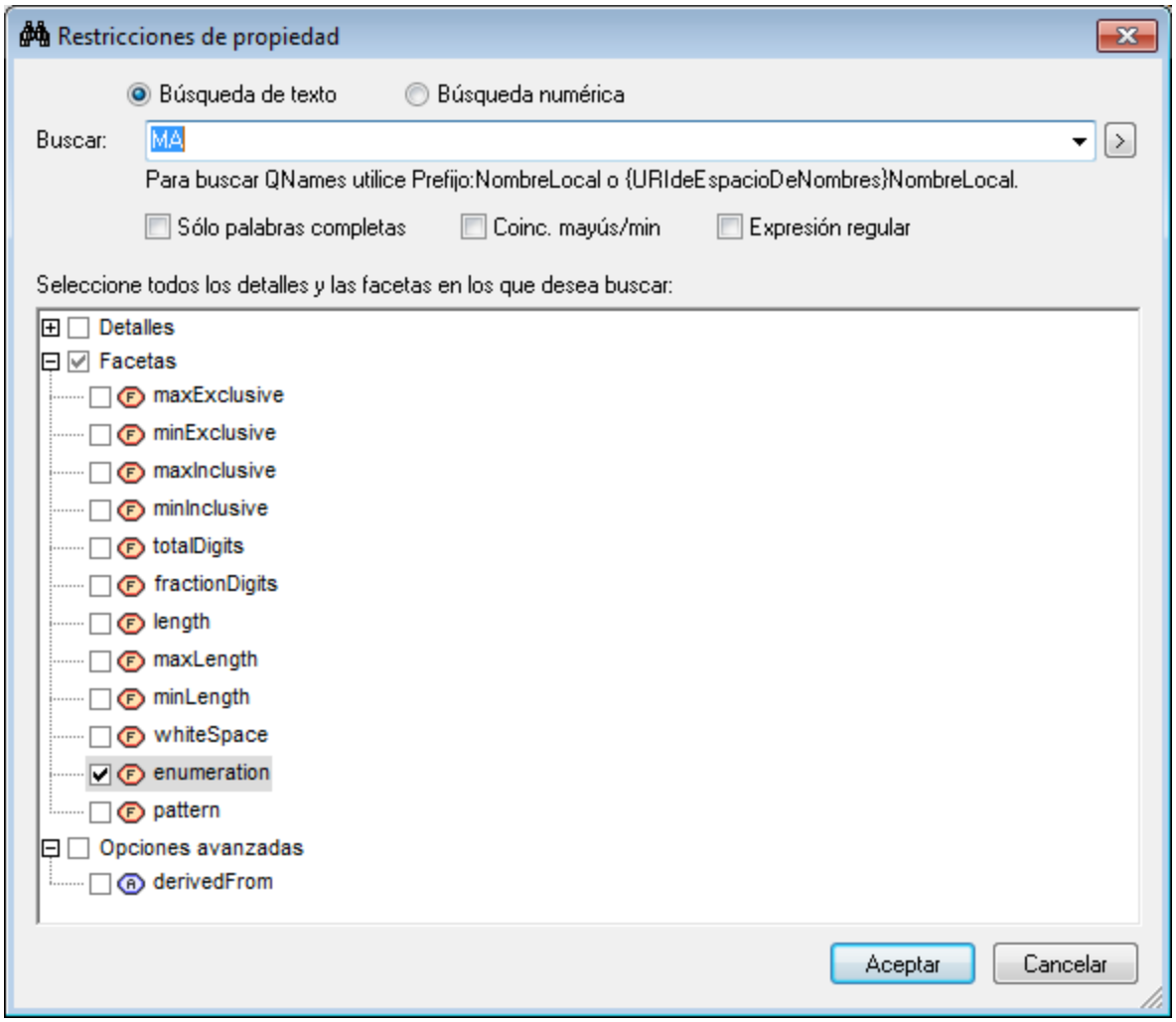
Restricciones adicionales

El panel "Restricciones adicionales" sirve para especificar el valor de la propiedad en la que se debe buscar. Por ejemplo, si busca un elemento llamado `state` que tenga una enumeración `MA` (de Massachusetts, el estado de EE UU), podría limitar la búsqueda especificando el valor `MA` de la propiedad `enumeration` en el panel "Restricciones adicionales". Para ello siga estos pasos:

1. En el panel "Restricciones adicionales", haga clic en el icono **Añadir** o **Insertar** (*ver imagen siguiente*).

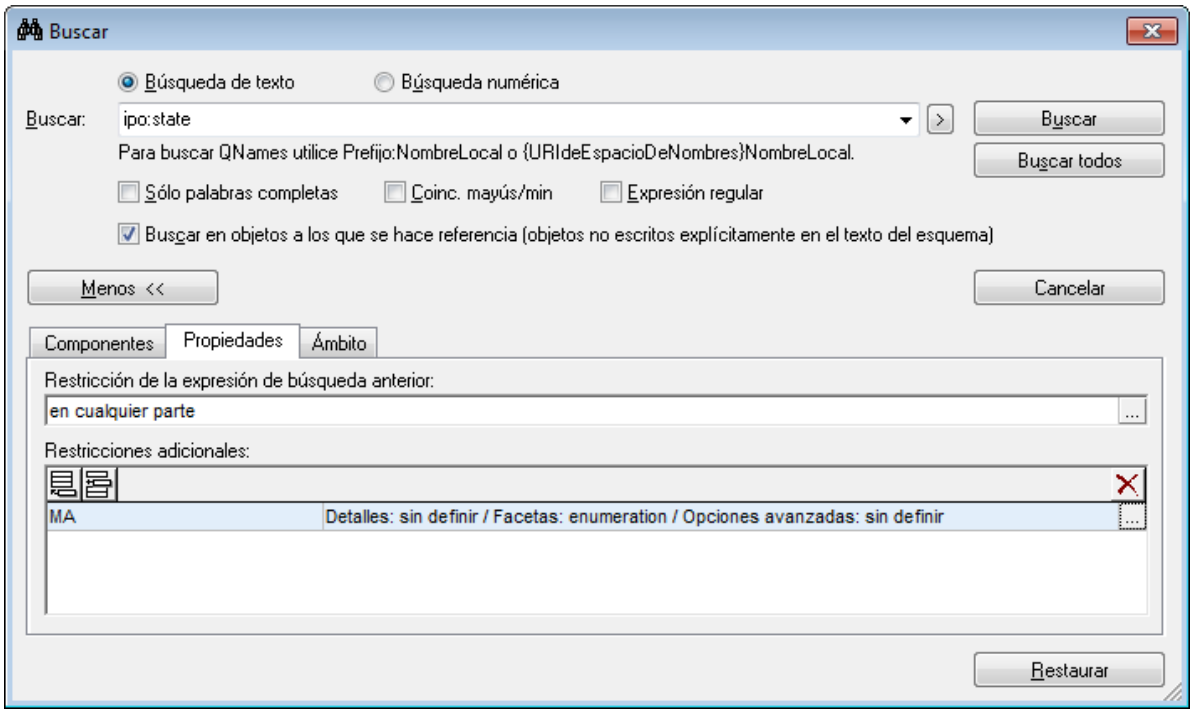


2. En el panel se añade una fila y se abre el cuadro de diálogo "Restricciones de propiedad". Desactive todas las propiedades y seleccione sólo la propiedad `enumeration` (*ver imagen siguiente*).



3. En el campo de texto de la parte superior del diálogo, escriba el valor de la enumeración que se debe buscar, en este caso, `MA` (*ver imagen anterior*).
4. Haga clic en **Aceptar**. La restricción adicional se añade en la fila recién creada del panel "Restricciones adicionales" (*ver imagen siguiente*).

© 2016-2022 Altova GmbH



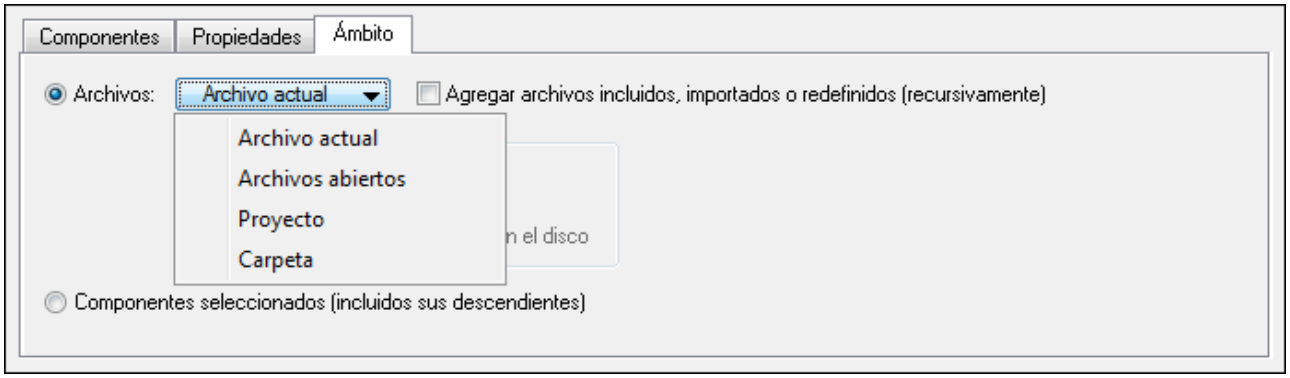
Observe que en la imagen anterior el término de búsqueda es `ipo:state`. En la pestaña Propiedades la entrada `en cualquier parte` indica que la búsqueda se debe realizar en todas las propiedades, pero la restricción adicional indica que la búsqueda se debe limitar a las enumeraciones cuyo valor sea `MA`.

Puede añadir más restricciones adicionales para seguir acotando la búsqueda. Para eliminar una restricción adicional, selecciónela y después haga clic en el icono **Eliminar**.

Nota: tenga en cuenta que cuando se abren los cuadros de diálogo "Buscar" y "Reemplazar", la pestaña Propiedades aún incluye las restricciones adicionales de la última búsqueda realizada.

6.8.4 Ámbito

Puede fijar el ámbito de la búsqueda en la pestaña Ámbito (*ver imagen siguiente*). Puede seleccionar si la búsqueda se limita a determinados archivos o al componente seleccionado en la vista Esquema.



Si se selecciona la opción *Archivos* podrá elegir una de estas opciones:

- *Archivo actual:* que incluye una opción adicional para buscar en los archivos incluidos, importados y redefinidos.
- *Archivos abiertos*: se busca en todos los archivos de esquema XML (XSD) que están abiertos en XMLSpy. Sólo están habilitados los botones **Buscar todos** y **Reemplazar todos**. La opción de búsqueda paso a paso no está disponible.
- *Proyecto:* se selecciona el proyecto activo y hay una opción para omitir las carpetas externas de la búsqueda. Sólo están habilitados los botones **Buscar todos** y **Reemplazar todos**. La opción de búsqueda paso a paso no está disponible. Si la vista predeterminada para los archivos de tipo `.xsd` no es la vista Esquema (**Herramientas | Opciones | Tipos de archivo | Vista predeterminada**), entonces no se buscará en los archivos `.xsd`.
- *Carpeta:* puede realizar la búsqueda en una carpeta concreta. También hay una opción para buscar en las subcarpetas de la carpeta. Sólo están habilitados los botones **Buscar todos** y **Reemplazar todos**. La opción de búsqueda paso a paso no está disponible. Si la vista predeterminada para los archivos de tipo `.xsd` no es la vista Esquema (**Herramientas | Opciones | Tipos de archivo | Vista predeterminada**), entonces no se buscará en los archivos `.xsd`.
- *Agregar archivos incluidos, importados y redefinidos*: para incluir estos archivos en el ámbito de la búsqueda, marque su casilla de verificación.

En el cuadro de diálogo "Reemplazar" puede elegir si el texto de reemplazo se copia directamente en el archivo del disco o si primero se abre el archivo en XMLSpy. Para ello elija el botón correspondiente del cuadro de diálogo.

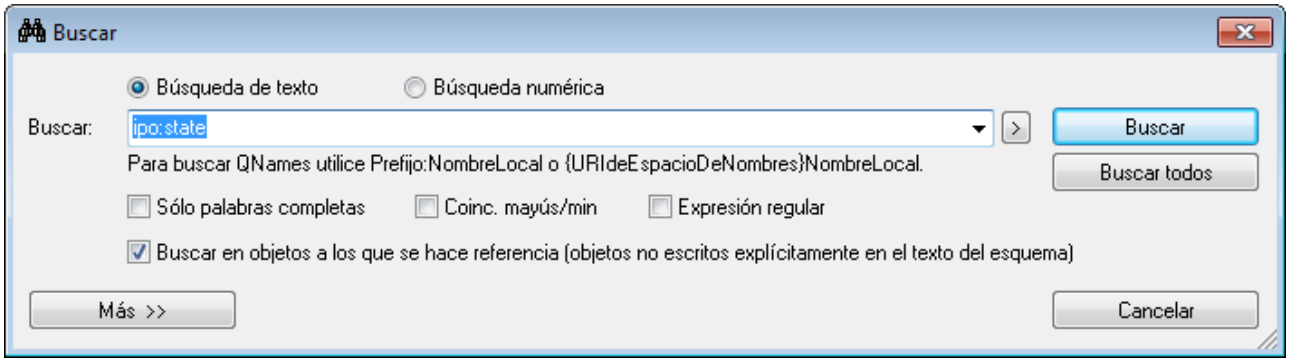
6.8.5 Comandos Buscar y Reemplazar

El comando **Buscar** del cuadro de diálogo "Buscar" no funciona igual que el del cuadro de diálogo "Reemplazar". A continuación describimos cómo funcionan el comando **Buscar** y el comando **Reemplazar** en estos dos cuadros de diálogo.

Cuadro de diálogo "Buscar"

Tras escribir el término de búsqueda e indicar otros criterios para limitar la búsqueda, puede hacer clic en el botón **Buscar** (**F3**) o en el botón **Buscar todos** (*ver imagen siguiente*).

© 2016-2022 Altova GmbH



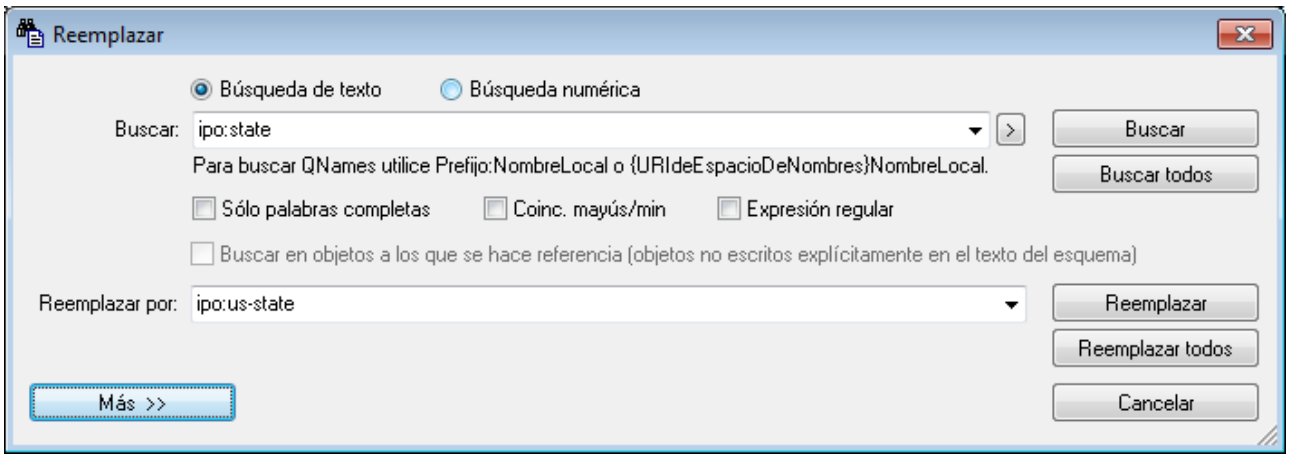
Al hacer clic en el comando **Buscar** (**Ctrl+F**) se cierra el cuadro de diálogo "Buscar" y se encuentra la siguiente repetición del término de búsqueda en el ámbito de búsqueda especificado y según los criterios de búsqueda indicados. La siguiente repetición se encuentra en relación con el componente que está seleccionado en la vista Esquema. Si la búsqueda llega al final del ámbito de búsqueda, la búsqueda no se iniciará de nuevo desde el principio del ámbito de búsqueda. Por tanto, antes de iniciar la búsqueda, compruebe que el componente que está seleccionado en la vista Esquema aparece antes que la parte del documento en la que desea buscar.

El resultado devuelto por el comando **Buscar** se resalta en la vista Esquema y se notifica en la ventana Buscar en esquemas. En la ventana Buscar en esquemas puede hacer clic en el resultado y el componente se resaltará en la vista Esquema.

Al hacer clic en el comando **Buscar todos** se cierra el cuadro de diálogo "Buscar" y la ventana Buscar en esquemas muestra todos los resultados de la búsqueda.

Cuadro de diálogo "Reemplazar"

En el cuadro de diálogo "Reemplazar" (*imagen siguiente*), al hacer clic en el botón **Buscar** se encuentra la siguiente repetición del término de búsqueda en relación con la selección actual de la vista Esquema. Para reemplazar esta repetición del término de búsqueda, haga clic en el botón **Reemplazar**.



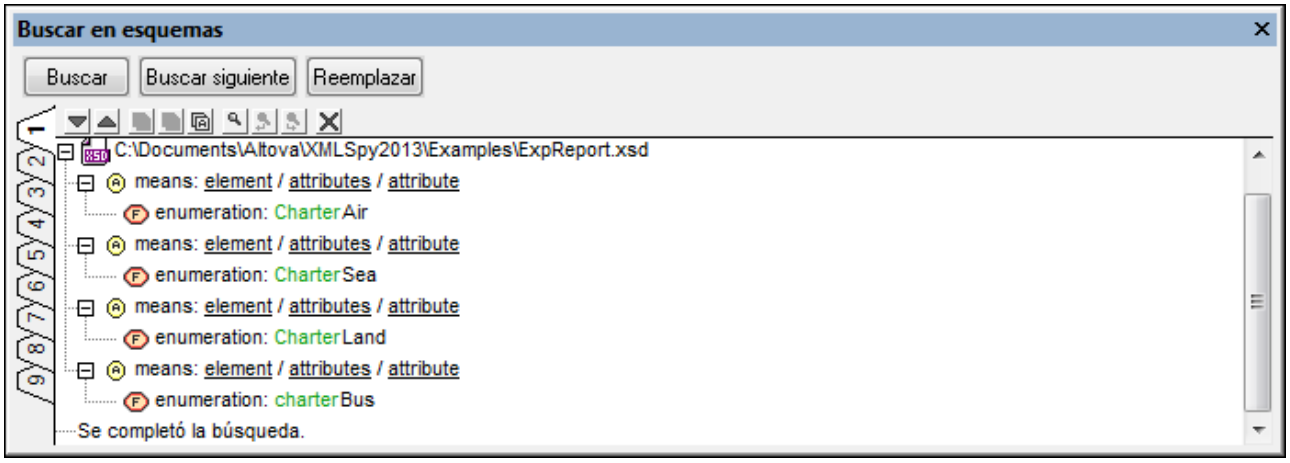
Al hacer clic en el botón **Buscar todos** se cierra el cuadro de diálogo "Reemplazar" y los resultados de la búsqueda aparecen en la ventana Buscar en esquemas.

Al hacer clic en el botón **Reemplazar todos** se reemplazan todas las repeticiones del término de búsqueda, se cierra el cuadro de diálogo "Reemplazar" y aparecen todos los términos encontrados y reemplazados en la ventana Buscar en esquemas.

Nota: el campo de entrada *Reemplazar por* no admite el uso de expresiones regulares.

6.8.6 Resultados e información

Cada vez que se ejecuta el comando **Buscar**, **Buscar todos** o **Reemplazar todos**, los resultados de la ejecución aparecen en la ventana Buscar en esquemas (*imagen siguiente*). El término de búsqueda aparce en color verde (en la imagen siguiente, el término de búsqueda es `charter`, ya sea en mayúsculas o minúsculas). Observe que en los resultados también se indica la ubicación del archivo de esquema.



Los comandos **Buscar todos** y **Reemplazar todos** muestran todas las repeticiones del término de búsqueda encontradas en el ámbito de la búsqueda.

Nota: con los botones **Buscar** y **Reemplazar** situados en la parte superior de la ventana puede abrir los cuadros de diálogo "Buscar" y "Reemplazar" respectivamente. Con el botón **Buscar siguiente** puede buscar la siguiente repetición del término de búsqueda.

Características de la ventana Buscar en esquemas

La ventana Buscar en esquemas incluye 9 pestañas distintas (numeradas del 1 al 9). La función de estas pestañas es que el usuario pueda guardar los resultados de una búsqueda en una pestaña, realizar otra búsqueda en otra pestaña y comparar los resultados de ambas. Al hacer clic en un resultado de la ventana Buscar en esquemas, el componente correspondiente se resalta en la ventana principal de la vista Esquema. Es decir, puede buscar y navegar hasta el componente con gran rapidez.

La barra de herramientas de la ventana Buscar en esquemas incluye estos comandos:

- Los iconos **Siguiente** y **Anterior** seleccionan respectivamente el mensaje siguiente y el mensaje anterior al mensaje seleccionado.
- Los comandos **Copiar mensaje** copian en el portapapeles el mensaje seleccionado, el mensaje seleccionado y sus mensajes secundarios o todos los mensajes.

© 2016-2022 Altova GmbH

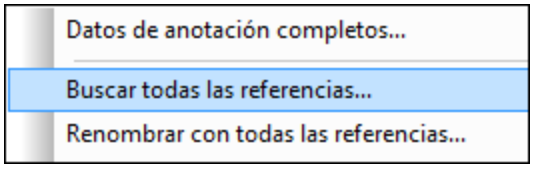
- Los comandos **Buscar** buscan texto dentro de la ventana Buscar en esquemas.
- El comando **Borrar** elimina todos los mensajes de la pestaña activa.

6.8.7 Buscar y renombrar componentes globales

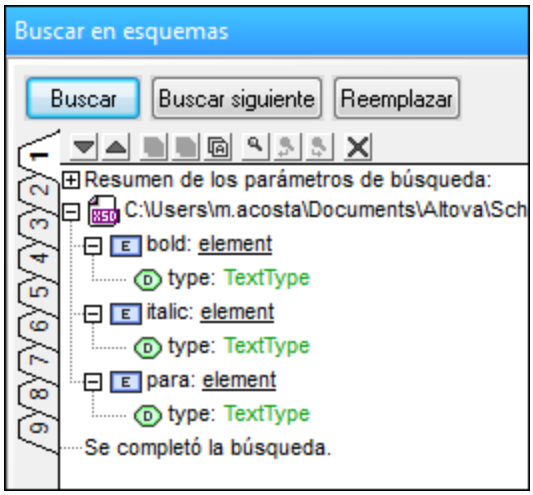
Los componentes globales con nombre del esquema XML se pueden buscar y renombrar, ya sea en un archivo determinado o en todos los archivos de esquema relacionados con un archivo dado. Los componentes globales con nombre son todos los componentes globales excepto Include, Import, Redefine, Annotation, los comentarios y las instrucciones de procesamiento.

El proceso funciona de la siguiente manera:

1. En la vista general del esquema, seleccione el componente global que desea buscar o renombrar.
2. Haga clic con el botón derecho en el componente y seleccione el comando **Buscar todas las referencias** o **Renombrar con todas las referencias**, según corresponda.



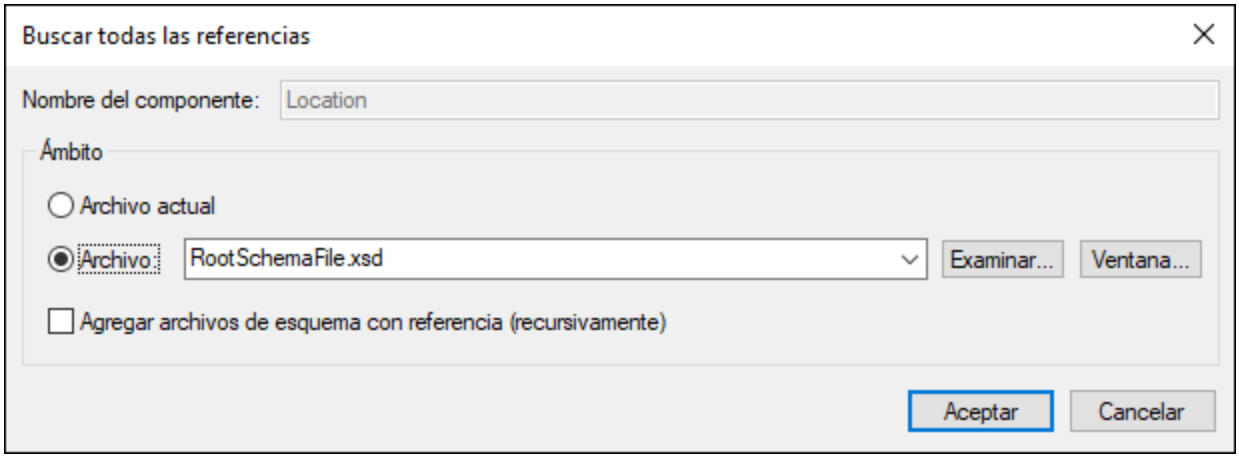
3. En el cuadro de diálogo que aparece seleccione el ámbito de la búsqueda o de la operación de cambio de nombre. En el caso de una operación de cambio de nombre, escriba el nombre nuevo del componente global.
4. Al hacer clic en **Aceptar**, los resultados de la búsqueda aparecen en la ventana Buscar en esquemas (*imagen siguiente*).



En los resultados también se indica la ubicación de los archivos en los que se encontraron referencias al componente global (*ver imagen anterior*). También se indica qué componentes se encontraron y se renombraron.

Buscar todas las referencias

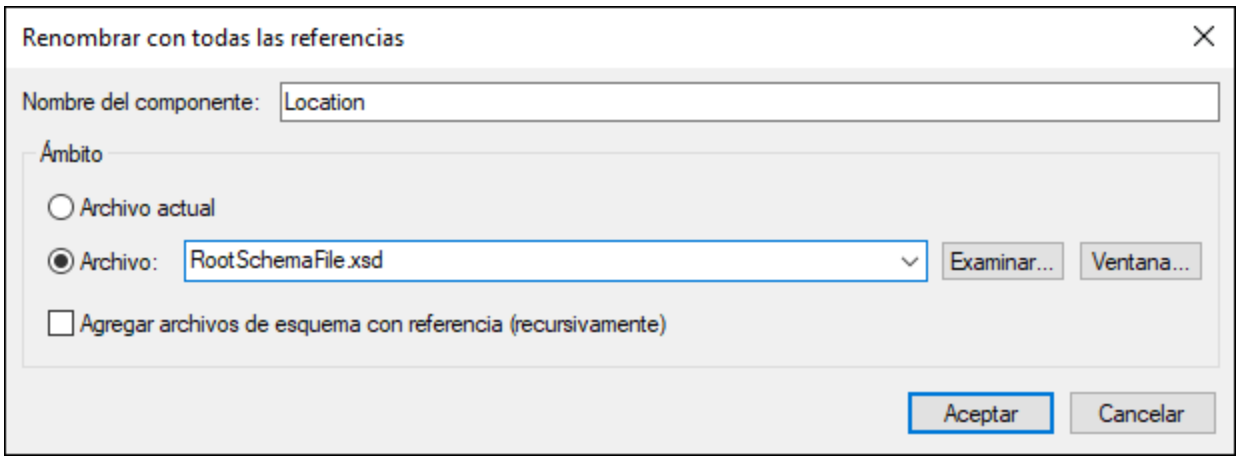
Para abrir el cuadro de diálogo "Buscar todas las referencias" (*imagen siguiente*) siga estos pasos: (i) haga clic con el botón derecho en el componente global pertinente en la vista general del esquema y (ii) seleccione el comando **Buscar todas las referencias** en el menú contextual que aparece.



El nombre del componente global aparece en el campo *Nombre del componente*, que está atenuado y no se puede editar. Puede elegir si la búsqueda se realiza en el archivo actual o en otro archivo (que puede buscar o seleccionar en la lista de archivos abiertos). También puede indicar si la búsqueda se debe realizar en los archivos relacionados (incluidos, importados y redefinidos). Para ello marque la casilla *Agregar archivos de esquema con referencia*.

Renombrar con todas las referencias

Para renombrar un componente global, haga clic con el botón derecho en el componente y en el menú contextual seleccione la opción **Renombrar con todas las referencias**. Aparece el cuadro de diálogo "Renombrar con todas las referencias" (*imagen siguiente*).



En el campo de texto *Nombre del componente* escriba el nombre nuevo que desea darle al componente global seleccionado. Puede elegir si la búsqueda y la operación de cambio de nombres se realiza en el archivo actual o en otro archivo (que puede buscar o seleccionar en la lista de archivos abiertos). También puede indicar si la

© 2016-2022 Altova GmbH

búsqueda y operación de cambio de nombre se debe realizar en los archivos relacionados (incluidos, importados y redefinidos). Para ello marque la casilla *Agregar archivos de esquema con referencia*.

7 XSLT

Sitio web de Altova: Editor de XSLT

Este apartado sobre XSLT se divide a su vez en varios subapartados:

- Edición de documentos XSLT 491: describe las funciones de edición que ofrece XMLSpy para documentos XSLT.
- Procesamiento XSLT 493: explica las diferentes maneras de realizar transformaciones XSLT en la interfaz gráfica de XMLSpy con ayuda de diferentes motores de transformación. Este subapartado incluye también información importante sobre la configuración de XSLT en XMLSpy.
- Esquema XSL 496: describe las ventanas Esquema XSL e Información XSL, que ofrecen ayuda a la hora de visualizar archivos XSLT, navegar por ellos y administrarlos.

Evaluación de XPath

Cuando está activo un documento XML, puede usar la ventana XPath/XQuery 127 para evaluar expresiones XPath. Esta función es muy práctica porque permite comprobar cómo se evaluará la expresión XPath. Escriba una expresión XPath y especifique si se debe evaluar como expresión relativa a la raíz del documento o a un nodo de contexto seleccionado en el documento. El resultado de la evaluación aparecerá inmediatamente en la ventana XPath. Para más información sobre la ventana XPath, consulte el apartado Introducción | Ventana de resultados: XPath/XQuery 127.

Depurador y generador de perfiles XSLT

XMLSpy también incluye un generador de perfiles XSLT 558 y un depurador XSLT 531 para ayudarle a crear hojas de estilos XSLT sin errores y eficaces con gran rapidez.

Funciones XSLT adicionales

Las descripciones de los diferentes comandos de menú 1392 (de la *Comandos de menú*) incluyen más información sobre las funciones descritas en esta sección.

Motores XSLT de Altova

Para saber cómo se implementan los motores XSLT 1.0, 2.0 y 3.0 de Altova, consulte la información sobre motores XSLT y XQuery 2114 en los anexos 2113 de este manual.

RaptorXML para procesamiento por lotes y desde la línea de comandos

La interfaz gráfica de XMLSpy permite procesar por lotes las carpetas de proyecto. Sin embargo, si necesita mayor flexibilidad, puede probar RaptorXML, un procesador de Altova que permite un rapidísimo procesamiento de datos (validación XML, transformaciones XSLT, ejecuciones de XQuery, etc.). RaptorXML es ideal si quiere realizar transformaciones XSLT desde la línea de comandos o por procesamiento por lotes.

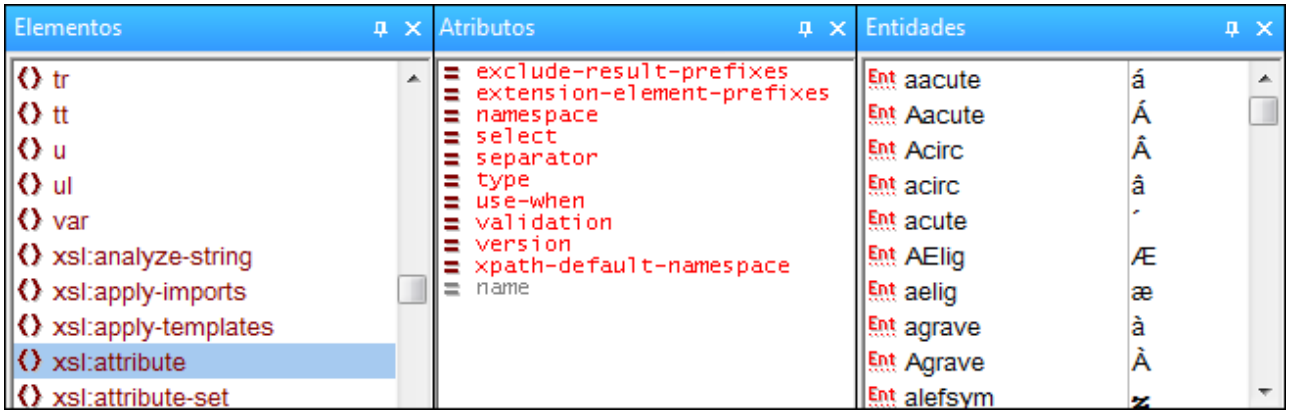
© 2016-2022 Altova GmbH

7.1 Documentos XSLT

Puede editar documentos XSLT 1.0, 2.0 y 3.0 en la vista Texto[145] y en la vista Cuadrícula[169], así como editarlos tal y como si se tratasen de documentos XML en la vista Texto[290] y en la vista Cuadrícula[169]. Puede configurar la vista predeterminada en la que se deben abrir los documentos XSLT en la sección *Tipos de archivo* del cuadro de diálogo "Opciones".

Ayudantes de entrada

Mientras trabaja con documentos XSLT, XMLSpy incluye ayudantes de entrada para elementos, atributos y entidades. La información que ofrecen los ayudantes de entrada sobre los diferentes componentes está integrada en XMLSpy y no depende de las referencias que contenga el documento XSLT.



Es necesario tener en cuenta estos aspectos:

1. Si se crea un documento XSLT nuevo desde el cuadro de diálogo "Crear un documento nuevo" (**Archivo | Nuevo**), los atributos y elementos XSLT correspondientes (XSLT 1.0, 2.0 o 3.0, dependiendo del tipo de documento elegido en el diálogo) se cargan automáticamente en los ayudantes de entrada. También se cargan elementos y atributos HTML, así como conjuntos de entidades Latin-1, caracteres especiales y símbolos HTML 4.0.
2. Si se crea un documento XML nuevo desde el cuadro de diálogo "Crear un documento nuevo" (**Archivo | Nuevo**) y se le da contenido XSLT, sólo estará disponible el ayudante de entrada Entidades, que incluirá caracteres XML.
3. Si se abre un documento XSLT que se creó como documento XSLT desde el cuadro de diálogo "Crear un documento nuevo" (**Archivo | Nuevo**), estarán disponibles los ayudantes de entrada descritos en el punto 1 anterior.
4. Si se abre un documento XSLT que **no** se creó como documento XSLT desde el cuadro de diálogo "Crear un documento nuevo" (**Archivo | Nuevo**), estarán disponibles los ayudantes de entrada descritos en el punto 1 anterior. Además, los ayudantes de entrada de la vista Texto incluirán elementos y atributos XSL:FO.
5. Los prefijos de los elementos del ayudante de entrada Elementos son siempre estos: `xsl:` para los elementos XSLT (los elementos HTML no llevan prefijo) y `fo:` para los elementos XSL:FO. Como consecuencia, para poder usar los ayudantes de entrada, las declaraciones de espacios de nombres del documento XSLT deben definir prefijos que coincidan con los prefijos integrados que aparecen en los ayudantes de entrada.

Finalización automática

En la vista Texto la función de finalización automática aparece mientras se teclea en forma de lista emergente. La primera opción de la lista emergente que coincida con el texto tecleado aparece resaltada. Cuando lo que se teclea es un elemento, en la lista emergente los elementos están ordenados por orden alfabético y se resalta el primer elemento que coincida con el texto tecleado. Lo mismo ocurre con los atributos. Las opciones de la lista emergente vienen determinadas por las reglas que afectan a los ayudantes de entrada (descritas más arriba).

Edición inteligente de expresiones XPath

XMLSpy ofrece funciones de edición inteligente de XPath en las partes del documento XSLT donde se pueden introducir expresiones XPath (por ejemplo, dentro del valor de un atributo `select`):

- Color de sintaxis para las construcciones XPath (incluida una función para resaltar los pares de etiquetas) mientras se teclea.
- Información rápida que aparece cuando se pasa el cursor por encima de una función XPath.
- Sugerencias contextuales sobre funciones y ejes XPath mientras se teclea. Puede avanzar y retroceder en la lista de sugerencias con las teclas de fecha **Arriba** y **Abajo**. Si la sugerencia contextual hace referencia a una función, pase el cursor sobre ella para obtener más información sobre ella.
- Si se asignó un archivo XML en la ventana Información 500, los elementos y atributos de este archivo XML también aparecerán en las sugerencias contextuales.

Validación de documentos XSLT

Los documentos XSLT se pueden validar con un esquema XSLT que está integrado en XMLSpy (clic en **XML | Validar (F8)**). El esquema integrado apropiado se selecciona automáticamente para la validación, dependiendo de si el documento XSLT es XSLT 1.0, 2.0 o 3.0 (lo cual se indica en el atributo `version` del elemento `xsl:stylesheet`).

© 2016-2022 Altova GmbH

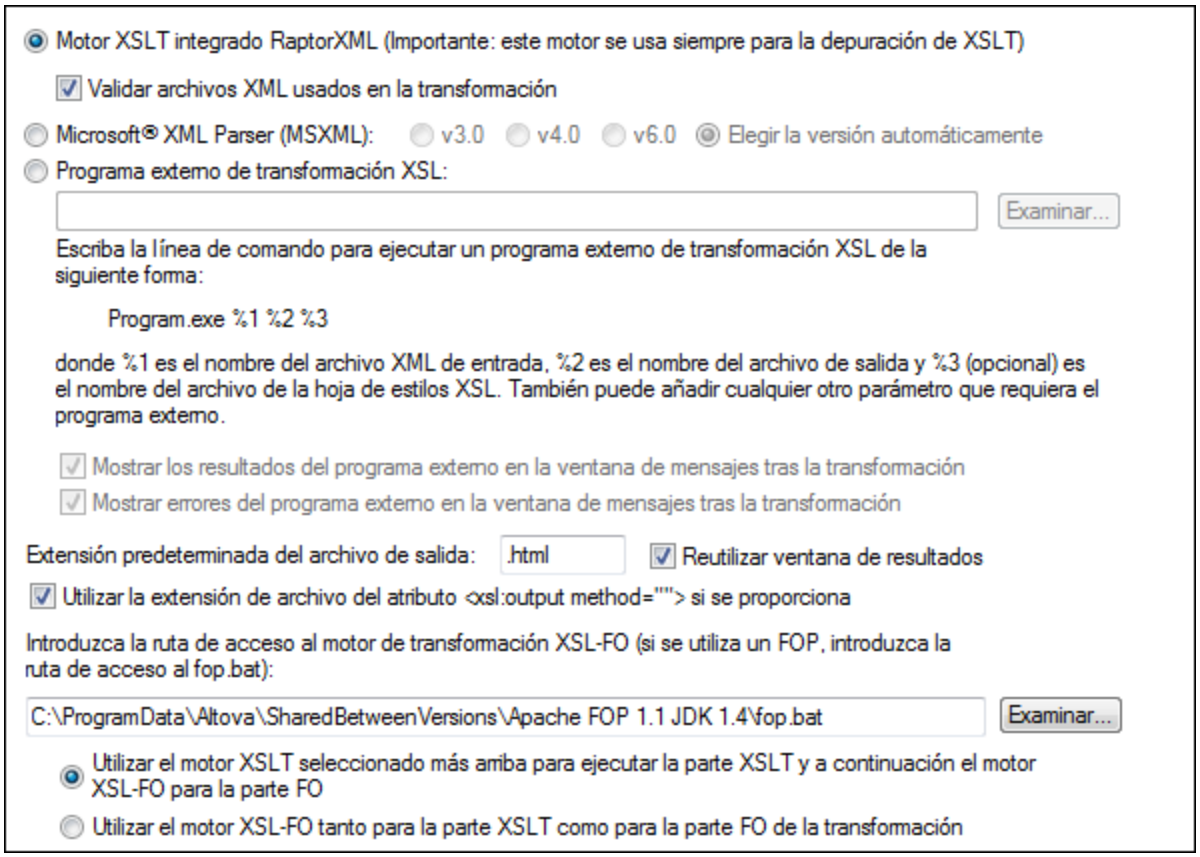
7.2 Procesamiento XSLT

En la interfaz gráfica de XMLSpy hay dos tipos de transformaciones XSLT:

- El comando **XSL/XQuery | Transformación XSL (F10)** se utiliza para realizar una transformación sencilla con una hoja de estilos XSLT en los formatos de salida indicados y descritos en la hoja de estilos.
- El comando **XSL/XQuery | Transformación XSL-FO** se utiliza para: (i) transformar XML en FO y después en PDF en un proceso de dos pasos y (ii) para transformar FO en PDF en un solo paso.

Indicar el procesador XSLT para la transformación

Es necesario especificar el motor XSLT que se usa para la transformación en la sección XSL del cuadro de diálogo "Opciones" 1612 (*imagen siguiente*).



Las opciones de esta pestaña se explican en detalle en la *Comandos de menú* 1612. El motor de transformación elegido en la sección *XSL* se usará para todas las transformaciones XSLT. Recuerde que para las transformaciones XSL:FO, hay una opción más: el motor XSLT que viene con algunos procesadores FO. Para seleccionar esta opción, seleccione el botón de opción correspondiente al final de la sección *XSL* (*ver imagen anterior*).

Indicar el procesador FO

En el cuadro de texto situado al final de la sección XSL del cuadro de diálogo "Opciones"[1612] (*imagen anterior*) puede indicar el procesador FO que se debe utilizar para transformar FO en PDF.

Motores XSLT 1.0, 2.0 y 3.0 de Altova

La versión XSLT de una hoja de estilos se especifica en el atributo `version` del elemento `xsl:stylesheet` (o `xsl:transform`). XMLSpy contiene los motores integrados XSLT 1.0, 2.0 y 3.0 de Altova y, dependiendo del valor del atributo `version` (`1.0`, `2.0` ó `3.0`), se elige un motor u otro para la transformación.

Transformación XSLT

El comando **Transformación XSLT (F8)** se puede usar en los casos descritos a continuación:

- Para transformar un documento XML que está activo en XMLSpy y tiene asignado[1403] un documento XSLT. Si no tiene asignado un documento XSLT, se le pedirá que le asigne uno nada más hacer clic en el comando **Transformación XSLT (F8)**.
- Para transformar un documento XSLT que está activo en XMLSpy. Al hacer clic en el comando **Transformación XSLT (F8)**, se le pedirá el archivo XML que desea procesar con la hoja de estilos XSLT activa.
- Para transformar carpetas y archivos de un proyecto. Haga clic con el botón derecho en la carpeta o archivos del proyecto y seleccione el comando **Transformación XSL**.

Asignación inversa

Si se habilita la característica de asignación inversa[1401], las transformaciones XSLT se llevan a cabo de modo que el documento de salida se pueda relacionar con los documentos XSLT+XML originales. Esto significa que, si se hace clic en un nodo del documento de salida, la aplicación resaltará la **instrucción XSLT** y los **datos de origen XML** que produjeron dicho nodo. Además, si se hace clic en una instrucción XSLT o en un nodo de datos XML, la aplicación resaltará los nodos correspondientes en los otros dos documentos. Consulte la descripción del comando **XSL/XQuery | Habilitar asignación inversa**[1401] para obtener más información.

Transformación XSL:FO

El comando **Transformación XSL:FO** se puede usar en los casos descritos a continuación:

- Para transformar un documento XML que está activo en XMLSpy y tiene asignado[1403] un documento XSLT. El documento XML se transforma primero en FO con el motor XSLT indicado. Después se procesa el documento FO con el procesador FO indicado y se genera un documento PDF de salida. Si el documento XML no tiene un XSLT asignado, se le pedirá que le asigne uno nada más hacer clic en el comando **Transformación XSL:FO**.
- Para transformar un documento FO en PDF usando el procesador indicado.
- Para transformar un documento XSLT que está activo en XMLSpy. Al hacer clic en el comando **Transformación XSL:FO**, se le pedirá el archivo XML que desea procesar con la hoja de estilos XSLT activa.
- Para transformar carpetas y archivos de un proyecto. Haga clic con el botón derecho en la carpeta o archivos del proyecto y seleccione el comando **Transformación XSL:FO**.

Para más información sobre las opciones del cuadro de diálogo Resultado de la transformación XSL:FO[1395], consulte la *Comandos de menú*[1395].

© 2016-2022 Altova GmbH

Parámetros para XSLT

Si usa los motores XSLT de Altova, puede almacenar los parámetros de XSLT en un práctico cuadro de diálogo. Todos los parámetros almacenados se pasan al documento XSLT cada vez que se realice una transformación. Para más información, consulte la descripción del comando Parámetros de XSL / Variables de XQuery 1396.

Procesamiento por lotes con RaptorXML

RaptorXML es una aplicación independiente que incluye el procesador más reciente de Altova, con validador XML, motores XSLT y motor XQuery. Se puede usar desde la línea de comandos, a través de la interfaz de COM, en programas Java y en aplicaciones .NET para validar documentos XML, transformar documentos XML usando hojas de estilos XSLT y ejecutar documentos XQuery.

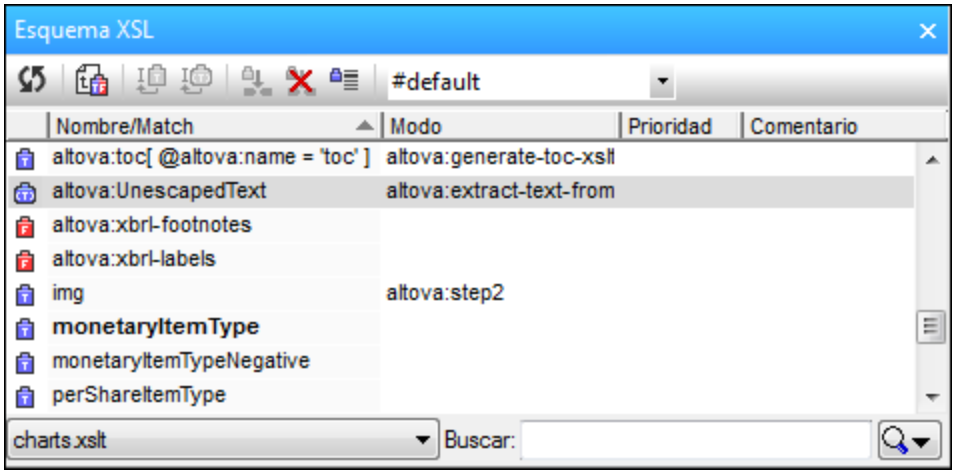
Por tanto, con RaptorXML puede automatizar las tareas de transformación XSLT. Por ejemplo, puede crear un archivo por lotes que llame a RaptorXML para que transforme un conjunto de documentos. Para más información, consulte la documentación de RaptorXML.

7.3 Esquema XSL

Cuando el documento activo en XMLSpy es un documento XSLT, la ventana Esquema XSL[497] muestra información sobre la estructura del documento y la pestaña *XSLT*[500] de la ventana Información[500] (que se habilita cuando el documento activo en XMLSpy es un documento XSLT) muestra información sobre los archivos relacionados con el documento XSLT activo. Además, estas dos ventanas ofrecen varios comandos que le ayudarán a editar el documento XSLT y a gestionar los archivos relacionados con el documento.

En la **ventana Esquema XSL**[497] (*imagen siguiente*), puede realizar estas operaciones:

- Visualizar las plantillas y funciones del documento XSLT activo y de todos los documentos XSLT importados e incluidos.
- Ordenar las plantillas y funciones según su nombre o expresión del atributo `match`, modo, prioridad o comentarios.
- Buscar determinadas plantillas según su nombre o expresión del atributo `match`.
- Usar el esquema XSL para navegar hasta la plantilla correspondiente en el documento XSLT.
- Insertar rápidamente llamadas a plantillas con nombre.
- Fijar una plantilla con nombre como punto de entrada para las transformaciones.

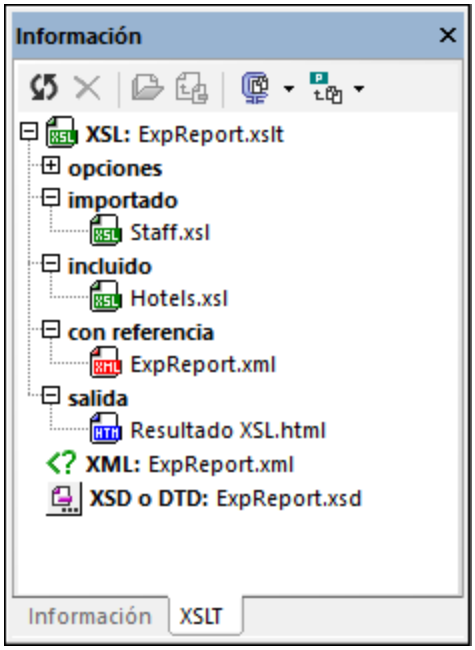


Consulte el apartado *Ventana Esquema XSL*[497] para obtener más información.

En la **pestaña XSLT de la ventana Información**[500] (*imagen siguiente*), puede realizar estas operaciones:

- Ver información sobre todos los archivos relacionados con el documento XSLT activo, como la ubicación de los archivos importados e incluidos.
- Indicar qué archivo XML se debe transformar con el documento XSLT activo. También puede indicar el archivo de esquema (XSD o DTD) que se debe usar para validar el archivo XML seleccionado.
- Abrir un archivo relacionado desde la ventana Información.
- Organizar rápidamente todos los archivos relacionados en proyectos de XMLSpy.
- Comprimir todos los archivos relacionados en una ubicación definida por el usuario.

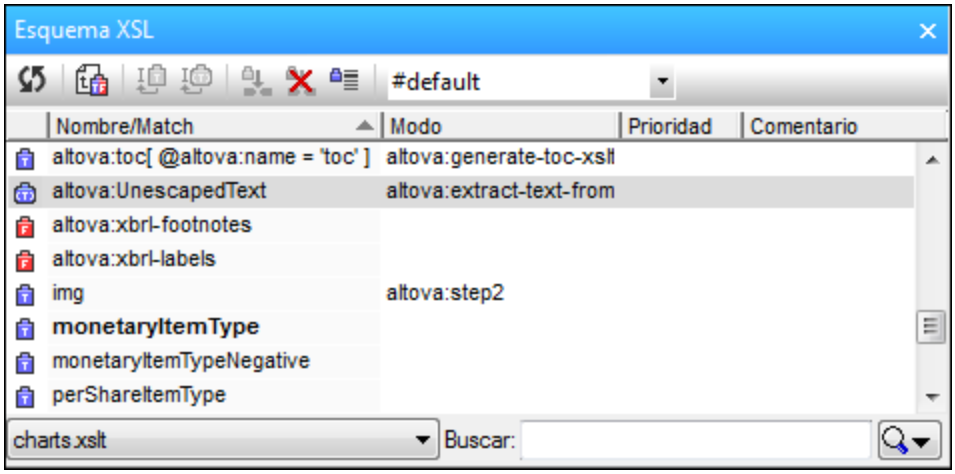
© 2016-2022 Altova GmbH



La ventana Esquema XSL 497 y la pestaña XSLT de la ventana Información 500 se describen en detalle en los subapartados siguientes.

7.3.1 Ventana Esquema XSL

En la ventana Esquema XSL (*imagen siguiente*) aparecen todas las plantillas y funciones del documento XSLT activo. Las plantillas se identifican con iconos azules (para plantillas que no tienen un parámetro y para las plantillas que tienen parámetros). Las funciones se identifican con este icono rojo . En el cuadro combinado situado en la parte inferior izquierda de la ventana, puede elegir si se muestran las funciones y plantillas (i) del documento XSLT activo solamente (como en la imagen siguiente) o (ii) del documento XSLT activo y de todas las hojas de estilo incluidas e importadas.



Hay dos tipos de plantillas: (i) las plantillas con nombre y (ii) las plantillas que coinciden con una expresión XPath. Cada plantilla de la lista aparece:

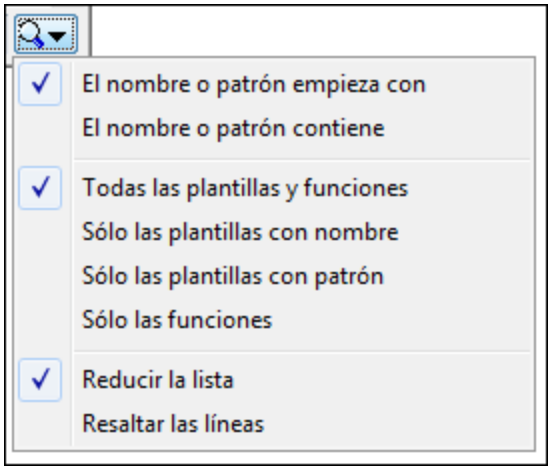
- Con su nombre (si la plantilla tiene un atributo `name`) y/o su expresión XPath (si la plantilla tiene el atributo `match`). Si la plantilla tiene ambos atributos (es decir, un atributo `name` y un atributo `match`), entonces aparecen los dos: primero el valor del atributo `name` y después el valor del atributo `match` (p. ej. la plantilla `UnescapedText` de la imagen anterior)
- Con su modo, si lo tuviese. Recuerde que una plantilla puede tener más de un modo
- Con su prioridad, si la tuviese.
- Con el comentario que precede a la plantilla o función, si lo hubiese.

Las funciones de la hoja de estilos sólo se presentan con su nombre. Las funciones no tienen modo ni prioridad.

Operaciones

En la ventana Esquema XSL puede realizar estas operaciones:

- *Filtrar:* la lista de la ventana se puede filtrar para que muestre: (i) todas las plantillas y funciones (filtro predeterminado cada vez que se abre XMLSpy); (ii) sólo las plantillas con nombre; (iii) sólo las plantillas que tienen expresiones XPath; o (iv) sólo funciones. Para seleccionar uno de estos filtros, haga clic en la punta de flecha situada a la derecha del cuadro Buscar (*ver imagen siguiente*) y seleccione el filtro que desea utilizar (en el segundo grupo de comandos del menú desplegable). El filtro seleccionado se aplica inmediatamente y no deja de aplicarse hasta que seleccione otro filtro o cierre XMLSpy.



- *Ordenar y localizar:* cada columna se puede ordenar alfabéticamente con solo hacer clic en el encabezado de la columna. Cada vez que se hace clic en el encabezado de la columna, se invierte el orden alfabético. Después de ordenar una columna alfabéticamente, puede seleccionar un elemento de la lista, teclear un término y se resaltará el primer elemento de la lista que contenga el término. Así puede ir rápidamente a la plantilla que tenga un nombre o expresión, modo o prioridad determinados.
- *Buscar:* escriba el nombre o expresión XPath que desea buscar en el cuadro Buscar (situado en la parte inferior derecha). Los resultados de la búsqueda aparecen mientras se teclea el término de búsqueda. El cuadro de búsqueda también incluye algunas opciones de búsqueda (*ver imagen anterior*): (i) buscar expresiones o nombres que empiecen con el término de búsqueda (opción predeterminada) o que contengan el término de búsqueda (el primer grupo de comandos del menú desplegable) y (ii) mostrar sólo los resultados de la búsqueda (opción predeterminada) o resaltar los

© 2016-2022 Altova GmbH

resultados en la lista de plantillas y funciones (el tercer grupo de comandos del menú desplegable). Las opciones seleccionadas se aplican inmediatamente y no dejan de aplicarse hasta que seleccione otra opción o cierre XMLSpy.

- *Volver a cargar:* después de realizar cambios en la hoja de estilos, haga clic en el icono **Sincronizar** de la barra de herramientas de la ventana y se actualizará el esquema XSL.
- *Ir a un elemento:* si selecciona una plantilla o función en la ventana Esquema XSL y hace clic en el icono **Ir a la definición** de la barra de herramientas de la ventana, la plantilla o función se resaltará en la ventana principal. Otra manera de hacer esto es hacer doble clic en el elemento.
- *Acciones relacionadas con plantillas con nombre:* puede llevar a cabo varias acciones relacionadas con plantillas con nombre: (i) puede insertar llamadas a la plantilla con nombre (con `xsl:call-template`) en la hoja de estilos en el punto de inserción del cursor y (ii) puede fijar una plantilla con nombre como punto de entrada para una transformación. Los comandos para estas acciones se ejecutan a través de los iconos de la barra de herramientas y se describen a continuación.

Modo de plantilla para la transformación

En la barra de herramientas, el cuadro combinado llamado *Configurar modo para la transformación* enumera (i) todos los modos de la hoja de estilos; (ii) una entrada vacía (que selecciona el modo predeterminado); y (iii) si se trata de una hoja de estilos XSLT 3.0, el modo `#sin nombre`. Seleccione un modo de la lista para definirlo como modo de la transformación. El modo `#sin nombre` (para todas las versiones de XSLT) afecta a las plantillas que no tienen el atributo `mode`.

Si se trata de una hoja de estilos XSLT 1.0 o XSLT2.0, el modo predeterminado es el modo `#sin nombre`. De modo que, si selecciona la entrada vacía, se selecciona el modo predeterminado `#sin nombre`, que afecta a todas las plantillas que no tienen el atributo `mode`.

En las hojas de estilos XSLT 3.0, el elemento de nivel superior `xslt` puede tener el atributo `default-mode`, que almacena el modo predeterminado de la transformación. Si selecciona la entrada vacía (modo predeterminado) en la lista del cuadro combinado *Configurar modo para la transformación,* se usará el modo del atributo `default-mode` como modo de transformación. Si selecciona `#sin nombre` en la lista del cuadro combinado, la transformación se aplicará a todas las plantillas que no tengan el atributo `mode`.

Nota: una plantilla puede recibir el valor de modo `#all` para que sea aplicable a todos los modos.

Plantillas con nombre

Al seleccionar una plantilla con nombre, se habilitan algunos comandos de la barra de herramientas de la ventana (*ver imagen siguiente*) relacionados con las plantillas con nombre.



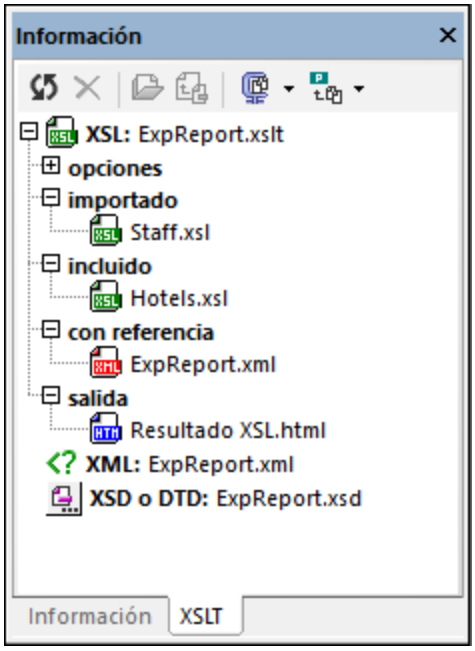
Estos comandos son (*de izquierda a derecha en la imagen anterior*):

- *Insertar xsl:call-template:* este comando se habilita al seleccionar una plantilla con nombre en la ventana Esquema XSL. El comando inserta un elemento `xsl:call-template` en el punto de inserción del cursor en la hoja de estilos. El atributo `name` del elemento `xsl:call-template` que se inserta en la hoja de estilos recibe el valor del atributo `name` de la plantilla con nombre seleccionada en la ventana Esquema XSL. De este modo el elemento `xsl:call-template` realiza una llamada a la plantilla con nombre seleccionada.

- *Insertar xsl:call-template with param:* este comando se habilita al seleccionar en la ventana Esquema XSL una plantilla con nombre que tenga elementos secundarios `xsl:param`. Como en el caso del comando **Insertar xsl:call-template**, el comando inserta un elemento `xsl:call-template`, pero en este caso con un elemento secundario `xsl:with-param` por cada elemento secundario `xsl:param` de la plantilla con nombre seleccionada en la ventana Esquema XSL. El nombre del elemento `xsl:call-template` insertado y de sus elementos secundarios `xsl:with-param` será el mismo que el de la plantilla con nombre seleccionada y de sus secundarios `xsl:param` respectivamente.
- *Establecer la plantilla con nombre seleccionada como punto de entrada para la transformación*: cuando se establece una plantilla con nombre como el punto de entrada para una transformación, las transformaciones ejecutadas en XMLSpy empiezan por esta plantilla con nombre. En la ventana Esquema XSL dicha plantilla con nombre aparece en negrita (*ver la primera imagen de esta sección*).
- *Eliminar la plantilla con nombre seleccionada como punto de entrada para la transformación*: se habilita después de establecer una plantilla con nombre como punto de entrada para las transformaciones.
- *Saltar a la plantilla con nombre seleccionada como punto de entrada para la transformación*: se habilita después de establecer una plantilla con nombre como punto de entrada para las transformaciones. Cuando en la ventana Esquema XSL está seleccionado otro elemento, al hacer clic en este icono se resalta la plantilla con nombre seleccionada como punto de entrada para las transformaciones.

7.3.2 Ventana Información

La pestaña XSLT de la ventana Información (que se habilita cuando el documento activo en XMLSpy es un documento XSLT) muestra todos los archivos XSLT importados e incluidos en el documento XSLT activo y además permite seleccionar un archivo XML como documento XML de origen para la transformación XSLT.

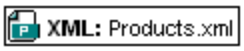


La pestaña XSLT de la ventana Información muestra estos archivos:

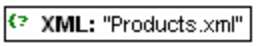
© 2016-2022 Altova GmbH

- *Archivos XSLT:* todos los archivos XSLT importados e incluidos (*ver imagen anterior*). La ubicación de cada archivo aparece en un mensaje emergente al pasar el puntero del ratón sobre el nombre del archivo XSLT. Al hacer doble clic en un archivo importado o incluido, el archivo se abre en una ventana nueva (otra opción es seleccionar el archivo y hacer clic en el icono **Abrir** de la barra de herramientas de la ventana). El icono **Ir a ubicación de la inclusión/importación** de la barra de herramientas resalta la declaración de inclusión/importación en el documento XSLT activo.

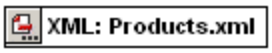
- *Archivos XML:* puede asignar un archivo XML a la hoja de estilos XSLT activa. La ubicación del archivo XML asignado aparece en un mensaje emergente al pasar el puntero del ratón sobre el nombre del archivo XML. Si se asigna un archivo XML y hace clic en el comando **XSL/XQuery | Transformación XSL (F10)**, se ejecuta una transformación en el archivo XML definido usando el documento XSLT activo como hoja de estilos. Para asignar un archivo XML haga clic en el icono **XML** y busque el archivo que desea asignar. El archivo seleccionado aparece en negrita. Otra opción es asignar el archivo XML desde el cuadro de diálogo "Propiedades de proyecto" (*XML de entrada para la transformación XSL/XQuery/XQuery Update*) o insertando una instrucción de procesamiento en el documento XSLT: `<?altova_samplexml "Products.xml"?>`. En cualquier caso, el archivo XML aparecerá en la ventana Información con el icono que corresponda:

  archivo asignado desde el cuadro de diálogo "Propiedades de proyecto"

  archivo asignado con una instrucción de procesamiento del documento XSLT

  archivo asignado haciendo clic en el icono XML y buscando el archivo (el nombre del archivo aparece en negrita)

  Si se realizara más de una asignación, el orden de prioridad es: (i) propiedades de proyecto, (ii) instrucción de procesamiento y (iii) archivo seleccionado con el icono **XML**. Para abrir el archivo XML, haga doble clic en el nombre del archivo o selecciónelo y haga clic en el icono **Abrir** de la barra de herramientas.

- *Archivo XSD/DTD:* si el archivo XML asignado tiene una referencia a un esquema (esquema XML o DTD), este archivo de esquema aparece también en la ventana Información (entrada **XSD/DTD**). El archivo de esquema también se puede seleccionar desde el cuadro de diálogo "Propiedades de proyecto" (*Validación*) o haciendo clic en el icono **XSD/DTD** y buscando el esquema. Si el archivo de esquema se selecciona desde el cuadro de diálogo "Propiedades de proyecto", junto al nombre del esquema aparece el icono de proyecto. Si el esquema está asociado al archivo XML, junto al nombre del esquema aparece el icono **XSD/DTD**. Si el usuario usa el icono **XSD/DTD** para seleccionar el esquema, el nombre del esquema aparece en negrita. Si se realizara más de una asignación, el orden de prioridad es: (i) propiedades de proyecto, (ii) archivo seleccionado con el icono **XSD/DTD** y (iii) referencia al esquema en el archivo XML. La ubicación del archivo de esquema aparece en un mensaje emergente al pasar el puntero sobre el nombre del archivo. Para abrir el archivo de esquema, haga doble clic en el nombre del archivo o selecciónelo y haga clic en el icono **Abrir** de la barra de herramientas de la ventana.

  Nota: si selecciona un archivo XML o de esquema (XSD/DTD) desde el cuadro de diálogo "Propiedades de proyecto" y desea borrar esta selección, vuelva al cuadro de diálogo y anule la configuración. Si el archivo se selecciona con los iconos **XML** o **XSD/DTD** y desea anular la selección, primero seleccione el archivo en la ventana Información y después haga clic en el icono **Borrar** de la barra de herramientas de la ventana.

Opciones

Edición inteligente de XPath: si se asigna un archivo XML, entonces XMLSpy conoce la estructura del archivo XML y la función de edición inteligente de XPath[491] se habilita para los elementos y atributos. Allá donde se pueda insertar una expresión XPath en el documento XSLT, aparece una lista emergente con los elementos y atributos disponibles. Esta opción está activada por defecto. Para deshabilitarla, desactive la casilla de verificación. Al cerrar el archivo XSLT la opción de configuración se guarda también.

Iconos de la barra de herramientas

Estos son los iconos de la barra de herramientas (*de izquierda a derecha en la imagen siguiente*):



- *Volver a cargar la información:* la ventana Información se actualiza para reflejar los cambios realizados en el documento XSLT.
- *Borrar la asignación de XML/XSD*: se borra la asignación del archivo XML o XSD/DTD realizada con el icono **XML** o **XSD/DTD** respectivamente. Seleccione el archivo y después haga clic en este icono.
- *Abrir documento*: abre el documento seleccionado.
- *Ir a ubicación de la importación/inclusión:* si selecciona un archivo importado o incluido y hace clic en este icono, la declaración de importación o inclusión se resalta en el documento XSLT.
- *Comprimir todos los documentos como archivo ZIP:* los documentos que aparecen en la ventana Información se comprimen y se guardan en un archivo ZIP en una ubicación definida por el usuario. También puede comprimir sólo los documentos seleccionados. Para ello haga clic en el comando **Comprimir los documentos seleccionados** en el menú desplegable de este icono.
- *Agregar todos los archivos al proyecto:* agrega todos los archivos al proyecto actual. También puede agregar al proyecto sólo los documentos seleccionados. Para ello haga clic en el comando **Agregar archivos seleccionados al proyecto** en el menú desplegable de este icono.

© 2016-2022 Altova GmbH

7.4 Optimizador de velocidad XSL

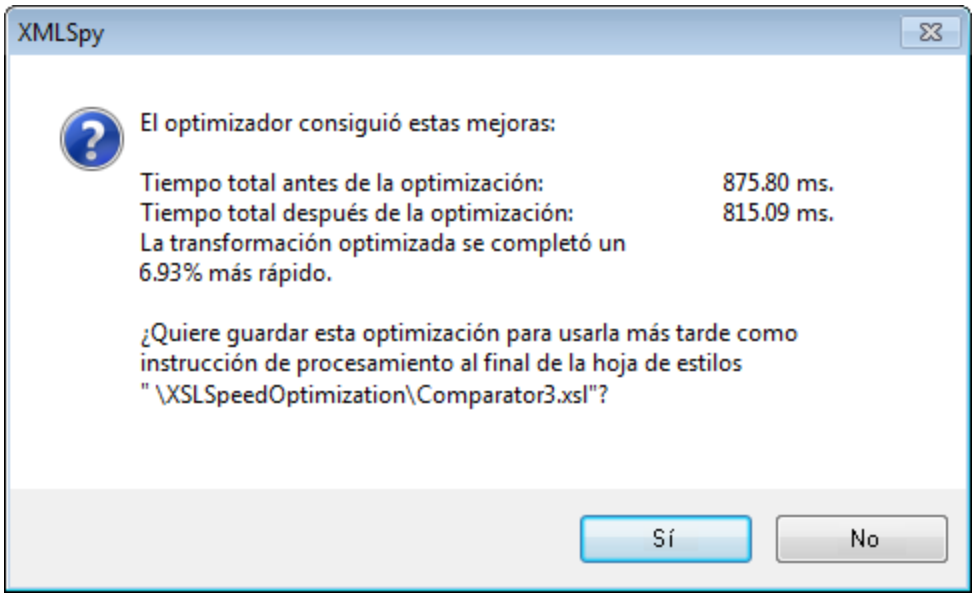
El optimizador de velocidad XSL sirve para optimizar hojas de estilos XSLT y que las transformaciones se lleven a cabo más rápido. Así es como funciona: el optimizador ejecuta la hoja de estilos XSLT que se debe optimizar sobre un documento XML (es decir, sobre documentos XML) y analiza su rendimiento. De este análisis se deriva una estrategia de optimización que se puede guardar con la hoja de estilos XSLT (en forma de instrucción de procesamiento situada al final de la hoja de estilos). Después puede usar la hoja de estilos optimizada para realizar transformaciones más rápido.

Optimizar una hoja de estilos XSLT

Para poder optimizar una hoja de estilos es necesario suministrar un documento XML que sirva de muestra. El documento debe ser lo suficientemente grande como para que se utilicen todas las partes de la hoja de estilos XSLT y se analice toda la hoja de estilos correctamente.

Siga estas instrucciones para optimizar la hoja de estilos XSLT:

1. Abra la hoja de estilos XSLT o el documento XML en XMLSpy y haga clic en el comando de menú **XSL/XQuery | Optimizador de velocidad XSL**[1394] (o haga clic en el botón del optimizador en la barra de herramientas).
2. Aparece un cuadro de diálogo donde debe seleccionar un documento XML o una hoja de estilos XSLT (dependiendo de qué documento esté activo en la vista). Elija el archivo y haga clic en **Aceptar** para iniciar el análisis. (Si la hoja de estilos XSLT o el documento XML ya tiene asignado un documento XML[1404] o un documento XSLT[1403], respectivamente, este paso se pasa por alto y el análisis comienza en cuanto se invoca el comando.)
3. Si el análisis de optimización no consigue derivar ninguna estrategia de optimización, se emite un mensaje a tal efecto. (Más abajo se explican los posibles motivos[504] por los que no se puede derivar la estrategia de optimización.) Si el análisis finaliza correctamente, aparece un mensaje confirmando el resultado del análisis (imagen siguiente).



El aviso anterior ofrece la opción de guardar la optimización (instrucciones) en la hoja de estilos XSLT (en forma de instrucción de procesamiento al final de la hoja de estilos). Haga clic en **Sí** para guardar la

optimización y **No** para descartarla. Cuando se guarda una optimización, se reemplaza la optimización guardada con anterioridad.

Ahora puede usar la hoja de estilos optimizada para realizar transformaciones más rápido.

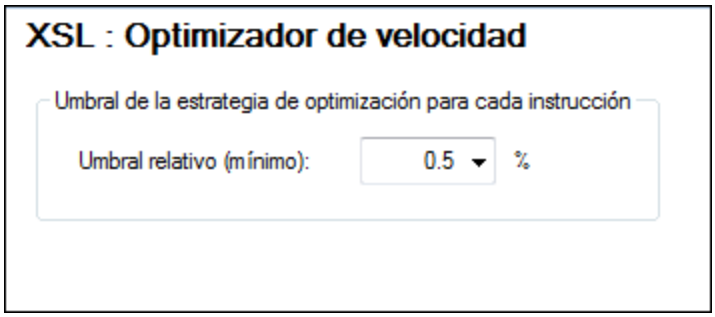
Motivos por los que no se puede derivar la estrategia de optimización

Si el optimizador no puede derivar una estrategia de optimización, puede haber varios motivos:

- La hoja de estilos XSLT ya es eficaz y no necesita ser optimizada.
- El documento XML suministrado es demasiado pequeño. Vuelva a intentarlo con un documento más grande.
- Puede que el umbral para la optimización sea demasiado alto. Cambie el umbral en en la sección *XSL: Optimizador de velocidad*[1616] del cuadro de diálogo "Opciones"[1616] (*ver más abajo*).
- La optimización necesaria para la estructura XSLT no está disponible en el optimizador. En este caso puede ponerse en contacto con el equipo de soporte técnico de Altova.

Configurar el optimizador de velocidad XSL

El optimizador se puede configurar en la sección *XSL: Optimizador de velocidad*[1616] del cuadro de diálogo "Opciones"[1616] (**Herramientas | Opciones**).



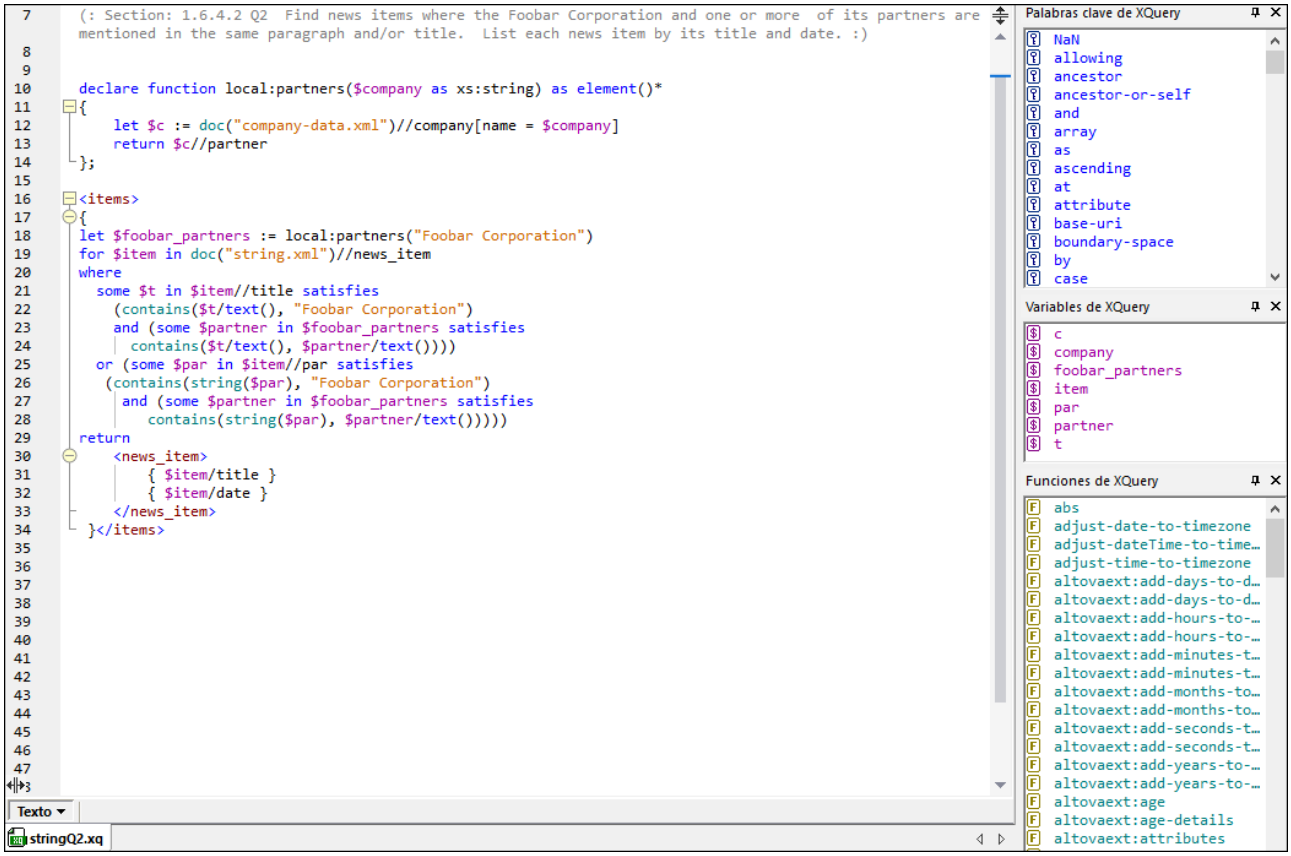
En esta pestaña puede especificar el umbral del optimizador para cada instrucción XSLT de una hoja de estilos XSLT. Puede elegir valores comprendidos entre un 0.1% de la duración total de la transformación y un 99%. Si una instrucción tarda en ejecutarse más tiempo del indicado como umbral, entonces se invoca el análisis de optimización. De lo contrario, en análisis no se lleva a cabo. Si el análisis de optimización no puede derivar una estrategia de optimización, puede que el umbral indicado en las opciones de configuración sea demasiado alto. Baje el umbral y vuelva a intentarlo.

© 2016-2022 Altova GmbH

8 XQuery

Sitio web de Altova: Editor de XQuery

Los documentos XQuery y XQuery Update 519 se editan en la vista Texto (*imagen siguiente*), que incluye ayudantes de entrada y funciones de color de sintaxis y edición inteligente para facilitar la edición. En la vista Texto también puede validar documentos XQuery y ejecutarlos (con un archivo XML opcional si lo desea) con el motor integrado XQuery de Altova.



Nota: los archivos XQuery y XQuery Update sólo se pueden editar en la vista Texto. Esta es la única vista disponible para los archivos XQuery.

Asociaciones de archivos XQuery y XQuery Update

XMLSpy distingue entre documentos XQuery y XQuery Update. Los documentos XQuery suelen tener la extensión `.xq`, mientras que los documentos XQuery Update tienen la extensión `.xqu`. Sin embargo, puede asociar más extensiones de archivo a estos tipos de archivo y cambiar las asociaciones de tipo de archivo en la sección *Tipos de archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones | Tipos de archivo** 1587).

La asociación de tipo de documento de una extensión de archivo es muy importante porque, dependiendo de esta asociación, XMLSpy ejecutará XQuery o XQuery Update 519 cuando el usuario seleccione el comando **Ejecutar XQuery/XQuery Update**.

Contenido de esta sección

Esta sección se divide en estos apartados:

- Editar documentos XQuery 507
- Evaluar expresiones XQuery 514
- Validar documentos XQuery 515
- Ejecutar XQuery/XQuery Update 516
- XQuery Update Facility 519
- XQuery y bases de datos XML 526

Características e información relacionada:

- Depurador y generador de perfiles XSLT/XQuery 530
- Implementación del motor XQuery 2114
- Ventana de resultados XPath/XQuery 127
- Herramientas | Opciones | Tipos de archivo 1587
- Herramientas | Opciones | XQuery 1616

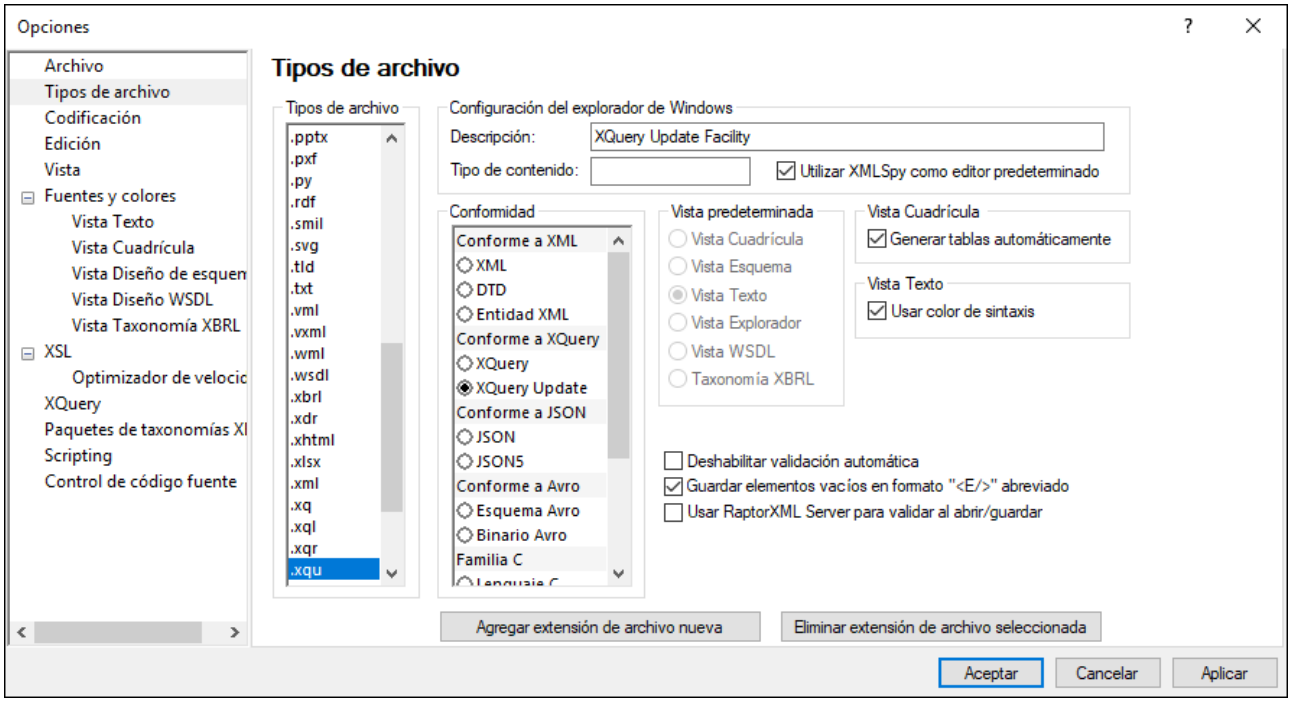
RaptorXML para procesamiento por lotes y desde la línea de comandos

La interfaz gráfica de XMLSpy permite procesar por lotes las carpetas de proyecto. Sin embargo, si necesita mayor flexibilidad, puede probar RaptorXML, el procesador más reciente de Altova que incluye un motor XQuery. RaptorXML es la herramienta ideal para ejecutar documentos XQuery desde la línea de comandos o para procesamiento por lotes.

© 2016-2022 Altova GmbH

8.1 Editar documentos XQuery

XMLSpy distingue entre documentos XQuery y XQuery Update. El tipo de documento (XQuery o XQuery Update) se asigna a una extensión de archivo determinada en la sección *Tipos de archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones | Tipos de archivo**[1587]). Por tanto, cuando se abre un archivo de tipo XQuery o XQuery Update en XMLSpy, se habilitan las funciones de edición XQuery de la vista Texto.



Extensiones de archivo asociadas a XQuery y XQuery Update en XMLSpy

XQuery	`.xq` `.xql` `.xqr` `.xquery`
XQuery Update	`.xqu`

Nota: las funciones de edición de esta sección son idénticas para los documentos XQuery y los documentos XQuery Update.

Ejecutar XQuery/XQuery Update

El comando **XSL/XQuery | XQuery/**Ejecutar XQuery/XQuery Update se ejecuta automáticamente código XQuery o XQuery Update, dependiendo del tipo de archivo XQuey que se seleccione. Para más información consulte el apartado *Ejecutar XQuery/XQuery Update*[516].

8.1.1 Documentos XQuery

Los documentos XQuery y XQuery Update se abren automáticamente en el modo XQuery de la vista Texto si son conformes a XQuery y XQuery Update respectivamente. Las asociaciones de archivo se definen en la

sección *Tipos de archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones | Tipos de archivo** 1587).

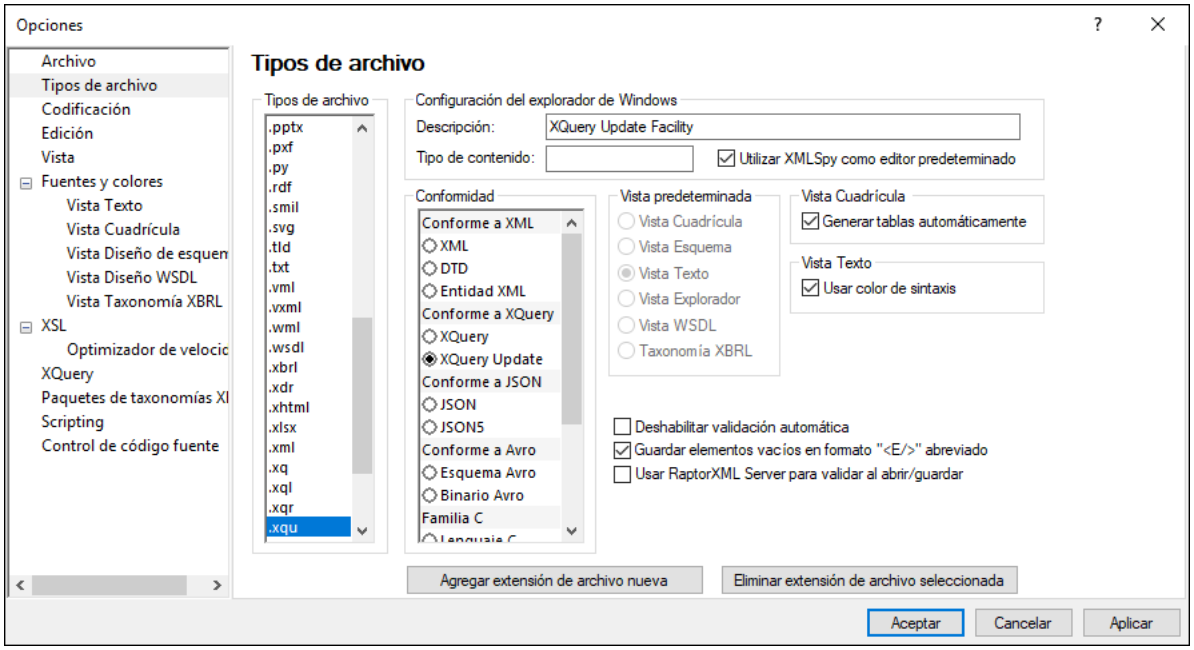
Extensiones de archivo asociadas a XQuery y XQuery Update en XMLSpy

XQuery	`.xq` `.xql` `.xqr` `.xquery`
XQuery Update	`.xqu`

Asociar más extensiones de archivo conformes a XQuery/XQuery Update

Para configurar más extensiones de archivo como conformes a XQuery:

1. Seleccione **Herramientas | Opciones**. Aparece el cuadro de diálogo "Opciones" (*imagen siguiente*).
2. Haga clic en la sección *Tipos de archivo*.



3. Haga clic en el botón **Agregar extensión de archivo nueva** para agregar la extensión de archivo nueva a la lista de tipos de archivo.
4. En el cuadro ***Conformidad*** seleccione *Conforme a XQuery* o *Conforme a XQuery Update*.

Además recomendamos seleccionar estas opciones en la misma pestaña del cuadro de diálogo:

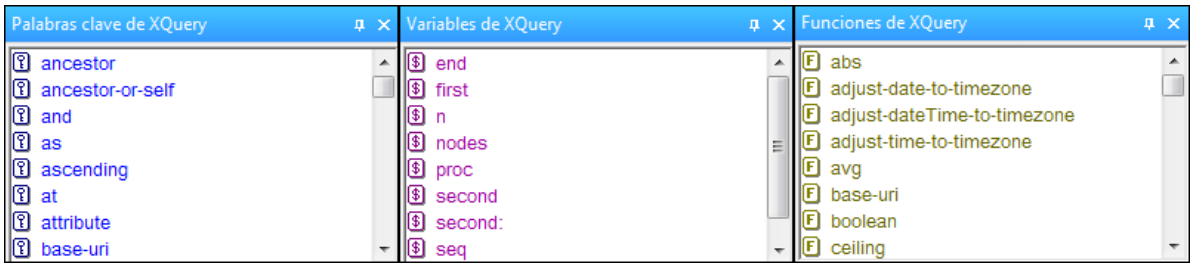
- **Descripción:** XML Query Language o XQuery Update Facility
- **Tipo de contenido**: texto/xml
- Si desea usar XMLSpy como editor predeterminado para los archivos XQuery, active la casilla **Utilizar XMLSpy como editor predeterminado**.

© 2016-2022 Altova GmbH

8.1.2 Ayudantes de entrada para XQuery

El modo XQuery de la vista Texto incluye tres ayudantes de entrada:

- **Palabras clave** de XQuery (azul)
- **Variables** de XQuery (violeta)
- **Funciones** de XQuery (verde oliva)

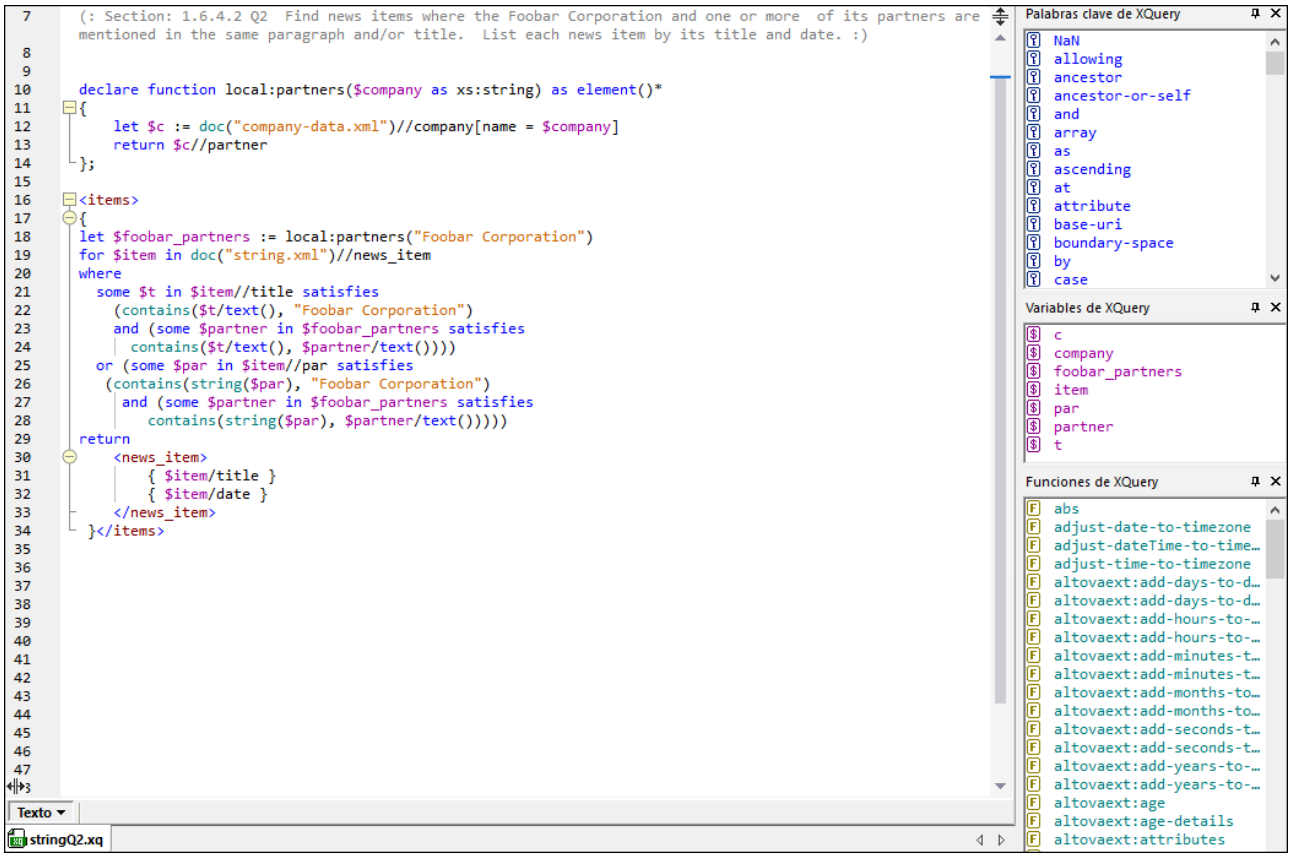


Tenga en cuenta estos aspectos de los ayudantes de entrada:

- Los elementos son de un color diferente en cada ayudante de entrada. El color del ayudante de entrada se corresponde con el color de sintaxis usado en la vista Texto. Los colores de los ayudantes de entrada no se pueden cambiar.
- Las palabras claves y funciones de los ayudantes de entrada son las palabras claves y funciones compatibles con los motores XQuery de Altova.
- Las variables son las variables definidas en el documento XQuery propiamente dicho. Cuando en la vista Texto se escribe `$` seguido de un carácter, el carácter se incluye en el ayudante de entrada Variables (a no ser que ya exista una variable con ese mismo carácter). Cuando escriba una variable y ésta coincida con un nombre de variable ya existente, la variable nueva desaparece del ayudante de entrada.
- Para navegar por los ayudantes de entrada, haga clic en un elemento del ayudante y desplácese por la lista con ayuda de la barra de desplazamiento vertical, con la rueda del ratón o con las teclas de dirección.

Para insertar un elemento de un ayudante de entrada en el documento, sitúe el cursor en el punto de inserción deseado y haga doble clic en el elemento. Tenga en cuenta que algunas cadenas de caracteres representan tanto una palabra clave como una función (hablamos de `empty`, `unordered` y `except`). Se inserta el elemento en el que haga doble clic.

8.1.3 Color de sintaxis para XQuery



En el modo XQuery de la vista Texto se utiliza este color de sintaxis para el código XQuery:

- *(: Comentarios, incluidos los delimitadores de emoticonos, aparecen en gris :)*
- Las palabras clave de XQuery aparecen en azul: **palabra-clave**
- Las variables de XQuery, incluido el signo $, aparecen en violeta: **$start**
- Las funciones de XQuery, aunque no sus paréntesis, aparecen en color verde: **función()**
- Las cadenas de texto aparecen en naranja: **"Procedure"**
- El resto del texto, como las expresiones de ruta de acceso, aparecen en negro (*subrayadas en este ejemplo*):

```
for $s in doc("report1.xml")//section[section.title = "Procedure"]
return ($s//incision)[2]/instrument
```

Puede cambiar estos colores predeterminados y otras propiedades de las fuentes en la sección Fuentes y colores[1602] del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).

8.1.4 Edición inteligente para XQuery

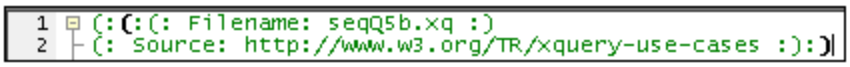
El modo XQuery de la vista Texto ofrece todas estas funciones de edición inteligente:

© 2016-2022 Altova GmbH

- Juego de paréntesis 511
- Palabras clave 511
- Variables 511
- Funciones 512
- Guías visuales 512

Juego de paréntesis

La función de juego de paréntesis resalta los paréntesis de apertura y cierre en una pareja de paréntesis para que pueda ver con claridad el contenido de la pareja de paréntesis. Esta función es muy práctica cuando los paréntesis están anidados, como en el caso de los comentarios de XQuery (*ver imagen siguiente*).



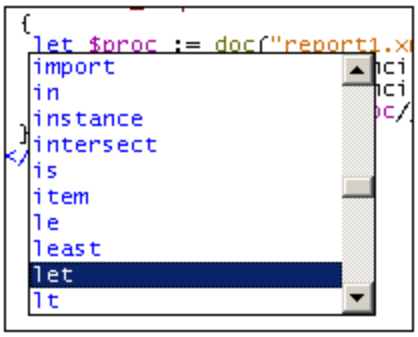
- la función de juego de paréntesis se activa al situar el cursor inmediatamente antes o después de un paréntesis (ya sea de apertura o cierre). Ambos paréntesis de la pareja de paréntesis se resaltan (en negro y en negrita). Observe la posición del cursor en la imagen anterior (después del paréntesis de cierre).

- la función de juego de paréntesis está disponible para los paréntesis propiamente dichos `()`, los corchetes `[]` y las llaves `{}`. El único tipo de paréntesis para el que no está disponible esta función son los corchetes angulares **<>**, que se utilizan en las etiquetas XML.

Nota: si coloca el cursor dentro de un paréntesis de apertura o cierre, se resaltan ambos paréntesis. Si pulsa **Ctrl+E** el cursor se desplaza al otro miembro de la pareja. Si pulsa **Ctrl+E** repetidas veces el cursor pasa de un paréntesis de la pareja al otro y así sucesivamente. Esta función ayuda a navegar por el documento con rapidez.

Palabras clave

Las palabras clave de XQuery son instrucciones utilizadas en expresiones de consulta y son de color azul. Para seleccionar una palabra clave, sitúe el cursor dentro de la palabra clave o justo antes o después de la palabra clave. Si selecciona una palabra clave y pulsa **Ctrl+Barra espaciadora**, aparece un menú desplegable con la lista de todas las palabras clave. Puede desplazarse por la lista y hacer doble clic en la palabra clave que desee utilizar para reemplazar la palabra clave actual.

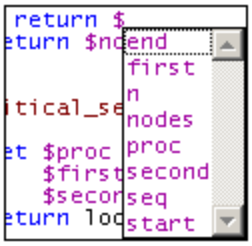


En la imagen anterior, el usuario colocó el cursor dentro de la palabra clave `let`. Al hacer doble clic en una palabra clave de la lista, esta sustituirá a la palabra clave `let`.

Variables

Los nombres de las variables llevan el prefijo **$** y son de color violeta. Este mecanismo de la función de edición inteligente es similar al utilizado con las palabras clave. Hay dos maneras de acceder a la lista emergente que contiene todas las variables:

- Escriba el carácter **$** y pulse **Ctrl+Barra espaciadora**
- Seleccione una variable y pulse **Ctrl+Barra espaciadora**. (Para seleccionar una variable, coloque el cursor inmediatamente después del carácter **$**, dentro del nombre de la variable o inmediatamente después de la variable.)



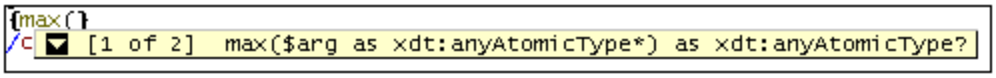
Para insertar una variable después del carácter **$** o para reemplazar la variable seleccionada, haga doble clic en la variable pertinente del menú desplegable emergente.

Funciones

Al igual que ocurre con las palabras claves y las variables, cuando se selecciona una función (en verde oliva) y se pulsa **Ctrl+Barra espaciadora**, aparece un menú con todas las funciones integradas. (Para seleccionar una función en el documento, coloque el cursor dentro del nombre de la función o justo antes o después del nombre de la función. No sirve colocar el cursor entre los paréntesis que siguen al nombre de la función). Al hacer doble clic en una función del menú emergente, esta sustituye a la función seleccionada en el documento.

Si desea ver la firma de una función (*imagen siguiente*), coloque el cursor justo después del paréntesis de apertura y pulse **Ctrl+Barra espaciadora**.

Nota: la visualización de firmas sólo está disponible para funciones de XQuery estándar.



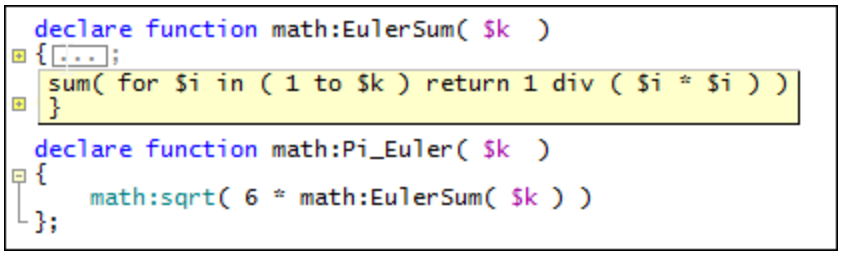
La punta de flecha apuntando hacia abajo indica que hay más de una función con ese nombre, pero con argumentos o tipos de valor devuelto diferentes. Haga clic en la punta de flecha para ver la firma de la siguiente función (en caso de que exista). Para ver las demás funciones con el mismo nombre haga clic en la punta de flecha repetidas veces. Otra opción es pulsar **Ctrl+Mayús+Arriba** o **Ctrl+Mayús+Abajo** para pasar de una función a otra.

Guías visuales

El plegamiento de texto (o plegamiento de código) funciona con las llaves de XQuery ({}), los comentarios de XQuery, los elementos XML y los comentarios XML y es una característica que permite expandir y contraer este tipo de nodos. Los nodos que se pueden expandir/contraer se marcan en el margen de plegamiento de código con el signo +/- (*imagen siguiente*). El margen puede activarse o desactivarse en el cuadro de diálogo "Configurar la vista Texto" 1489. Cuando está contraído, el nodo se marca con puntos suspensivos (*imagen siguiente*). Si pasamos el puntero por encima de los puntos suspensivos, aparece en pantalla un cuadro

© 2016-2022 Altova GmbH

emergente con el contenido del nodo contraído. Si el contenido del nodo es demasiado largo para el cuadro emergente, esto se indica con puntos suspensivos también.



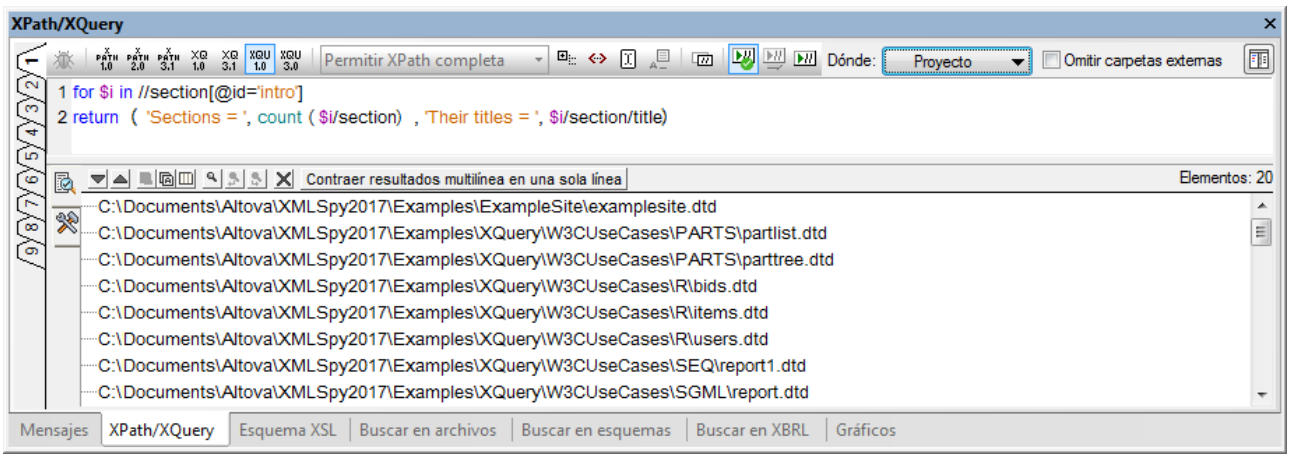
El icono **Expandir o contraer todos los plegamientos** de la barra de herramientas Texto expande **todos** los nodos o los contrae hasta el elemento de nivel superior.

Estas son las diferentes opciones que ofrece el icono +/- de los nodos:

Clic en [-]	Contrae el nodo.
Clic en [+]	Expande el nodo y sus nodos descendientes aparecen expandidos o contraídos, dependiendo de cuál fuera su estado antes de contraer el nodo matriz.
Mayús+clic en [-]	Contrae todos los nodos descendientes, pero el nodo en el que se hizo clic permanece expandido.
Ctrl+clic en [+]	Expande el nodo y todos sus descendientes.

8.2 Evaluar expresiones XQuery

En la ventana de resultados XPath/XQuery de XMLSpy puede evaluar expresiones XQuery con uno o varios documentos.



Esto se hace de la siguiente manera:

1. Escriba la expresión XQuery en el panel superior de la ventana.
2. En el cuadro combinado situado a la derecha de la barra de herramientas (*imagen anterior*) seleccione dónde está el documento XML que se debe consultar. Estas son las opciones que ofrece el cuadro combinado: **Archivo actual**, **Archivos abiertos**, **Proyecto** y **Carpeta**.
3. Haga clic en el icono **Evaluar expresión XPath/XQuery (F5)** de la barra de herramientas. La expresión se evalúa con los archivos XML. Si la ubicación indicada en el cuadro combinado contiene más de un archivo XML, XMLSpy busca estructuras de datos y contenido que cumplan la expresión en todos esos archivos. Los resultados aparecen en el panel inferior de la ventana.

En el ejemplo de la imagen anterior se realizó una consulta para un elemento llamado `section` que tiene el atributo `@id='intro'`. La consulta devuelve el número de apartados de esta sección `intro` y sus títulos.

Para más información consulte los apartados *Ventana de resultados XPath/XQuery* 127 y *Vista previa de actualizaciones* 519.

 © 2016-2022 Altova GmbH

8.3 Validar documentos XQuery

Siga estas instrucciones para validar un documento XQuery o XQuery Update:

1. Compruebe que el documento activo es el documento XQuery que desea validar.
2. Seleccione el comando **XML | Validar (F8)** o haga clic en el icono **Validar**  de la barra de herramientas.

XMLSpy valida el documento y comprueba si utiliza una sintaxis XQuery correcta.

8.4 Ejecutar XQuery/XQuery Update

XMLSpy puede ejecutar documentos XQuery y XQuery Update en estos casos:

- cuando el documento activo es un documento XQuery o XQuery Update
- y cuando el documento activo es un documento XML.

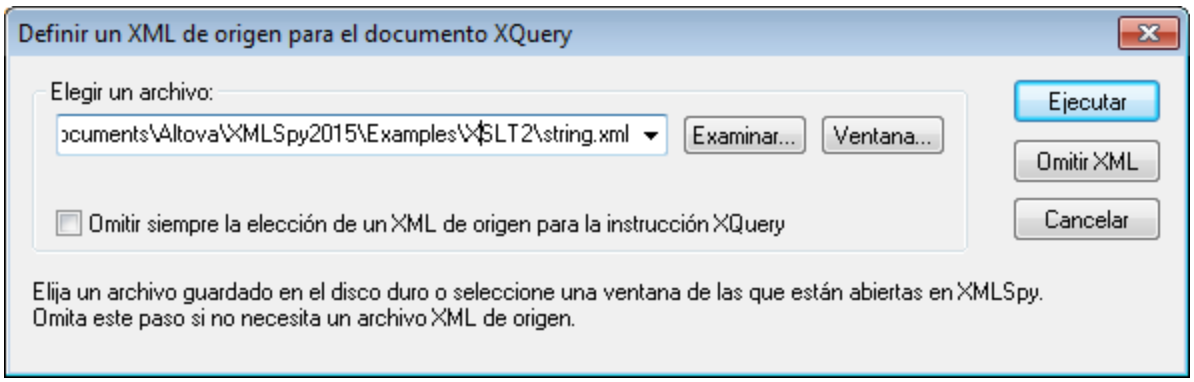
Notas:

- el tipo de documento se determina por medio de la extensión de archivo. XMLSpy reconoce los tipos de archivo en función de las definiciones realizadas en la sección *Tipos de archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones | Tipos de archivo**[1587]).

- si se trata de un documento XQuery Update también puede escribir expresiones Update en la ventana XPath/XQuery para obtener una vista previa de las actualizaciones. Si está de acuerdo con las actualizaciones, puede aplicarlas y guardar el archivo actualizado. Consulte los apartados *XQuery Update Facility*[519] y *Vista previa de actualizaciones*[519] para obtener más información.

Ejecución desde el documento XQuery/XQuery Update activo

1. Seleccione en la vista el documento XQuery/XQuery Update correspondiente.
2. Haga clic en el comando **XSL/XQuery | Ejecutar XQuery/XQuery Update** o en el icono  de la barra de herramientas. Esto abre este cuadro de diálogo:



3. Ahora tiene dos opciones: buscar el archivo XML e iniciar la ejecución o no elegir ningún XML de origen (botón **Omitir XML**).

Por lo general, los documentos XML no están asociados a un solo documento XML. Esto se debe a que las expresiones XQuery pueden seleccionar un número indefinido de documentos XML con la función `doc()`. Sin embargo, en XMLSpy, antes de ejecutar un documento XQuery, el usuario puede seleccionar un documento XML de origen para la ejecución. Cuando así lo haga, el nodo de documento del XML de origen seleccionado es el elemento de contexto inicial disponible en el nivel raíz del documento XQuery. Las rutas de acceso que empiecen con una barra diagonal se resolverán con este nodo de documento como elemento de contexto.

© 2016-2022 Altova GmbH

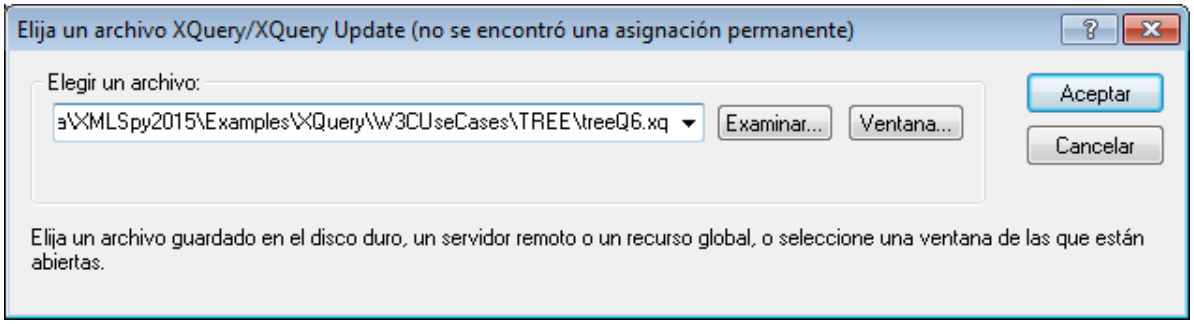
Nota: el comando **Ejecutar XQuery/XQuery Update** también está en el menú contextual de los elementos de la ventana Proyecto[122].

Resultado de la ejecución / actualización

- *Ejecución XQuery:* el documento resultante se genera como archivo temporal que se puede guardar en cualquier ubicación con el formato y la extensión de archivo deseados.
- *Actualización XQuery:* la actualización se guarda en el archivo o se abre el archivo actualizado, ofreciendo una vista previa, y después se guarda o se cierra sin guardar. El usuario puede elegir entre estas dos opciones (seleccionando el comportamiento predeterminado en la sección *XQuery* del cuadro de diálogo "Opciones" (**Herramientas | Opciones | XQuery**[1587]).

Ejecución desde el documento XML activo

1. Seleccione en la vista el documento XML correspondiente.
2. Haga clic en el comando **XSL/XQuery | Ejecutar XQuery/XQuery Update** o en el icono de la barra de herramientas. Esto abre este cuadro de diálogo:



3. Busque el archivo XQuery o XQuery Update y haga clic en **Aceptar**.

Resultado de la ejecución / actualización

- *Ejecución XQuery:* el documento resultante se genera como archivo temporal que se puede guardar en cualquier ubicación con el formato y la extensión de archivo deseados.
- *Actualización XQuery:* la actualización se guarda en el archivo o se abre el archivo actualizado, ofreciendo una vista previa, y después se guarda o se cierra sin guardar. El usuario puede elegir entre estas dos opciones (seleccionando el comportamiento predeterminado en la sección *XQuery* del cuadro de diálogo "Opciones" (**Herramientas | Opciones | XQuery**[1587]).

Asignación inversa

Si se habilita la característica de asignación inversa[1401], la ejecución XQuery se lleva a cabo de modo que el documento de salida se pueda relacionar con los documentos XQuery+XML originales. Esto significa que, si se hace clic en un nodo del documento de salida, la aplicación resaltará la **instrucción XQuery** y los **datos de origen XML** que produjeron dicho fragmento de los resultados. Además, si se hace clic en una instrucción XQuery o en un nodo de datos XML, la aplicación resaltará los nodos correspondientes en los otros dos documentos. Consulte la descripción del comando **XSL/XQuery | Habilitar asignación inversa**[1401] para obtener más información.

Variables de XQuery

Si usa los motores XQuery de Altova, puede almacenar las variables de XQuery en un práctico cuadro de diálogo de la interfaz gráfica. Todas las variables almacenadas se pasan al documento XQuery cada vez que se ejecuta un documento XQuery con XMLSpy. Para más información, consulte la descripción del comando Parámetros de XSL / Variables de XQuery 1396.

Motores XQuery de Altova

Para más información sobre la implementación de los motores XQuery de Altova y el procesamiento de archivos XQuery, consulte el anexo Implementación del motor XQuery 2117.

© 2016-2022 Altova GmbH

8.5 XQuery Update Facility

XQuery Update Facility es una extensión del lenguaje XQuery que permite modificar partes de documentos XML. Cuando se ejecuta código XQuery, se vuelve a generar el documento entero, que debe volver a guardarse en su ubicación. Sin embargo, cuando solo es necesario modificar ciertas partes del documento, este proceso de trabajo puede ser poco eficiente. Gracias a la extensión Update Facility, ahora se pueden actualizar solamente esas partes del documento.

XQuery Update Facility se describe como extensión de XQuery 1.0 y XQuery 3.1 en estas especificaciones:

- XQuery Update Facility 1.0 (Recomendación del W3C del 17 de marzo de 2011)
- XQuery Update Facility 3.0 (Borrador de trabajo del W3C del 19 de febrero de 2015)

XQuery Update Facility en XMLSpy

A continuación se enumera el funcionamiento general de XQuery Update en XMLSpy:

- Cada actualización la lleva a cabo una expresión de actualización. Por ejemplo, una expresión de actualización puede indicar que se cambie el nombre de un nodo del documento XML:
  ```
  rename node /documents/doc-01 as "document-01"
  ```
- En la práctica, el documento XQuery Update incluye varias expresiones de actualización.
- A medida que se ejecutan las expresiones de actualización del documento, el resultado de cada una de ellas no se aplica automáticamente, sino que se añade a la denominada *lista de actualizaciones pendientes* (o *LAP*). Es decir, la LAP contiene los resultados de todas las expresiones de actualización. Todas las actualizaciones de la LAP se aplican de una sola vez.
- En XMLSpy las actualizaciones de la LAP se pueden aplicar de dos maneras:

  - Tras haber sido aprobadas por el usuario en la interfaz gráfica: la ventaja es que la actualización se puede anular si la vista previa muestra resultados no deseados. (La vista previa se genera al ejecutar el comando **Ejecutar XQuery/XQuery Update** 516 o cuando se evalúan expresiones XQuery Update en la ventana XPath/XQuery 519.)

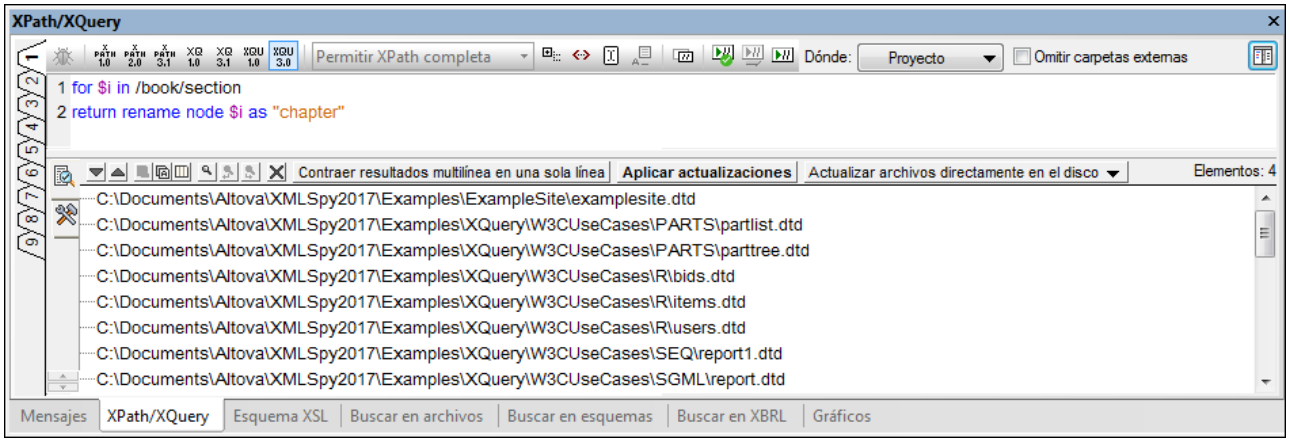
  - Sin necesidad de que intervenga el usuario: la ventaja es que al actualización se lleva a cabo en silencio. Esta aplicación directa de las actualizaciones (sin generar una vista previa) está disponible al ejecutar el comando **Ejecutar XQuery/XQuery Update** 516 o cuando se evalúan expresiones XQuery Update en la ventana XPath/XQuery 519.

La vista previa de XQuery Update de XMLSpy es una potente característica que le permitirá ver el efecto que tendrían las expresiones de actualización en el documento XML antes de aplicarlas. Para más información consulte el apartado Vista previa de actualizaciones 519.

8.5.1 Vista previa de actualizaciones

Si quiere modificar un documento XML con ayuda de expresiones XQuery Update, puede ver el efecto de dichas expresiones en una vista previa antes de aplicarlas y de guardar las modificaciones en el documento XML.

En la ventana de resultados XPath/XQuery (*imagen siguiente*), puede escribir expresiones de actualización y obtener una vista previa de las actualizaciones en la *lista de actualizaciones pendientes* (llamada *LAP*). Esta vista previa aparece en el panel inferior de la ventana. Si está de acuerdo con las actualizaciones de la LAP, podrá aplicar las actualizaciones en el documento y guardarlo. Si no quiere realizar ninguno de los cambios de la LAP, puede no aplicar las actualizaciones o no guardar el documento.



Para generar una lista de actualizaciones pendientes para el documento XML activo:

1. Seleccione la opción **XQU 1.0** o **XQU 3.0** en la barra de herramientas de la ventana XPath/XQuery (*imagen anterior*), según corresponda.
2. En el panel superior de la ventana escriba las expresiones de actualización que desee utilizar. Para ver una descripción de las expresiones de actualización y de su sintaxis consulte este apartado 522.
3. En el cuadro combinado situado al final de la barra de herramientas, seleccione la ubicación donde se deben realizar las actualizaciones: (i) **Archivos abiertos**, (ii) **Archivo actual**, (iii) **Proyecto** o (iv) **Carpeta**.
4. Para ejecutar las expresiones de actualización y ver la LAP, haga clic en el icono **Evaluar expresión XPath/XQuery (F5)** de la barra de herramientas.

Barra de herramientas de la ventana XPath/XQuery

La ventana XPath/XQuery tiene una barra de herramientas con varios iconos. Los que están marcados en rojo en la siguiente imagen se describen a continuación, de izquierda a derecha:



- **Mostrar encabezado:** active este icono si quiere que el nombre del archivo donde se realizarán las actualizaciones
- **Mostrar resultado completo:** active este icono para ver todo el contenido del nodo o solo sus atributos.
- **Definir como origen para XPath/XQuery:** define la posición actual del cursor en el archivo activo como origen de las expresiones XPath relativas. La expresión se evalúa en relación a este nodo.
- XMLSpy puede introducir automáticamente la expresión de localización de los nodos para que usted no tenga que hacerlo a mano. Siga estas instrucciones: (i) ponga el cursor en la posición de la expresión XPath donde desea introducir la ruta de localización; (ii) ponga el cursor en la etiqueta de

 © 2016-2022 Altova GmbH

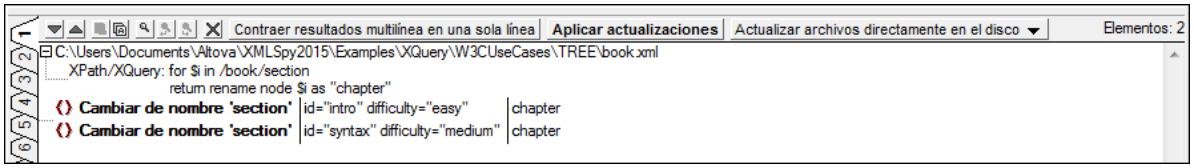
apertura del nodo al que desea apuntar; (iii) ahora haga clic en el icono "Copia la XPath de la selección actual en el campo de edición" para introducir la ruta de localización del nodo seleccionado en la expresión. La ruta de localización se introduce como ruta de acceso absoluta que empieza en el nodo raíz del documento.

- **Validar archivos XML:** activa la validación de los archivos XML.
- **Evaluar la expresión XPath/XQuery mientras se teclea:** este comando se habilita si en el cuadro combinado está elegida la opción Archivo actual. Si activa este comando, la LAP se genera mientras se escribe la expresión de actualización.
- **Evaluar expresión XPath/XQuery:** este comando evalúa las expresiones de actualización y genera la LAP.
- **Dónde:** este cuadro combinado indica el ámbito de la operación.

Lista de actualizaciones pendientes

El panel de la LAP muestra todas las actualizaciones que se llevarán a cabo. Si activó el comando Mostrar encabezado en la barra de herramientas de la ventana, la lista incluye la ubicación de los archivos de destino. La LAP se divide en tres secciones verticales (*imagen siguiente*):

i. La actualización que se debe llevar a cabo
ii. El contenido del nodo de destino que se debe actualizar
iii. El resultado que tendrá la actualización



El panel de la LAP también incluye una barra de herramientas con varios iconos y botones:

- **Siguiente** y **Anterior**: estos iconos seleccionan el mensaje siguiente y el mensaje anterior respectivamente.
- **Copiar...**: estos iconos copian la línea seleccionada o todas las líneas en el portapapeles respectivamente.
- **Buscar**: estos iconos buscan texto en el panel.
- **Borrar**: elimina todas las líneas del panel.
- Botón **Contraer resultados multilínea en una sola línea**: contrae los elementos de la lista que tengan varias líneas en una sola línea.
- Botón **Aplicar actualizaciones**: aplica las actualizaciones pendientes en las ubicaciones de destino. Estas actualizaciones se pueden guardar en el archivo directamente o XMLSpy puede abrir el archivo actualizado para que el usuario lo guarde o no (ver siguiente punto).
- Cuadro combinado **Actualizar archivos directamente**: permite elegir (i) si los archivos se actualizan en silencio en el disco o (ii) si los archivos actualizados se abren primero en la aplicación. Seleccione esta opción para ver los archivos y decidir si guarda los cambios o no.

Nota: si actualiza archivos directamente en el disco, aparece una lista de archivos modificados. Cada elemento de esta lista muestra la ubicación del archivo. Haga clic en la ubicación para ir al archivo.

8.5.2 Operaciones y sintaxis de XQuery Update

La extensión XQuery Update Facility permite realizar todas estas operaciones:

- Eliminar 522 nodos
- Insertar 522 nodos antes, después o dentro de un nodo determinado
- Cambiar el nombre 523 de los nodos
- Reemplazar 524 nodos con una secuencia de elementos
- Reemplazar el valor 524 de los nodos con el valor de cadena de una secuencia de elementos

Los apartados que aparecen a continuación describen las palabras claves y la sintaxis de todas estas operaciones.

8.5.2.1 Eliminar nodos

Descripción y sintaxis
Elimina uno o varios nodos.

```
delete node secuenciaNodos
delete nodes secuenciaNodos
```

Detalles

- La expresión *secuenciaNodos* devuelve una secuencia de nodos que se deben eliminar. Todos los nodos seleccionados se marcarán para ser eliminados.
- No importa si utiliza **node** (en singular) o **nodes** (en plural). No tiene que coincidir con el número de elementos de *secuenciaNodos*.

Ejemplos

```
for $i in /book/section return
delete nodes $i/@id
```

8.5.2.2 Insertar nodos

Descripción y sintaxis
Inserta uno o varios nodos antes, después o dentro del nodo de destino indicado.

```
insert (node|nodes) elementos into nodoDestino
insert (node|nodes) elementos as first into nodoDestino
insert (node|nodes) elementos as last into nodoDestino
insert (node|nodes) elementos before nodoDestino
insert (node|nodes) elementos after nodoDestino
```


© 2016-2022 Altova GmbH

Detalles

- La expresión *elementos* debe devolver una secuencia de elementos. Incluso si utiliza la palabra clave **node|nodes**, *elementos* puede ser una secuencia de elementos no nodos.
- La expresión *nodoDestino* debe apuntar a un solo nodo de destino.
- Si se usa la palabra clave **into**, *nodoDestino* debe ser un nodo de elemento o un nodo de elemento de documento.
- Si se usa la palabra clave **as first** o **as last**, el elemento se inserta como primer o último secundario respectivamente.
- Si solo se usa la palabra clave **into**, los atributos se anexan a los atributos disponibles y los elementos se insertan como primeros secundarios.
- Si se usa la palabra clave **before** o **after**, *nodoDestino* puede se de cualquier tipo.
- Si se inserta un atributo, no puede llamarse igual que ningún atributo ya existente.

Ejemplos

```
for $i in /book/section return
insert nodes (attribute id { 'somevalue' }, <newelement> contenido que incluya los números
"{ 1 to 3}"</newelement>)
into $i
```

8.5.2.3 Renombrar nodos

Descripción y sintaxis
Cambia el nombre de un nodo de elemento, atributo o instrucción de procesamiento.

```
rename node nodoDestino as nombre
```

Detalles

- La expresión *nodoDestino* debe apuntar a un solo nodo de destino, que puede ser un elemento, un atributo o una instrucción de procesamiento.
- La expresión *nombre* debe dar como resultado un QName o una cadena de texto.
- Si se construye un QName, el espacio de nombres obligatorio se declara localmente.

Ejemplos

```
rename node /book/title as 'header-1'

rename node /book/title as QName("http://www.altova.com/xquf", "header-1")
```

8.5.2.4 Reemplazar nodos

Descripción y sintaxis
Reemplaza un nodo con una secuencia formada por cualquier tipo de elementos.

```
replace node nodoDestino with elementos
```

Detalles

- La expresión *nodoDestino* debe apuntar a un solo nodo de destino.
- La expresión *elementos* debe devolver una secuencia de elementos. Esta secuencia reemplazará al nodo de destino.
- Excepto si se trata de un nodo de atributo, el nodo de destino se puede reemplazar con cualquier tipo de secuencia.
- Los nodos de atributo solamente se pueden reemplazar con otro nodo de atributo (*ver ejemplo*.)

Ejemplos

```
replace node //hr with '<line/>'

for $i in //@height return
replace node $i with (attribute line-height{'12pt'})
```

8.5.2.5 Reemplazar valores de nodos

Descripción y sintaxis
Reemplaza el valor de un nodo con el valor de cadena de una secuencia de elementos.

```
replace value of node nodoDestino with elementos
```

Detalles

- La expresión *nodoDestino* debe apuntar a un solo nodo de destino.
- La expresión *elementos* debe devolver una secuencia de elementos.
- El contenido del nodo de destino se reemplaza con el valor de cadena de la secuencia que devuelve la expresión *elementos*. Esto significa que el nodo de destino incluirá un nodo de texto solamente.

Ejemplos

```
for $i in //title return
replace value of node $i with ('Draft Title')
```


© 2016-2022 Altova GmbH

8.5.2.6 Función fn:put

La función **fn.put** forma parte de la especificación XQuery Update Facility 1.0 y es una extensión de la biblioteca de funciones integrada de XQuery. En esta documentación asumimos que el prefijo de espacio de nombres **fn:** está enlazado con el espacio de nombres `http://www.w3.org/2005/xpath-functions`.

```
fn:put($nodo como nodo(), $uri como xs:string) como secuencia-vacía()
```

La función almacena un documento o elemento en la ubicación indicada por `$uri`. Por lo general se invoca para crear un recurso en un sistema de almacenamiento externo, como un sistema de archivos o una base de datos. El efecto que produce la función `fn:put` depende de la implementación porque tiene lugar fuera del dominio de XQuery. El propósito de esta función es que si se invoca en un nodo de documento y no se emite ningún error, entonces otra consulta puede acceder al documento almacenado mediante la invocación de `fn:doc` con el mismo URI.

Consulte la *especificación para obtener más información.*

8.6 XQuery y bases de datos XML

Puede usar un documento XQuery para consultar una base de datos XML (BD XML). De momento, esta función de XQuery sólo es compatible con las bases de datos DB2 de IBM. El mecanismo para consultar una BD XML con un documento XQuery consiste en: (i) indicar al motor XQuery que debe consultar datos XML en una base de datos (en lugar de datos XML en un documento XML) y (ii) acceder a los datos XML de la base de datos.

Estos son los pasos necesarios para implementar este mecanismo:

1. Configurar el documento XQuery 526 para consultar la BD XML, insertando la palabra clave `XQUERY` al principio del documento.
2. Habilitar la compatibilidad con BD 526 (en la ventana Información) para el documento activo y conectarse a la BD 526 (desde el cuadro de diálogo "Conexión rápida").
3. Insertar extensiones de XQuery propias de BD 527 para poder acceder a los datos de la BD y ponerlos a disposición de las operaciones de XQuery.
4. Ejecutar el documento XQuery 526 en XMLSpy.

Configurar el documento XQuery a fin de consultar la BD XML

Para configurar el documento XQuery a fin de consultar la BD XML, abra el documento XQuery (o cree un documento XQuery nuevo) y escriba la palabra clave `XQUERY` (en mayúsculas o minúsculas, da igual) al principio del documento (antes del prólogo) como en los ejemplos que aparecen a continuación.

```
XQUERY (: Recuperar detalles de todos los clientes :)
declare default element namespace "http://www.altova.com/xquery/databases/db2";
<a> {db2-fn:xmlcolumn("CUSTOMER.INFO")} </a>
```

Es necesario insertar la palabra clave `XQUERY` aunque el documento use la expresión opcional `xquery version`:

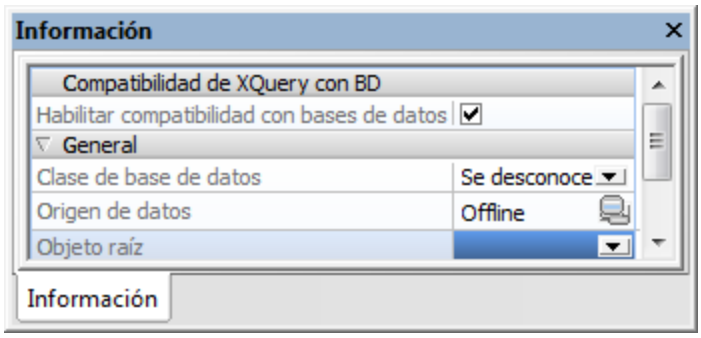
```
XQUERY xquery version "1.0"; (: Recuperar detalles de todos los clientes :)
declare default element namespace
"http://http://www.altova.com/xquery/databases/db2";
<a> {db2-fn:xmlcolumn("CUSTOMER.INFO")} </a>
```

Nota: los motores XQuery integrados de XMLSpy leen la palabra clave `XQUERY` y entienden que es necesario acceder a una BD XML. Por tanto, se producirá un error si intenta ejecutar un documento XQuery que contenga la palabra clave `XQUERY` en un documento XML que no esté en una BD XML.

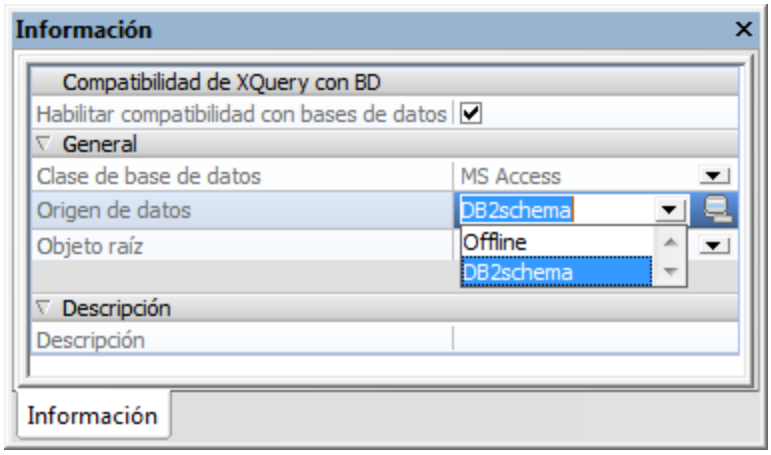
Habilitar la compatibilidad con BD y conectarse a la BD

Para habilitar la compatibilidad del documento XQuery con bases de datos, marque la casilla *Habilitar compatibilidad con bases de datos* de la ventana Información (*ver imagen siguiente*). Tenga en cuenta que es necesario habilitar esta opción en cada documento XQuery y cada vez que abra el documento XQuery.

© 2016-2022 Altova GmbH



Al marcar la casilla *Habilitar compatibilidad con bases de datos*, aparece un cuadro de diálogo de conexión rápida a bases de datos. De momento, esta función es compatible sólo con bases de datos DB2 de IBM. Para más información sobre cómo conectarse a una base de datos, consulte la sección Conectarse a un origen de datos [952]. Si ya existen conexiones a orígenes de datos, estas aparecen en el cuadro combinado *Origen de datos* de la ventana Información (*imagen siguiente*) y puede seleccionar uno de estos orígenes de datos como origen de datos para el documento XQuery activo. En la ventana Información también puede seleccionar el objeto raíz en el cuadro combinado *Objeto raíz*.



Para acceder al cuadro de diálogo de conexión rápida a bases de datos (que le permite conectarse a una BD) basta con hacer clic en el icono de la ventana Información.

Nota: cuando se cierra un documento XQuery también se cierra la conexión con la BD. Cuando vuelva a abrir el documento XQuery, también deberá volver a conectarse a la BD.

Extensiones del lenguaje XQuery propias de DB2 IBM

Hay dos funciones propias de DB2 IMB que se pueden usar en documentos XQuery para recuperar datos desde una base de datos DB2 de IBM:

- `db2-fn:xmlcolumn` recupera una columna XML entera sin realizar búsquedas ni aplicar filtros a la columna.
- `db2-fn:sqlquery` recupera valores de acuerdo con una instrucción SQL `SELECT`

Los datos XML que se recuperen usando estas funciones se pueden manipular con construcciones XQuery estándar (*ver los ejemplos que aparecen a continuación*).

db2-fn:xmlcolumn: el argumento de la función es un literal de cadena (que distingue entre mayúsculas y minúsculas) que identifica una columna XML en una tabla. El argumento del literal de cadena debe ser un nombre completo de columna de tipo XML. La función devuelve todos los datos XML de la columna en forma de secuencia, sin aplicarle una condición de búsqueda. En el ejemplo que sigue, todos los datos de la columna XML `INFO` de la tabla `CUSTOMER` se devuelven dentro de un elemento de nivel superior `<newdocelement>`:

```
XQUERY (: Recuperar detalles de todos los clientes :)
declare default element namespace "http://www.altova.com/xquery/databases/db2";
<newdocelement> {db2-fn:xmlcolumn("CUSTOMER.INFO")} </newdocelement>
```

Después se puede realizar una consulta en los datos recuperados usando construcciones XQuery. En el ejemplo que sigue los datos XML recuperados de la columna XML `INFO` de la tabla `CUSTOMER` se filtran usando una construcción XQuery a fin de recuperar sólo los perfiles de los clientes de Toronto.

```
XQUERY (: Recuperar detalles de los clientes de Toronto :)
declare default element namespace "http://www.altova.com/xquery/databases/db2";
<newdocelement> {db2-fn:xmlcolumn("CUSTOMER.INFO")/customerinfo[addr/city='Toronto']}
</newdocelement>
```

Nota: en el ejemplo anterior el elemento de documento de los archivos XML en cada celda es `customerinfo` y el nodo raíz de la secuencia XML devuelta por `db2-fn:xmlcolumn` se considera nodo abstracto sobre los nodos `customerinfo`.

db2-fn:sqlquery: la función toma como argumento una instrucción SQL `SELECT` y devuelve una secuencia de valores XML. Después se puede realizar una consulta en la secuencia recuperada usando construcciones XQuery. En el ejemplo que sigue la columna `INFO` se filtra para recuperar los registros de la tabla `CUSTOMER` que tengan un campo `CID` cuyo valor esté entre `1000` y `1004`. Tenga en cuenta que SQL no distingue entre mayúsculas y minúsculas, pero XQuery sí lo hace.

```
XQUERY (: Recuperar detalles de clientes según su Id de cliente:)
declare default element namespace "http://www.altova.com/xquery/databases/db2";

<persons>
   {db2-fn:sqlquery("SELECT info FROM customer WHERE CID>1000 AND CID<1004")/
   <person>
      <id>{data(@Cid)}</id>
      <name>{data(name)}</name>
   </person>}
</persons>
```

El documento XQuery anterior devuelve este resultado:

```
<persons xmlns="http://www.altova.com/xquery/databases/db2">
   <person>
       <id>1001</id>
       <name>Kathy Smith</name>
   </person>
   <person>
       <id>1002</id>
       <name>Jim Jones</name>
   </person>
   <person>
       <id>1003</id>
```


© 2016-2022 Altova GmbH

```
      <name>Robert Shoemaker</name>
   </person>
</persons>
```

Tenga en cuenta estos aspectos:

- La declaración default element namespace del prólogo afecta a todo el documento XQuery y se utiliza tanto para navegar por el documento XML como para construir elementos nuevos. Esto significa que el selector XQuery `name` se expande a `<default-element-namespace>:name` y que los elementos construidos, como `persons`, están en el espacio de nombres de elemento predeterminado.
- La instrucción SQL `Select` no distingue entre mayúsculas y minúsculas.
- La cláusula `WHERE` de la instrucción `Select` debe hacer referencia a otro elemento de la base de datos y no a un nodo del archivo XML al que se accede.
- La barra "`/`" situada después de la función `db2-fn:sqlquery` representa el primer elemento de la secuencia devuelta y este elemento es el nodo de contexto que se utilizará para navegar por el documento.

Ejecutar el documento XQuery

Para ejecutar el documento XQuery, seleccione el comando **XSL/XQuery | Ejecutar XQuery**. También puede pulsar la combinación de teclas **Alt+F10** o hacer clic en el icono "Ejecutar XQuery" . El resultado de la ejecución aparece en un documento nuevo.

9 Depurador y generador de perfiles XSLT y XQuery

XMLSpy ofrece un depurador XSLT/XQuery 531 y un generador de perfiles XSLT/XQuery 558 para ayudarle a crear documentos XQuery correctos y válidos más rápido. Estas características se describen en los apartados de esta sección.

© 2016-2022 Altova GmbH

9.1 Depurador XSLT y XQuery

Con el depurador XSLT y XQuery puede probar y depurar hojas de estilos XSLT y documentos XQuery. La interfaz del depurador XSLT y XQuery presenta vistas simultáneas del documento XSLT o XQuery, el documento de resultados y el documento XML de origen. Después puede recorrer el documento XSLT o XQuery paso a paso. El resultado correspondiente se genera paso a paso y, si se muestra el archivo XML de origen, por cada paso se resalta la posición correspondiente en el archivo XML. Asimismo, las ventanas de la interfaz ofrecen información sobre el proceso de depuración.

El depurador XSLT y XQuery siempre se abre dentro de una **sesión de depuración**. Hay varios tipos de sesiones de depuración:

- XSLT 1.0, que utiliza el motor XSLT 1.0 integrado de Altova
- XSLT 2.0, que utiliza el motor XSLT 2.0 integrado de Altova
- XSLT 3.0, que utiliza el motor XSLT 3.0 integrado de Altova
- XQuery 1.0, que utiliza el motor XQuery 1.0 integrado de Altova
- XQuery 3.1, que utiliza el motor XQuery 3.1 integrado de Altova

Nota: el depurador XSLT y XQuery no es compatible todavía con las extensiones XQuery Update Facility 1.0 y XQuery Update Facility 3.0.

El tipo de sesión de depuración que se abre en XMLSpy dependerá del tipo de documento desde el que se abra la sesión de depuración (que en adelante llamaremos *documento activo* o *archivo activo*). Las sesiones de depuración XSLT se abren desde archivos XSLT (cuya versión depende del valor del atributo `version` del elemento `xsl:stylesheet` (o `xsl:transform`) de la hoja de estilos XSLT (`"1.0"` para XSLT 1.0, `"2.0"` para XSLT 2.0 y `"3.0"` para XSLT 3.0). Las sesiones de depuración XQuery se abren desde archivos XQuery de forma similar. Si el archivo activo es un archivo XML, el tipo de sesión dependerá del archivo que usted elija ejecutar en el archivo XML (un archivo XSLT o un archivo XQuery) y del tipo de hoja de estilos XSLT elegida (XSLT 1.0, XSLT 2.0 o XSLT 3.0).

En esta tabla resumimos todo lo descrito hasta ahora.

Archivo activo	Archivo asociado	Sesión de depuración
XSLT 1.0	XML (obligatorio)	XSLT 1.0 (con el motor integrado XSLT 1.0 de Altova)
XSLT 2.0	XML (obligatorio)	XSLT 2.0 (con el motor integrado XSLT 2.0 de Altova)
XSLT 3.0	XML (obligatorio)	XSLT 3.0 (con el motor integrado XSLT 3.0 de Altova)
XQuery 1.0	XML (opcional)	XQuery 1.0 (con el motor integrado XQuery 1.0 de Altova)
XQuery 3.1	XML (opcional)	XQuery 3.1 (con el motor integrado XQuery 3.0 de Altova)
XML	XSLT 1.0, 2.0 o 3.0 / XQuery 1.0 o 3.1 (obligatorio)	XSLT 1.0, XSLT 2.0 o XSLT 3.0 o XQuery 1.0 o 3.1. La versión de XSLT depende del valor del atributo `version` del elemento `xsl:stylesheet` (o `xsl:transform`) de la hoja de estilos XSLT.

Para más información sobre los tres motores de Altova, consulte estas secciones:

- Motor XSLT 1.0 de Altova 2114
- Motor XSLT 2.0 de Altova 2114

- Motor XQuery 1.0 de Altova 2117

Automatizar las tareas XSLT y XQuery con Altova Raptor XML Server 2023

Altova RaptorXML Server incluye el validador XML y los motores XSLT 1.0/2.0/3.0 y XQuery 1.0/3.1 de Altova. Se puede utilizar desde la línea de comandos, a través de la interfaz de COM, en programas Java y en aplicaciones .NET para validar documentos XML, transformar documentos XML usando hojas de estilos y ejecutar documentos XQuery.

Por tanto, puede usar RaptorXML Server para automatizar las tareas XSLT y XQuery. Por ejemplo, puede crear un archivo por lotes que llame a RaptorXML para que transforme un grupo de documentos XML o para que ejecute un conjunto de documentos XQuery. Para más información, consulte la documentación de RaptorXML.

Sitio web de Altova: Depurador XSLT, Depurador XQuery

9.1.1 Funcionamiento e interfaz

El mecanismo utilizado para depurar archivos XSLT y XQuery con el depurador XSLT y XQuery se describe a continuación a grandes rasgos. Tenga en cuenta que, aunque ambos se inicien con el comando **XSL/XQuery | Iniciar el depurador**, una sesión de depuración y el depurador propiamente dicho son dos cosas distintas. Primero se inicia la sesión de depuración y después se recorre el documento XSLT o XQuery paso por paso con el depurador.

- Abra una sesión de depuración (con el comando **XSL/XQuery | Iniciar el depurador**). Dependiendo del tipo de archivo activo (ver Depurador XSLT y depurador XQuery 531), se selecciona una sesión de depuración u otra (XSLT 1.0/2.0/3.0 o XQuery 1.0/3.1). El depurador XSLT y XQuery sólo funciona en la vistas Texto y Cuadrícula. Si el documento activo no está en la vista Texto o Cuadrícula al iniciar la sesión de depuración, se solicita el permiso del usuario para cambiar a la vista Texto, que es la vista predeterminada del depurador XSLT y XQuery. También puede configurar el depurador para que se cambie a la vista Texto automáticamente en el cuadro de diálogo Configurar depuración 537.
- Recorra el documento XSLT o XQuery con los comandos **Paso a paso por instrucciones**, **Paso a paso para salir** y **Paso a paso por procedimientos** del menú **XSL/XQuery**. (Si en este momento hace clic en el comando **Iniciar el depurador**, el depurador recorre toda la transformación o ejecución, deteniéndose sólo en los puntos de interrupción. Si no se establecieron puntos de interrupción, el depurador recorre toda la transformación en un solo paso, sin mostrar los resultados de la depuración.) Si se asoció un archivo XML a la sesión de depuración, se resaltan las posiciones correspondientes del archivo XML. De forma simultánea se generan resultados para los pasos correspondientes en el archivo de resultados y el documento de resultados se genera paso a paso. De este modo puede analizar qué hace cada instrucción del archivo XSLT o XQuery.

© 2016-2022 Altova GmbH



Si lo desea, en lugar de la vista de tres documentos (XML, XSLT o XQuery y Resultados) que aparece en la imagen anterior, puede seleccionar una vista de dos documentos (XSLT o XQuery y Resultados) o una vista con uno de estos tres documentos.

- Mientras está abierta una sesión de depuración, la interfaz incluye ventanas de información con detalles sobre los aspectos de la transformación o ejecución (variables, inspección XPath, pila de llamadas, mensajes, información, etc.).
- Mientras está abierta una sesión de depuración, puede detener el depurador (lo cual no significa detener la sesión de depuración) para realizar cambios en los documentos. Mientras edita un documento durante una sesión de depuración están disponibles todas las funciones de edición de su versión de XMLSpy. Cuando se detiene el depurador, la interfaz del depurador XSLT y XQuery sigue abierta y puede acceder a toda la información de las ventanas de información. Tras detener el depurador en una sesión de depuración puede reiniciar el depurador (desde el principio del documento XSLT o XQuery) dentro de la misma sesión de depuración.
- Puede establecer puntos de interrupción en el archivo XSLT para interrumpir el proceso en determinados puntos. Esto agiliza las sesiones de depuración porque no es necesario detenerse de forma manual en cada instrucción del documento XSLT o XQuery.
- En el archivo XSLT también puede establecer puntos de seguimiento. Para las instrucciones en las que se establece un punto de seguimiento, el valor de dicha instrucción es el resultado al llegar a la instrucción.
- Al detener la sesión de depuración, se cierra la interfaz del depurador XSLT y XQuery y se vuelve al entorno de XMLSpy que estaba abierto antes de iniciar la sesión de depuración. Los datos de las ventanas de información desaparecen. Sin embargo, la información sobre los puntos de interrupción y seguimiento se conserva en los archivos correspondientes hasta que se cierren los archivos. (Es decir, si inicia otra sesión de depuración con un archivo que contenga puntos de interrupción, estos puntos de interrupción se usarán en la nueva sesión de depuración.)

Nota: la barra de herramientas del depurador, con los iconos del depurador, aparece automáticamente al iniciar una sesión de depuración.

9.1.2 Comandos e iconos de la barra de herramientas

Estos comandos de menú están presentes tanto en el menú **XSL/XQuery** como en forma de iconos en la barra de herramientas. Los iconos del depurador se habilitan automáticamente en la barra de herramientas al abrir una sesión de depuración. Estos iconos se describen a continuación.



Iniciar el depurador (Alt+F11)
Inicia o continúa el procesamiento del documento XSLT o XQuery hasta el final. Si se establecieron puntos de interrupción, el procesamiento se detiene en dichos puntos. Si aún no se inició la sesión de depuración, este comando inicia la sesión y se detiene en el primer nodo que se debe procesar. Si la sesión está activa, el documento XSLT o XQuery se procesa hasta el final o hasta que se encuentra el siguiente punto de interrupción. Si se establecieron puntos de seguimiento, la ventana Seguimiento muestra el valor de la instrucción en la que se estableció el punto de seguimiento al llegar a dicha instrucción.



Ver únicamente el documento activo
Maximiza la ventana del documento activo en la interfaz del depurador.



Ver XSLT/XQuery y resultado
Muestra las ventanas del documento XSLT o XQuery y oculta el documento XML.



Ver XML, XSLT/XQuery y resultado
Muestra los documentos XML, XSLT/XQuery y el archivo de resultados. Esta es la vista predeterminada cuando se asocia un documento XML para la sesión de depuración.



Detener el depurador
Se detiene el depurador, pero la sesión de depuración no se detiene. Este comando es muy práctico si desea editar un documento en medio de la sesión de depuración o si desea utilizar otros archivos dentro de la misma sesión de depuración. Tras detener el depurador, es necesario reiniciarlo para empezar la depuración desde el principio del documento XSLT o XQuery.



Paso a paso por instrucciones (F11)
Recorre paso a paso todos los nodos y expresiones XPath de la hoja de estilos. Este comando también sirve para reiniciar el depurador después de detenerlo.

© 2016-2022 Altova GmbH



Paso a paso por procedimientos (Ctrl+F11)
Pasa de un nodo al siguiente nodo del mismo nivel o al siguiente nodo en el nivel inmediatamente superior al nodo actual. Este comando también sirve para reiniciar el depurador después de detenerlo.



Paso a paso para salir (Mayús+F11)
Pasa de un nodo al siguiente nodo del mismo nivel que el nodo primario o al siguiente nodo en el nivel inmediatamente superior al nodo primario.



Mostrar nodo en ejecución
Muestra o selecciona el nodo que está actualmente en ejecución en el documento XSLT/XQuery y el nodo de contexto correspondiente del documento XML. Si hizo clic en otras pestañas que muestran o marcan determinado código en la hoja de estilos XSLT o en el archivo XML, puede usar este comando para volver a la posición donde estaba antes de hacer clic en esas otras pestañas.



Reiniciar el depurador
Borra el contenido de la ventana de resultados y reinicia la sesión de depuración con los archivos seleccionados en ese momento.



Insertar o quitar punto de interrupción (F9)
Inserta o quita un punto de interrupción en el punto de inserción del cursor. Puede definir puntos de interrupción inline para nodos tanto en los documento XSLT y XQuery como en documentos XML y determinar dónde se debe parar el procedimiento. Al establecer un punto de interrupción, aparece una línea discontinua roja sobre el nodo. No está permitido establecer puntos de interrupción en nodos finales y en los documentos XSLT se ignoran los puntos de interrupción establecidos en atributos. Este comando también está disponible en el menú contextual que aparece al hacer clic con el botón derecho en un nodo.



Insertar o quitar punto de seguimiento (Mayús+F9)
Inserta o quita un punto de seguimiento en el punto de inserción del cursor. Puede definir puntos de seguimiento inline para nodos de documentos XSLT. Cuando en una sesión de depuración se llega a una instrucción que tiene un punto de seguimiento, el resultado de la instrucción aparece en la ventana Seguimiento. Al establecer un punto de seguimiento, aparece una línea discontinua azul sobre el nodo. No está permitido definir puntos de seguimiento en nodos finales. Este comando también está disponible en el menú contextual que aparece al hacer clic con el botón derecho en un nodo.



Habilitar o deshabilitar punto de interrupción (Ctrl+F9)
Este comando (que carece de icono en la barra de herramientas) habilita o deshabilita los puntos de interrupción existentes en el documento. La marca visual roja del punto de interrupción se vuelve de color gris

cuando se deshabilita el punto de interrupción. El depurador no se detiene en los puntos de interrupción deshabilitados. Para deshabilitar o habilitar un punto de interrupción, haga clic con el botón derecho en el nombre del nodo y en el menú contextual seleccione **Puntos de interrupción y seguimiento | Habilitar o deshabilitar punto de interrupción** (o sitúe el cursor en el nombre del nodo y haga clic en el comando **XSLT/XQuery | Habilitar o deshabilitar punto de interrupción**).



Habilitar o deshabilitar punto de seguimiento (Mayús+Ctrl+F9)
Este comando (que carece de icono en la barra de herramientas) habilita o deshabilita los puntos de seguimiento existentes en el documento. La marca visual azul del punto de seguimiento se vuelve de color gris cuando se deshabilita el punto de seguimiento. El depurador no muestra el resultado de los puntos de seguimiento deshabilitados en la ventana Seguimiento. Para deshabilitar o habilitar un punto de seguimiento, haga clic con el botón derecho en el nombre del nodo y en el menú contextual seleccione **Puntos de interrupción y seguimiento | Habilitar o deshabilitar punto de seguimiento** (o sitúe el cursor en el nombre del nodo y haga clic en el comando **XSLT/XQuery | Habilitar o deshabilitar punto de seguimiento**).

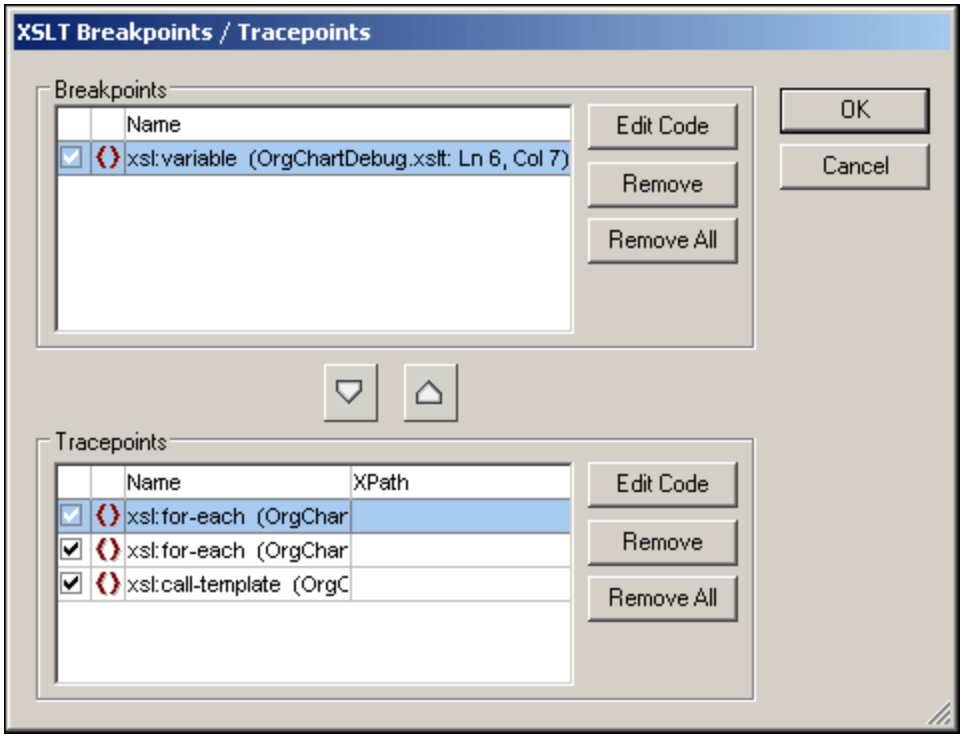


Finalizar la sesión del depurador
Finaliza la sesión de depuración y XMLSpy vuelve a la vista que estaba activa antes de iniciarse la sesión de depuración. Dependiendo de las opciones de configuración del diálogo Configurar depuración 537, los documentos de salida de la sesión de depuración seguirán abiertos o no.



Puntos de interrupción y seguimiento XSL
Este comando abre el cuadro de diálogo "Puntos de interrupción y seguimiento XSLT/XQuery", que incluye una lista con todos los puntos de interrupción y seguimiento (incluso los que están deshabilitados) presentes en todos los archivos de la sesión de depuración actual.

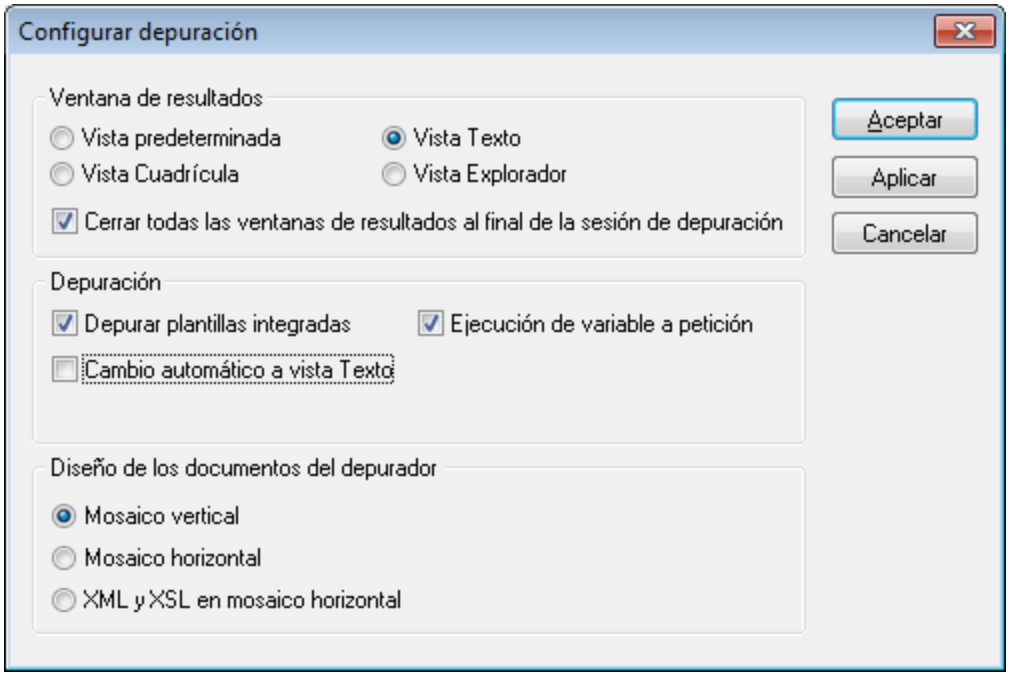
© 2016-2022 Altova GmbH



Las casillas de verificación indican si el punto de interrupción o seguimiento está habilitado (casilla marcada) o deshabilitado. Para eliminar un punto de interrupción o seguimiento, selecciónelo y haga clic en el botón **Quitar**. Para quitar todos los puntos de interrupción o seguimiento, haga clic en el botón **Quitar todos**. El botón **Editar código** le lleva al punto de interrupción o seguimiento en el archivo.

9.1.3 Configurar el depurador

El cuadro de diálogo de configuración del depurador XSLT y XQuery sirve para configurar la ventana de resultados y de depuración de todas las sesiones de depuración. Para abrir este cuadro de diálogo, seleccione el comando **XSL/XQuery | Configurar depuración** o haga clic en el icono  de la barra de herramientas. A continuación describimos las opciones de configuración del diálogo.



Ventana de resultados

Configura la vista de la ventana del archivo de resultados (predeterminada, Texto, Cuadrícula o Explorador). La vista predeterminada es la vista seleccionada para el tipo de archivo (que viene dado por su extensión de archivo: `.xslt` o `.xq` por ejemplo) en la sección *Tipos de archivo* del cuadro de diálogo "Opciones" (Herramientas | Opciones[1583]). Para las transformaciones XSLT, el tipo de archivo de resultados se define en el archivo XSLT. Para las ejecuciones de XQuery, el tipo de archivo de resultados viene dado por el formato de serialización elegido en el cuadro de opciones XQuery de este cuadro de diálogo (ver más adelante).

La opción *Cerrar todas las ventanas de resultados* le da la oportunidad de mantener abiertas las ventanas de los documentos de resultados de la sesión de depuración cuando esta finalice.

Depuración

La opción *Depurar plantillas integradas* sirve para que el depurador recorra el código de las plantillas integradas **paso a paso por instrucciones**. Esta opción no está relacionada con la **visualización** de las plantillas integradas cuando se hace clic en este tipo de plantillas en la pestaña Plantillas de la ventana de información ni cuando la pila de llamadas muestra un nodo del archivo de plantilla integrado.

El depurador XSLT sólo funciona en las vistas Texto o Cuadrícula. Si marca la casilla *Cambio automático a la vista Texto*, se cambiará automáticamente a la vista Texto de un documento si este no está en la vista Texto o Cuadrícula. (Tenga en cuenta que el depurador XQuery sólo funciona en la vista Texto.) Si marca la casilla *Ejecución de variable a petición,* el depurador recorrerá la definición de variable paso a paso por instrucciones cuando se llame a la variable. De lo contrario, el depurador no recorrerá la definición de variable paso a paso por instrucciones cuando encuentre una llamada a una variable, sino que continuará con el paso siguiente.

© 2016-2022 Altova GmbH

Diseño de los documentos del depurador

Los documentos del depurador son los documentos que están abiertos en el depurador. Puede elegir si estos documentos se organizan en un mosaico horizontal, en un mosaico vertical o si el XML y XSLT se organizan en un mosaico horizontal y el documento de resultados aparece en vertical junto a ellos.

9.1.4 Iniciar una sesión de depuración

La manera más sencilla de iniciar una sesión de depuración es desde un archivo XSLT, XQuery o XML. Si el archivo asociado necesario (véase la tabla de archivos 531) ya se asignó al archivo activo, la sesión de depuración se inicia inmediatamente. De lo contrario, se solicita al usuario que seleccione el archivo asociado que desea utilizar. Los archivos XQuery no requieren ni contienen una asociación de archivo XML y, por tanto, puede cambiar las opciones de configuración para que se solicite o no la asociación de archivo cada vez que se inicie una sesión de depuración XQuery desde un archivo XQuery.

Asociaciones predefinidas

Las asociaciones predefinidas son relevantes para las sesiones de depuración XSLT y se refieren a los casos en los que la asignación de archivo asociado ya está presente en el archivo activo. Para realizar una asignación en un archivo XML o XSLT:

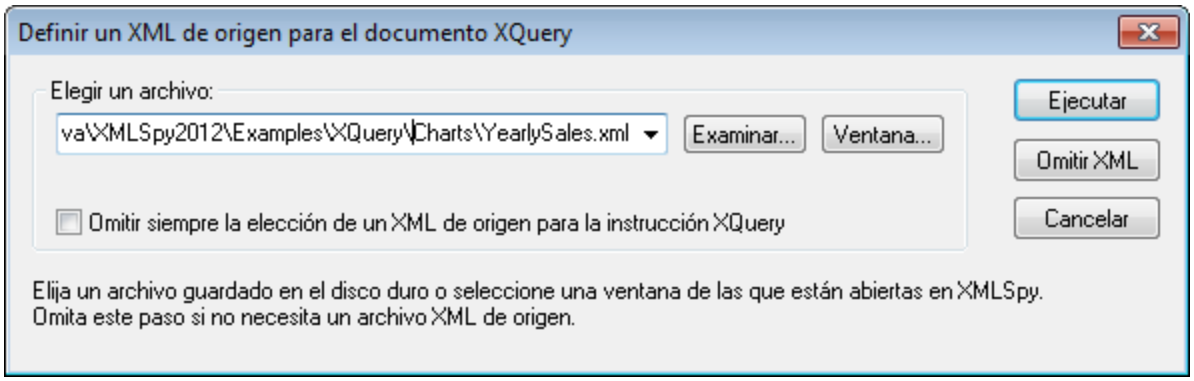
- Abra el archivo XML, haga clic en **XSL/XQuery | Asignar XSL** y seleccione el archivo XSLT.
- Abra el archivo XSLT, haga clic en **XSL/XQuery | Asignar archivo XML de muestra...** y seleccione el archivo XML.

Al hacer clic en **XSL/XQuery | Iniciar el depurador** se inicia directamente la sesión de depuración, sin solicitar el archivo asociado al archivo activo.

Asignación directa

Si el archivo activo no tiene una asociación predefinida, se solicita una asociación. En ese caso lo que ocurre al seleccionar **XSL/XQuery | Iniciar el depurador** es esto:

- Para los archivos XML se solicita al usuario que seleccione un archivo XSLT, XQuery o XQuery Update.
- Para los archivos JSON se solicita al usuario que seleccione un archivo XSLT, XQuery o XQuery Update.
- Para los archivos XSLT se solicita al usuario que seleccione un archivo XML.
- Para los archivos XQuery el usuario tiene la opción de seleccionar un archivo XML o no.



(El diálogo de la imagen aparece al iniciar una sesión de depuración cuando el archivo activo es un archivo XQuery.)

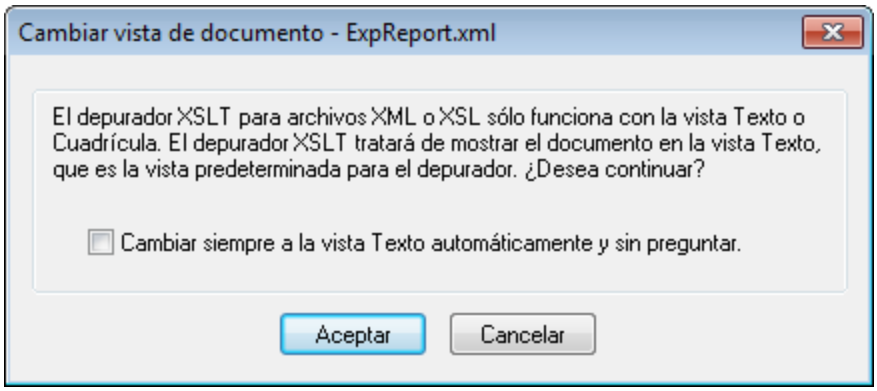
La sesión de depuración se inicia después de seleccionar el archivo asociado (o después de saltarse este paso, si la asociación es opcional).

Método alternativo de asociar un archivo

Puede realizar asociaciones predefinidas en el cuadro de diálogo Propiedades de proyecto. Haga clic en **Proyecto | Propiedades** y asigne los archivos necesarios. Para ello, marque la casilla **Utilizar este XSL** / **Utilizar este XML**.

Vista del depurador

El depurador XSLT y XQuery sólo funciona en las vistas Texto o Cuadrícula. Si el archivo XML o XSLT no está abierto en la vista Texto o Cuadrícula o si el archivo XML está asociado a un archivo SPS, entonces aparece el cuadro de diálogo de la imagen siguiente al iniciar la sesión de depuración.



Al hacer clic en **Aceptar** el documento se abre en la vista Texto. Recuerde que los archivos XQuery siempre se presentan en la vista Texto.

© 2016-2022 Altova GmbH

9.1.5 Ventanas de información

Las ventanas de información que se abren en la interfaz del depurador XSLT y XQuery durante una sesión de depuración ofrecen información sobre los aspectos de la transformación XSLT o la ejecución de XQuery. Esta información es importante porque le ayuda a depurar sus archivos XSLT o XQuery.

Las sesiones de depuración XSLT incluyen ocho ventanas de información, mientras que las sesiones de depuración XQuery incluyen cinco ventanas. Estas ventanas están organizadas por defecto en dos grupos y están situadas en la parte inferior de la interfaz del depurador XSLT y XQuery (*ver la imagen siguiente*). En esta sección describimos estas ventanas y la información que ofrecen.



Este es el diseño predeterminado de la interfaz del depurador XSLT y XQuery.

El primer grupo de ventanas de información consta de tres pestañas:

- Contexto 542 (para sesiones de depuración XSLT)
- Variables 542
- Inspección XPath 543

El segundo grupo de ventanas de información consta de cinco pestañas:

- Pila de llamadas 544
- Mensajes 544
- Plantillas 545 (para sesiones de depuración XSLT)
- Información 546
- Seguimiento 546

Por tanto, en el diseño predeterminado de la interfaz del depurador las ventanas están agrupadas en dos bloques, cada uno con una pestaña por cada ventana de información. Si, por ejemplo, desea ver la información sobre variables en el primer bloque de ventanas, haga clic en la pestaña **Variables**. Como resultado se muestra la ventana Variables y las ventanas Contexto e Inspección XPath se ocultan. Tenga en cuenta que en algunas pestañas puede usar la información como herramientas de navegación: al hacer clic en un elemento

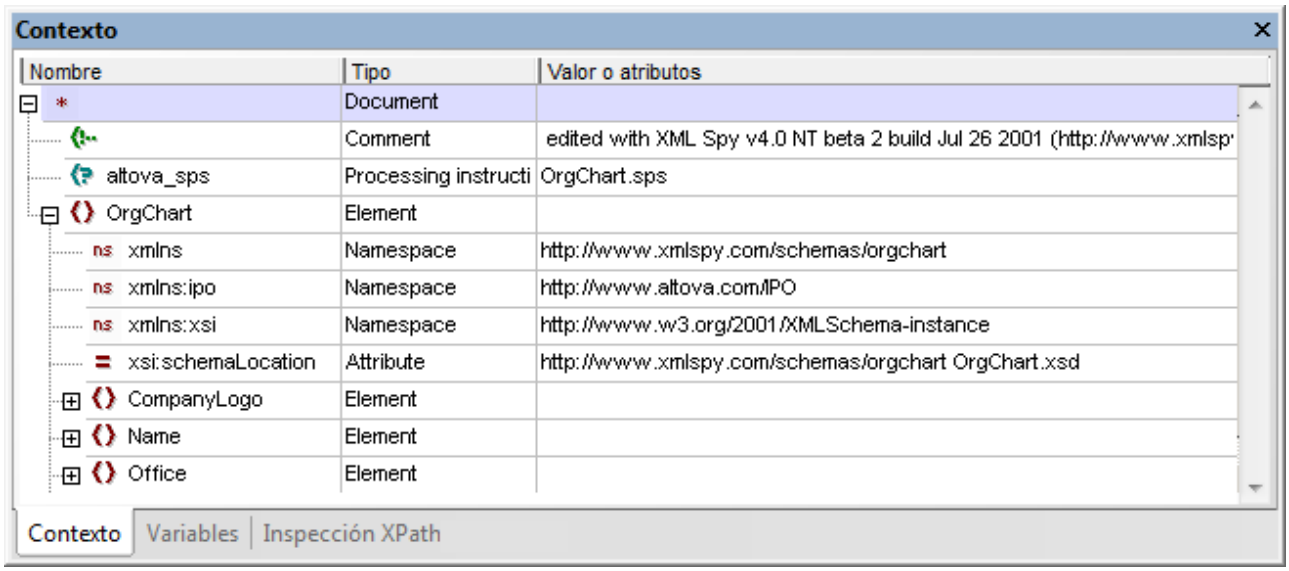
de la ventana, dicho elemento se resalta en el archivo XML, XSLT o XQuery. Para más información consulte la documentación de cada ventana (Contexto 542, Pila de llamadas 544, Plantillas 545).

Para ajustar el tamaño de los grupos de ventanas de información basta con arrastrar los bordes del grupo. También puede sacar las ventanas del grupo. Para ello haga clic en el nombre de la pestaña y arrastre la ventana fuera del grupo. Para añadir una ventana a un grupo, haga clic en su barra de título y arrástrela hasta la barra de título del grupo. Tenga en cuenta que no hay una opción para restaurar el diseño de las ventanas a su diseño predeterminado.

9.1.5.1 Contexto

La ventana Contexto está disponible sólo en sesiones de depuración XSLT. Las sesiones de depuración XQuery no incluyen esta ventana.

Mientras se procesa la hoja de estilos XSLT, el contexto del procesador está siempre dentro de una plantilla que coincide con cierta secuencia (de nodos o valores atómicos). La ventana Contexto muestra el contexto actual del procesamiento, que puede ser una secuencia de nodos, un único nodo o un valor atómico (como una cadena de texto, por ejemplo). Dependiendo del tipo de elemento de contexto, la ventana muestra su valor o atributo/s. Por ejemplo, si el elemento de contexto es de tipo "elemento", aparecen los atributos del elemento. Si el elemento de contexto es de tipo "atributo" o "texto", la ventana muestra el valor del nodo.

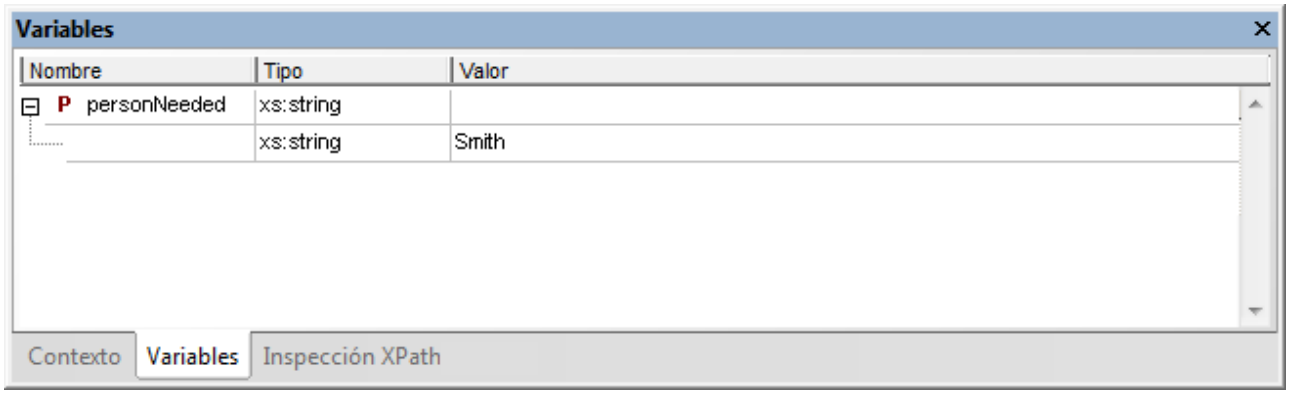


Al hacer clic en una entrada de la ventana Contexto, el elemento se resalta en el documento XML. Si el documento XML no aparece en la interfaz del depurador, se abrirá una ventana con el documento XML.

9.1.5.2 Variables

La ventana Variables está disponible en las sesiones de depuración XSLT y XQuery. Esta ventana muestra las variables y los parámetros que se utilizan en el documento XSLT/XQuery cuando están en su ámbito de aplicación, así como sus valores.

© 2016-2022 Altova GmbH

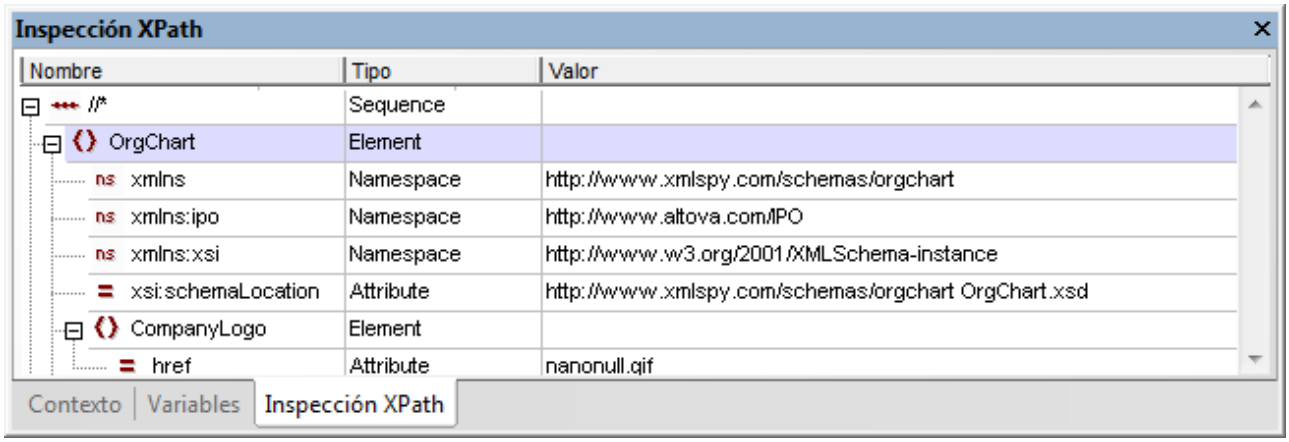


Los parámetros se identifican por la marca P, las variables globales (declaradas en el nivel superior de una hoja de estilos) se identifican por la marca G y las variables locales (declaradas dentro de una plantilla XSLT) se identifican por la marca L. El tipo de los valores de las variables y los parámetros también se identifican con diferentes iconos en el campo Valor. Los distintos tipos de valores son: grupo de nodos, fragmento de nodo, cadena, número y booleano.

9.1.5.3 Inspección XPath

La ventana Inspección XPath está disponible en sesiones de depuración XSLT y XQuery.

En esta ventana puede introducir las expresiones XPath que desee evaluar en un contexto o en varios. A medida que recorre el documento XSLT paso a paso, la expresión XPath se evalúa en el contexto actual y el resultado aparece en la columna Valor.



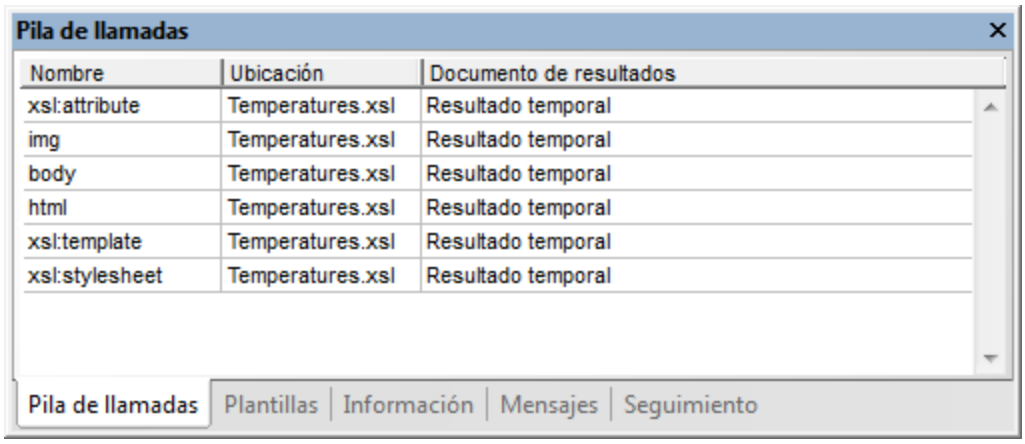
Para introducir una expresión XPath, haga doble clic en el campo de texto situado bajo la columna Nombre y escriba la XPath. También puede arrastrar una expresión desde un archivo y colocarla en la ventana Inspección XPath. Utilice expresiones que sean correctas en función de la versión de XPath que corresponda a la versión XSLT de la hoja de estilos (XPath 1.0 para XSLT 1.0, XPath 2.0 para XSLT 2.0 y XPath 3.0 para XSLT 3.0).

Nota: si se usaron espacios de nombres en el archivo XML o XSLT, es necesario usar los prefijos de espacio de nombres correctos en sus expresiones XPath.

9.1.5.4 Pila de llamadas

La ventana Pila de llamadas está disponible en las sesiones de depuración XSLT y XQuery.

La ventana Pila de llamadas ofrece una lista de las plantillas e instrucciones XSLT procesadas hasta el momento. Las primeras instrucciones y plantillas de la lista son las procesadas más recientemente.



Al hacer clic en una entrada de la ventana, la instrucción o plantilla XSLT seleccionada se resalta en la ventana del documento XSLT. Al hacer clic en una instrucción o plantilla que hace referencia a una plantilla integrada, la plantilla integrada aparece resaltada en una ventana separada que incluye todas las plantillas integradas.

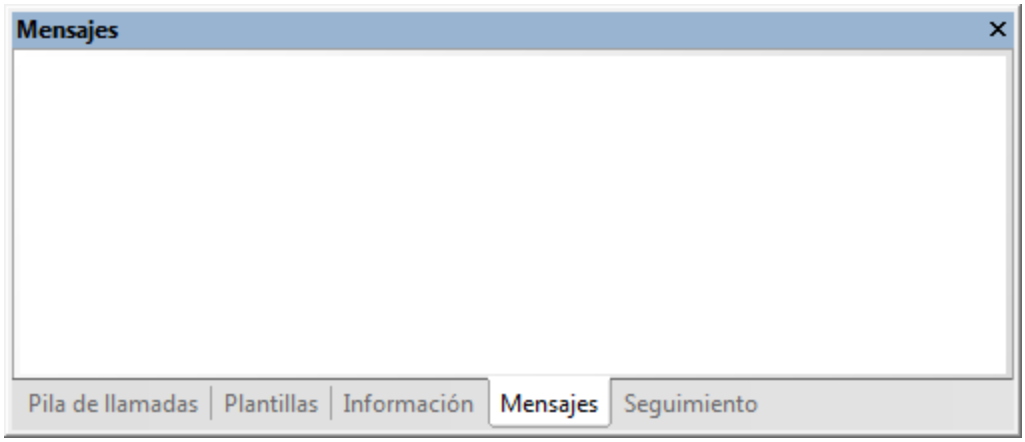
9.1.5.5 Mensajes

La ventana Mensajes está disponible en las sesiones de depuración XSLT y XQuery.

XSLT 1.0, XSLT 2.0 y XSLT 3.0

En las sesiones de depuración XSLT, la ventana Mensajes muestra mensajes de error, las instrucciones `xsl:message` y otros mensajes de error relacionados con el proceso de depuración.

© 2016-2022 Altova GmbH



XQuery

En las sesiones de depuración XQuery, la ventana Mensajes muestra mensajes de error.

9.1.5.6 Plantillas

La ventana Plantillas (*imagen siguiente*) está disponible sólo en las sesiones de depuración XSLT. Las sesiones de depuración XQuery no ofrecen esta ventana.

La ventana Plantillas muestra las plantillas utilizadas en la hoja de estilos XSLT, incluidas las plantillas integradas y las plantillas con nombre. Las plantillas coincidentes aparecen en la ventana con el nombre del nodo con el que coinciden, mientras que las plantillas con nombre aparecen con su nombre. La ventana también incluye el modo, prioridad y ubicación de la plantilla, sea del tipo que sea.

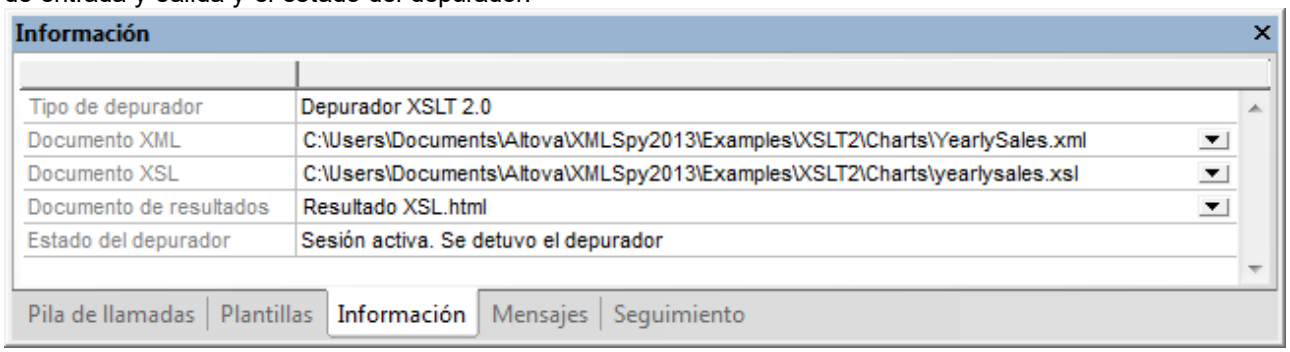


En la imagen anterior se observan tres tipos de plantillas coincidentes en la hoja de estilos XSLT: una plantilla que coincide (*Match*) con el nodo de documento `/` y dos plantillas que coinciden (*Match*) con los nodos `n1:italic` y `n1:bold`. Las demás plantillas son plantillas integradas (tal y como indica el campo Ubicación).

Al hacer clic en una entrada de esta ventana, la plantilla se resalta en la ventana del documento XSLT. Si hace clic en una plantilla integrada, la plantilla se resalta en una ventana separada que incluye todas las plantillas integradas.

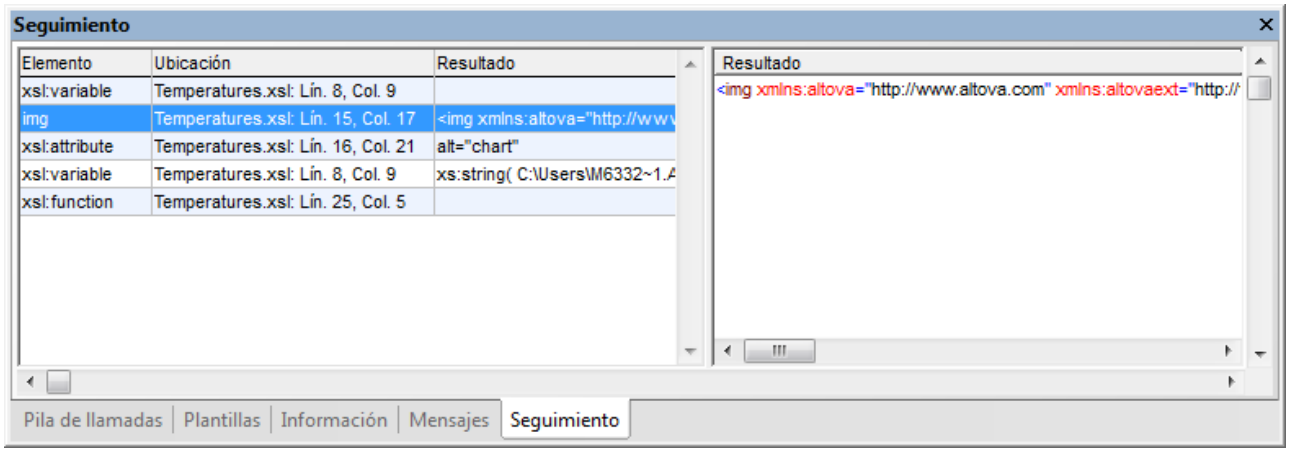
9.1.5.7 Información

La ventana Información está disponible en las sesiones de depuración XSLT y XQuery. Esta ventana ofrece metadatos sobre la sesión de depuración actual: qué depurador se está usando, el nombre de los documentos de entrada y salida y el estado del depurador.



9.1.5.8 Seguimiento

La ventana Seguimiento está disponible en las sesiones de depuración XSLT y XQuery.



La ventana Seguimiento muestra los elementos en los que se estableció un punto de seguimiento, la posición de los elementos en la hoja de estilos XSLT y el resultado generado cuando se ejecuta el elemento. Haga clic en una fila del panel izquierdo de la ventana para ver el resultado completo en el panel derecho de la ventana.

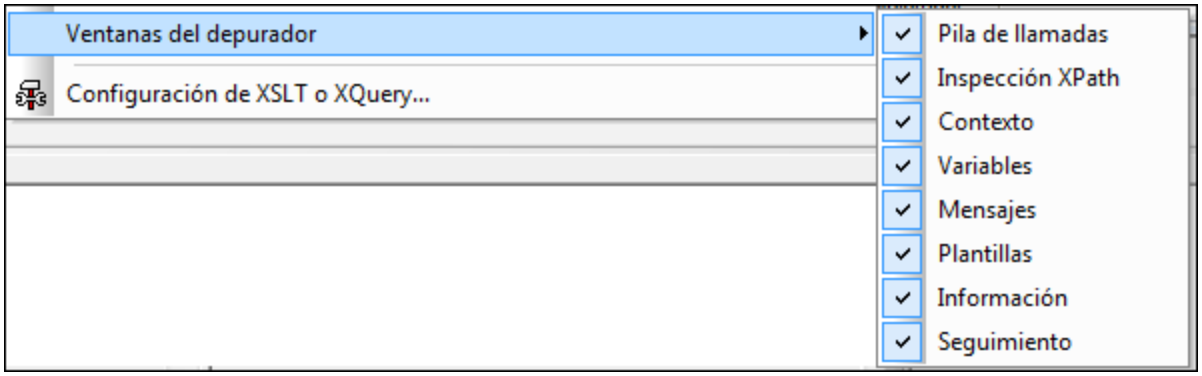
9.1.5.9 Organizar las ventanas de información

Puede cambiar la distribución de las ventanas de información en la interfaz del depurador XSLT y XQuery. Las ventanas pueden estar flotando, acopladas o agrupadas en un bloque de ventanas. Hay varias maneras de ordenar las ventanas.

© 2016-2022 Altova GmbH

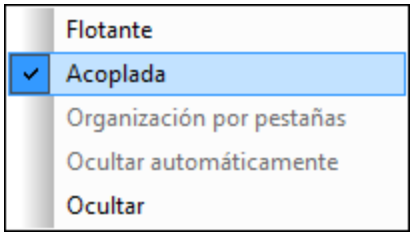
Menú

Si pasa el cursor por el elemento **Ventanas del depurador** del menú **XSL/XQuery**, aparece la lista de las ventanas de información. Para mostrar u ocultar una ventana, active o desactive la ventana en esta lista.



Menú contextual

Al hacer clic con el botón derecho en la barra de título o en la pestaña de una ventana aparece este menú contextual.



Seleccione si la ventana debe estar flotando, acoplada u oculta.

Arrastrar y colocar

Puede hacer clic en la pestaña o barra de título de una ventana y arrastrarla hasta la posición deseada.

Además puede acoplar una ventana en otra ventana o en la interfaz con ayuda de los controles de posicionamiento que aparecen cuando se arrastra una ventana:

- Cuando se arrastra una ventana sobre otra, aparece un control de posicionamiento circular (*ver imagen siguiente*). Este control se divide en cinco sectores de posicionamiento. Al colocar la ventana sobre uno de estos sectores y soltar el botón del ratón, la ventana se acopla en el sector correspondiente de la **ventana de destino**. Los cuatro sectores en forma de flecha acoplan la ventana en los flancos correspondientes de la ventana de destino. El botón central acopla la ventana en la ventana de destino en forma de pestaña. También puede acoplar una ventana en otra en forma de ventana si arrastra la ventana y la coloca sobre la barra de pestañas de la ventana de destino.



- Cuando se arrastra una ventana, aparece un control de posicionamiento que tiene cuatro flechas (*ver imagen siguiente*). Cada flecha se corresponde con un flanco de la **interfaz del depurador**. Al soltar la ventana sobre una de estas flechas, la ventana se acopla en el lado correspondiente de la interfaz del depurador.



También puede hacer doble clic en la barra de título de una ventana para ponerla en posición flotante o para acoplarla.

9.1.6 Puntos de interrupción

Con el depurador XSLT y XQuery puede definir puntos de interrupción en documentos XSLT, XQuery y XML. Los puntos de interrupción aparecen señalados con una línea discontinua roja (*ver imagen siguiente*).

Nota: se puede establecer un punto de interrupción y un punto de seguimiento en la misma instrucción. Esto se señala con una línea discontinua roja y azul (*ver imagen siguiente*).

```
<!– Local Variable reused –>
<xsl:variable name="OfficeName" select="n1:Name"/>
<!– Display the company name if the varaible is true ********* –>
<xsl:if test="$Show_Company_Name">
    <h3>
        <xsl:value-of select="$OfficeName"/>
    </h3>
    <xsl:message>
        <xsl:text>Company Named Displayed</xsl:text>
    </xsl:message>
</xsl:if>
```

Al iniciar el depurador en una sesión de depuración, el depurador se detiene en cada punto de interrupción que encuentre. De este modo puede identificar zonas que necesitan ser depuradas y centrarse en esas zonas en el documento XSLT, XQuery o XML. No hay límite en el número de puntos de interrupción que se pueden establecer.

Nota: los puntos de interrupción creados en un documento se conservan hasta cerrar el documento. Sin embargo, si cambia a la vista Esquema (en caso de usar documentos XSD), se eliminan los puntos de

© 2016-2022 Altova GmbH

interrupción. Al volver a la vista Texto o Cuadrícula (desde la vista Esquema) el documento ya no tiene puntos de interrupción.

Puntos de interrupción en documentos XML

Puede establecer puntos de interrupción en todos los nodos de un documento XML. El procesamiento se interrumpirá al principio del nodo.

Puntos de interrupción en documentos XSLT

En los documentos XSLT puede establecer puntos de interrupción:

- Al principio de plantillas e instrucciones `template` (p. ej. `xsl:for-each`).
- En una expresión XPath (XPath 1.0 o XPath 2.0).
- En cualquier nodo de un fragmento XML construido literalmente. El procesamiento se interrumpirá al principio del nodo.

Puntos de interrupción en documentos XQuery

En los documentos XQuery puede establecer puntos de interrupción:

- Al principio de las instrucciones XQuery.
- En una expresión XQuery.
- En cualquier nodo de un fragmento XML construido literalmente. El procesamiento se interrumpirá al principio del nodo.

Cómo insertar y quitar puntos de interrupción

Para insertar un punto de interrupción:

1. Coloque el cursor en el lugar del documento donde desea insertar el punto de interrupción (*según lo descrito más arriba*). En las sesiones de depuración XSLT puede establecer puntos de interrupción tanto desde la vista Texto como desde la vista Cuadrícula. Las sesiones de depuración XQuery se realizan en la vista Texto.
2. Después puede elegir entre estas tres opciones:

   - Seleccione el comando **XSL/XQuery | Insertar o quitar punto de interrupción**.
   - Pulse la tecla **F9**.
   - Haga clic con el botón derecho y en el menú contextual seleccione **Puntos de interrupción y seguimiento | Insertar o quitar punto de interrupción**.

Para quitar un punto de interrupción:

1. Coloque el cursor en el lugar del documento que contiene el punto de interrupción.
2. Después puede elegir entre estas tres opciones:

   - Seleccione el comando **XSL/XQuery | Insertar o quitar punto de interrupción**.
   - Pulse la tecla **F9**.
   - Haga clic con el botón derecho y en el menú contextual seleccione **Puntos de interrupción y seguimiento | Insertar o quitar punto de interrupción**.

También puede quitar un punto de interrupción desde el cuadro de diálogo "Puntos de interrupción y seguimiento":

1. Seleccione la opción de menú **XSL/XQuery | Puntos de interrupción y seguimiento...**.
2. Haga clic en el punto de interrupción en el cuadro de diálogo y pulse el botón **Quitar**.

El botón **Quitar todos** elimina todos los puntos de interrupción del cuadro de diálogo (y en todas las hojas de estilos XSLT).

Cómo deshabilitar y habilitar puntos de interrupción

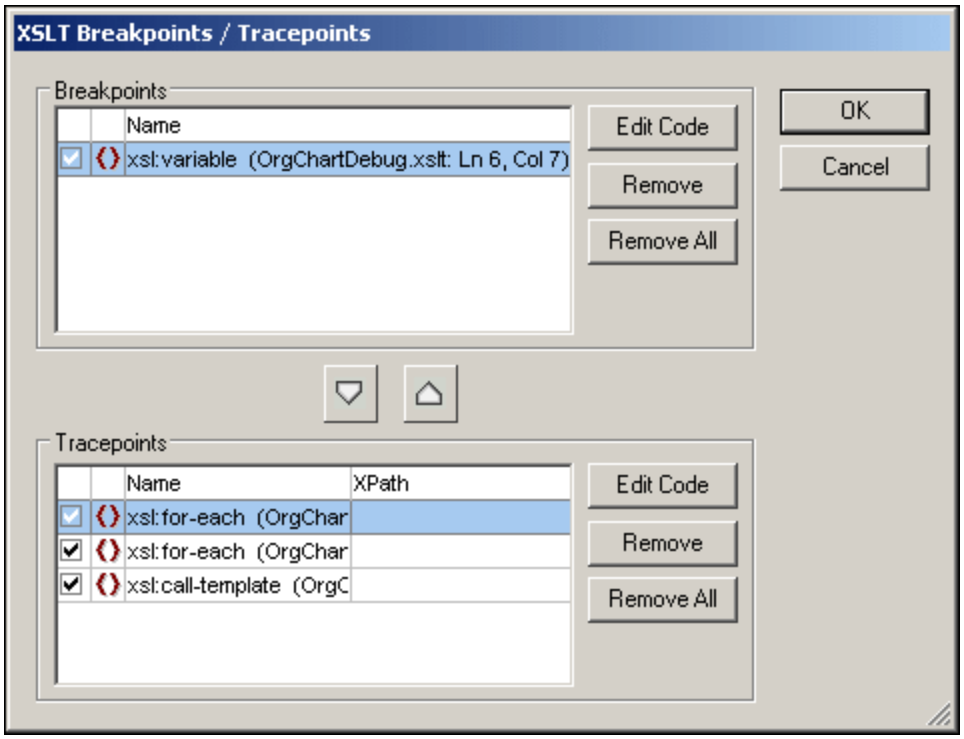
Si después de insertar puntos de interrupción desea saltarse algunos puntos pero no desea eliminarlos, puede deshabilitarlos y volverlos a habilitar cuando sea necesario.

Para deshabilitar un punto de interrupción:

1. Coloque el cursor en el nodo o en la expresión que contiene el punto de interrupción.
2. Seleccione el comando **XSL/XQuery | Habilitar o deshabilitar punto de interrupción** (o pulse **Ctrl+F9**). El punto de interrupción se vuelve de color gris, lo cual indica que está deshabilitado.

También puede deshabilitar puntos de interrupción desde el cuadro de diálogo "Puntos de interrupción y seguimiento":

1. Seleccione la opción de menú **XSL/XQuery | Puntos de interrupción y seguimiento...**. Aparece el cuadro de diálogo "Puntos de interrupción y seguimiento", que muestra todos los puntos de interrupción presentes en todos los archivos XML y XSLT que están abiertos.



2. Desactive las casillas de los puntos de interrupción que desea deshabilitar y haga clic en **Aceptar**. Los puntos de interrupción se vuelven de color gris, lo cual indica que están deshabilitados.

 © 2016-2022 Altova GmbH

Para habilitar un punto de interrupción:

1. Coloque el cursor en el nodo o en la expresión que contiene el punto de interrupción.
2. Seleccione **XSL/XQuery | Habilitar o deshabilitar punto de interrupción** (o pulse **Ctrl+F9**). El punto de interrupción se vuelve de color rojo, lo cual indica que está habilitado.

Búsqueda de puntos de interrupción

Para buscar un punto de interrupción determinado:

1. Seleccione la opción de menú **XSL/XQuery | Puntos de interrupción y seguimiento...**. Aparece el cuadro de diálogo "Puntos de interrupción y seguimiento".
2. Haga clic en el punto de interrupción que desea buscar.
3. Haga clic en el botón **Editar código**. Se cierra el cuadro de diálogo "Puntos de interrupción y seguimiento" y el cursor se sitúa directamente delante del punto de interrupción seleccionado en la vista Texto. En la vista Cuadrícula la celda de la tabla que contiene el punto de interrupción aparece resaltada en rojo.

Continuar con la depuración después de un punto de interrupción

Para continuar con la depuración después de un punto de interrupción:

- Seleccione el comando **XSL/XQuery | Paso a paso por instrucciones** o el comando **XSL/XQuery | Iniciar el depurador**.

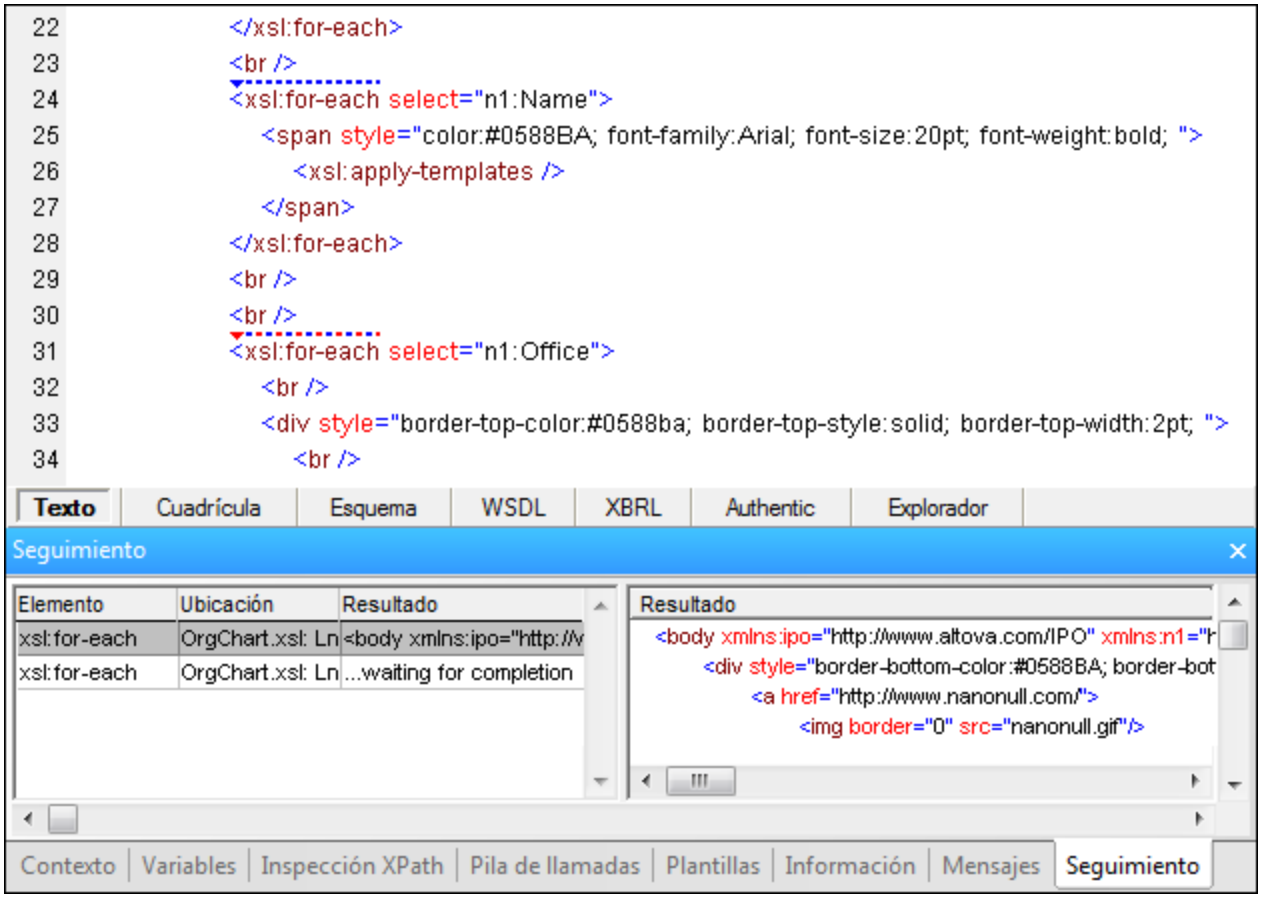
9.1.7 Puntos de seguimiento

Con el depurador XSLT y XQuery puede definir puntos de seguimiento en documentos XSLT.

Los puntos de seguimiento sirven para seguir el contenido generado por una instrucción o para ver el resultado de una expresión XPath en el punto donde se estableció el punto de seguimiento, sin necesidad de editar la hoja de estilos XSLT con el elemento `xsl:message`, por ejemplo, para generar mensajes de depuración.

En las hojas de estilos XSLT los puntos de seguimiento se señalan con una línea discontinua azul (*ver imagen siguiente*).

Nota: se puede establecer un punto de seguimiento y un punto de interrupción en la misma instrucción. Esto se señala con una línea discontinua azul y roja (*ver imagen siguiente*).



El depurador reproduce el contenido generado por cada instrucción que incluye un punto de seguimiento. Los resultados aparecen en la ventana Seguimiento. No hay límite en el número de puntos de seguimiento que se pueden establecer en una hoja de estilos XSLT.

Nota: los puntos de seguimiento creados en un documento se conservan hasta cerrar el documento.

Puntos de seguimiento en documentos XSLT

En las hojas de estilos XSLT puede establecer puntos de seguimiento en instrucciones XSLT y resultados literales.

Puntos de seguimiento en documentos XML y XQuery

Puede establecer puntos de seguimiento en documentos XML y XQuery.

Cómo insertar y quitar puntos de seguimiento

Para insertar un punto de seguimiento:

1. Coloque el cursor en el lugar del documento XSLT donde desea insertar el punto de seguimiento. Durante la sesión de depuración puede insertar puntos de seguimiento tanto desde la vista Texto como desde la vista Cuadrícula.

© 2016-2022 Altova GmbH

2. Después puede elegir entre estas tres opciones:

   - Seleccione el comando **XSL/XQuery | Insertar o quitar punto de seguimiento**.
   - Pulse **Mayús+F9**.
   - Haga clic con el botón derecho y en el menú contextual seleccione **Insertar o quitar punto de seguimiento**.

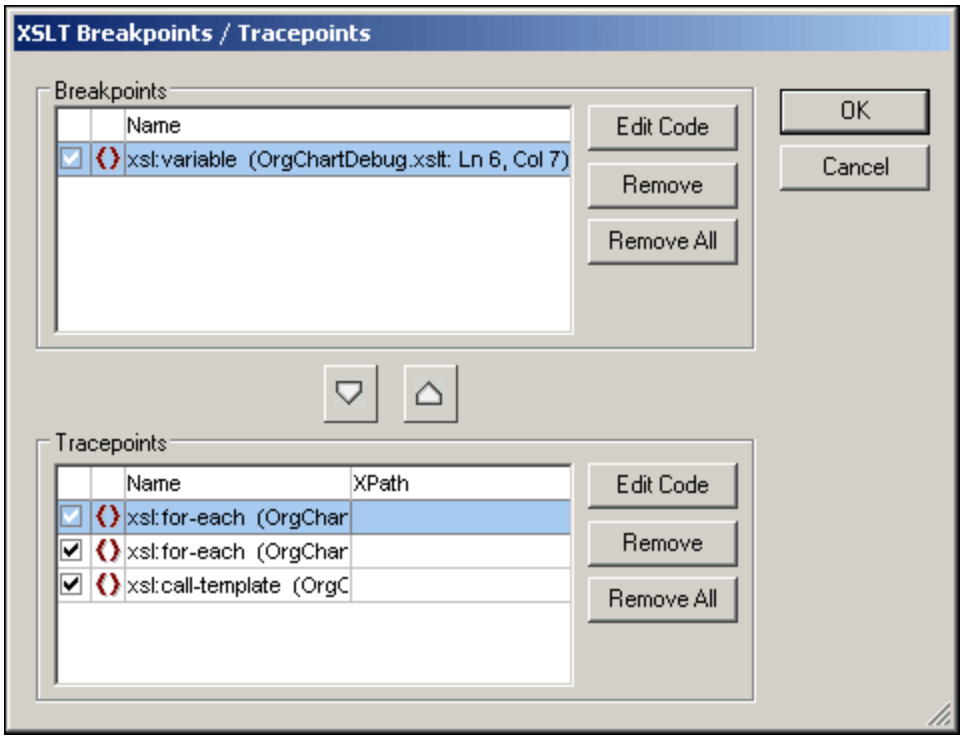
Para quitar un punto de seguimiento:

1. Coloque el cursor en el lugar del documento XSLT que contiene el punto de seguimiento.
2. Después puede elegir entre estas tres opciones:

   - Seleccione el comando **XSL/XQuery | Insertar o quitar punto de seguimiento**.
   - Pulse **Mayús+F9**.
   - Haga clic con el botón derecho y en el menú contextual seleccione **Insertar o quitar punto de seguimiento**.

También puede quitar puntos de seguimiento desde el cuadro de diálogo "Puntos de interrupción y seguimiento XSLT":

1. Seleccione la opción de menú **XSL/XQuery | Puntos de interrupción y seguimiento...**.
2. En el cuadro de diálogo que aparece haga clic en el punto de seguimiento que desea quitar (*ver imagen siguiente*) y pulse el botón **Quitar**.



El botón **Quitar todos** elimina todos los puntos de seguimiento que aparecen en el cuadro de diálogo (y de todas las hojas de estilos XSLT).

Establecer una XPath para un punto de seguimiento

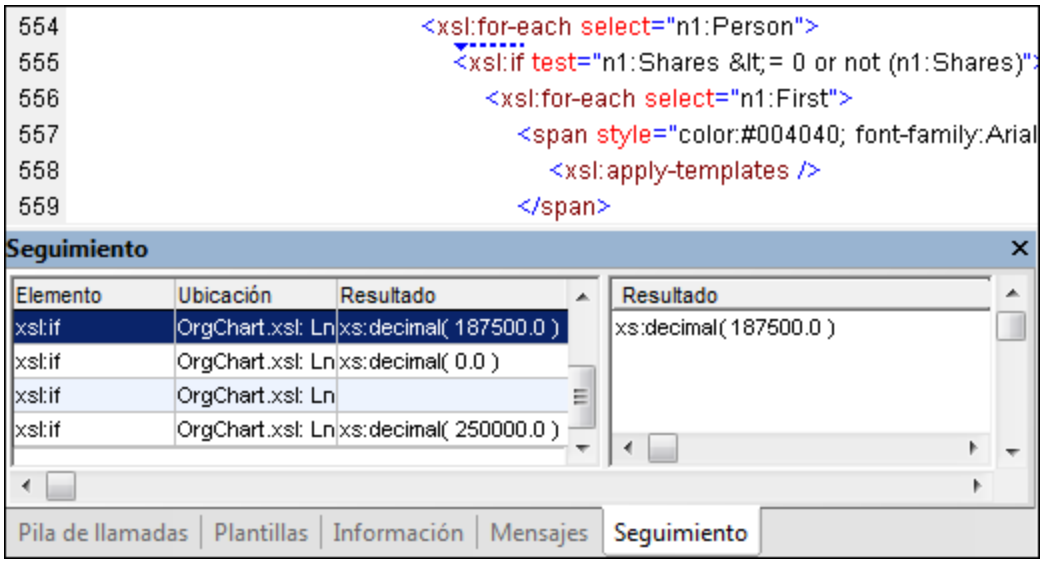
Puede establecer una XPath para un punto de seguimiento. Cuando se establece una XPath para un punto de seguimiento, el resultado de la evaluación de la XPath aparece en la ventana Seguimiento, que normalmente muestra el contenido generado por la instrucción en la que se estableció el punto de seguimiento. La XPath se evalúa en relación al nodo de contexto del punto donde se estableció el punto de seguimiento.

Para establecer una XPath para un punto de seguimiento:

1. Seleccione la opción de menú **XSL/XQuery | Puntos de interrupción y seguimiento...**. Aparece el cuadro de diálogo "Puntos de interrupción y seguimiento XSLT", que muestra los puntos de seguimientos presentes en todas las hojas de estilos XSLT que están abiertas.
2. Escriba la XPath en la columna XPath del punto de seguimiento pertinente.

Ejemplo

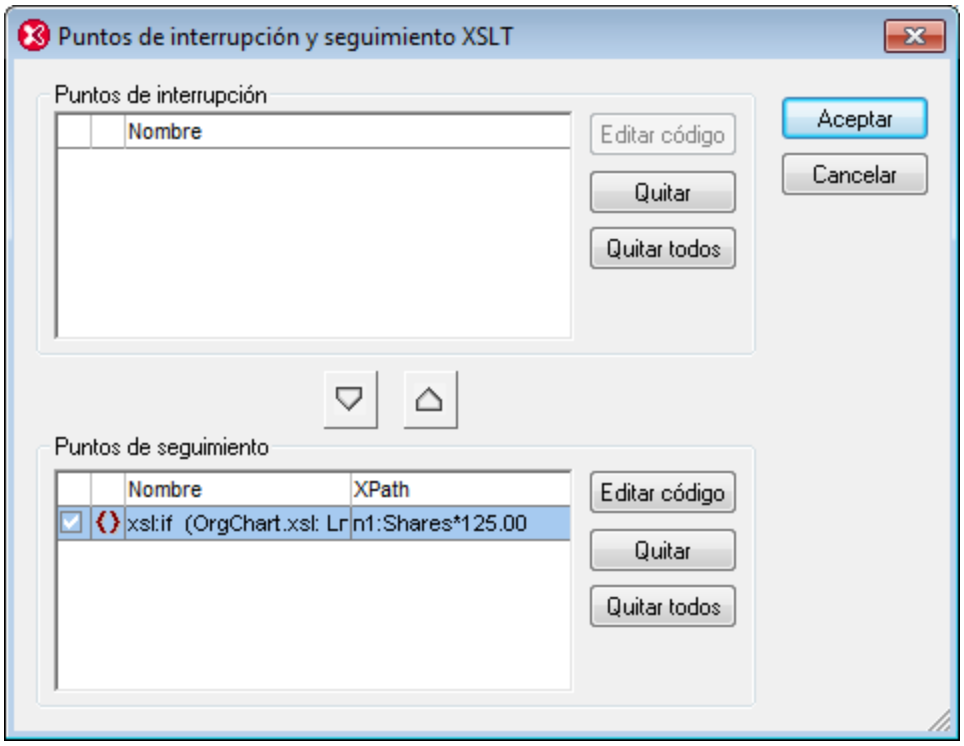
En el ejemplo que aparece a continuación, el punto de seguimiento se estableció con el nodo `Person` como nodo de contexto. El elemento `Person` contiene el elemento `Shares`. En este ejemplo queremos ver el número de acciones (Shares) que tiene cada persona multiplicado por 125 (el valor de cada acción).



Para ello siga estos pasos:

1. Establezca un punto de seguimiento en la línea inmediatamente posterior a la instrucción `xsl:for-each` que selecciona el elemento `n1:Person` (es decir, en la línea 555 de la imagen anterior).
2. Abra el cuadro de diálogo "Puntos de interrupción y seguimiento XSLT" y escriba la XPath `n1:Shares*125.00` para el punto de seguimiento creado en el paso anterior.

© 2016-2022 Altova GmbH

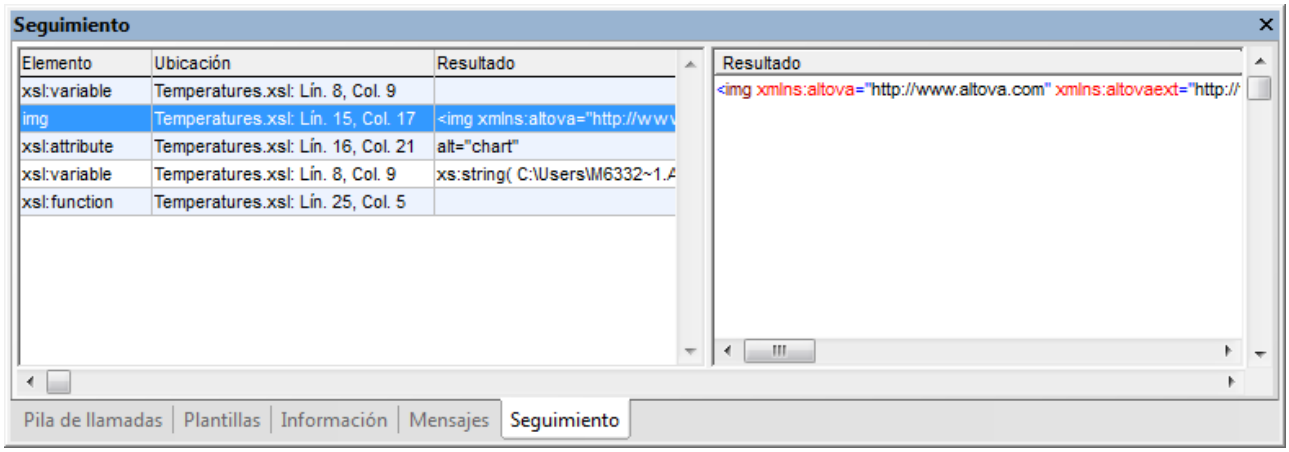


3. Inicie el depurador. Los resultados de la XPath que escribió para el punto de seguimiento aparecen ahora en la ventana Seguimiento.

Ventana Seguimiento

Seleccione el comando **XSL/XQuery | Iniciar el depurador** para empezar el proceso de depuración. El resultado de las instrucciones para las que se establecieron puntos de seguimiento aparece en la ventana Seguimiento 546 (*ver imagen siguiente*). Haga clic en una fila de la ventana Seguimiento para ver el resultado completo de la instrucción en el panel derecho de la ventana (*ver imagen*).

Nota: los resultados no aparecen en la ventana Seguimiento hasta que se finaliza la instrucción que incluye el punto de seguimiento.



Cómo deshabilitar y habilitar puntos de seguimiento

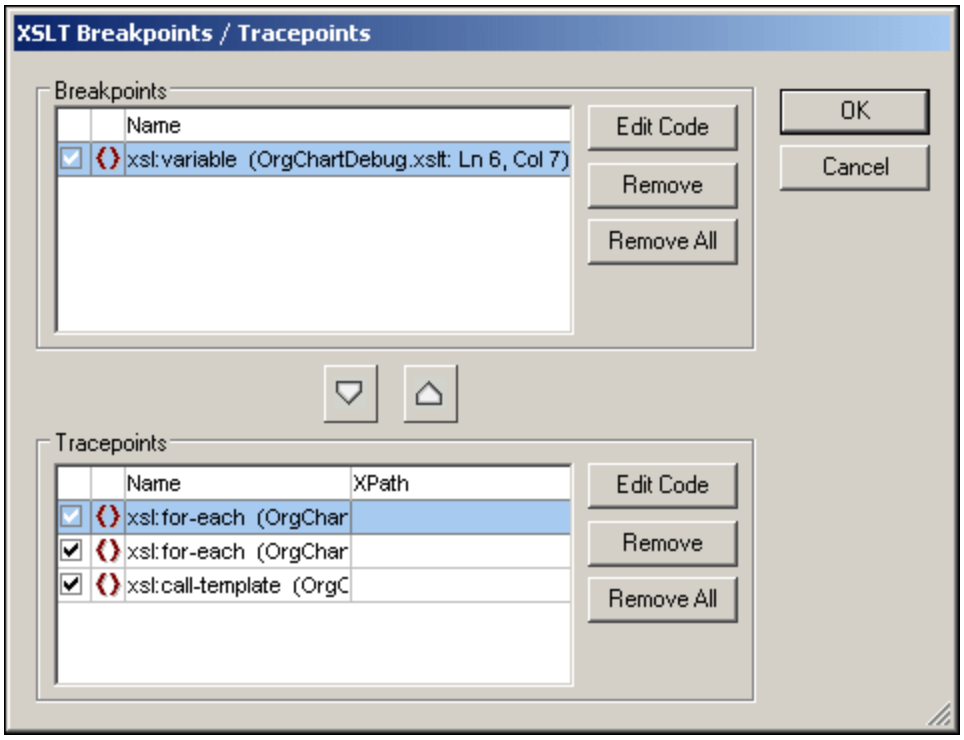
Si después de insertar puntos de seguimiento desea saltarse algunos puntos pero no desea eliminarlos, puede deshabilitarlos y volverlos a habilitar cuando sea necesario.

Para deshabilitar un punto de seguimiento:

1. Coloque el cursor en el punto de la hoja de estilos XSLT que contiene el punto de seguimiento.
2. Seleccione el comando **XSL/XQuery | Habilitar o deshabilitar punto de seguimiento** (o pulse **Ctrl+Mayús+F9**). El punto de seguimiento se vuelve de color gris, lo cual indica que está deshabilitado.

También puede deshabilitar puntos de seguimiento desde el cuadro de diálogo "Puntos de interrupción y seguimiento XSLT":

1. Seleccione la opción de menú **XSL/XQuery | Puntos de interrupción y seguimiento...**. Aparece el cuadro de diálogo "Puntos de interrupción y seguimiento XSLT", que incluye los puntos de seguimiento presentes en todas las hojas de estilos XSLT que están abiertas.



2. Desactive la casilla de verificación de los puntos de seguimiento que desea deshabilitar y haga clic en **Aceptar**. Los puntos de seguimiento se vuelven de color gris, lo cual indica que están deshabilitados.

Para volver a habilitar un punto de seguimiento:

1. Coloque el cursor en el punto del documento XSLT que contiene el punto de seguimiento.
2. Seleccione el comando **XSL/XQuery | Habilitar o deshabilitar punto de seguimiento** (o pulse **Ctrl+Mayús+F9**). El punto de seguimiento se vuelve de color azul, lo cual indica que está habilitado.

© 2016-2022 Altova GmbH

Cómo buscar puntos de seguimiento

Para buscar un punto de seguimiento determinado:

1. Seleccione la opción de menú **XSL/XQuery | Puntos de interrupción y seguimiento...**. Aparece el cuadro de diálogo "Puntos de interrupción y seguimiento XSLT".
2. Haga clic en el punto de seguimiento que desea buscar.
3. Haga clic en el botón **Editar código**. Se cierra el cuadro de diálogo "Puntos de interrupción y seguimiento XSLT" y el cursor se sitúa directamente delante del punto de seguimiento seleccionado en la vista Texto del documento XSLT. En la vista Cuadrícula la celda de la tabla que contiene el punto de interrupción aparece resaltada en azul.

9.1.8 Teclas de acceso rápido

A continuación puede ver las teclas de acceso rápido predeterminadas para los comandos más frecuentes del depurador.

Comandos del depurador

F9	Insertar o quitar punto de interrupción
F9 + Mayús	Insertar o quitar punto de seguimiento
F9 + CTRL	Habilitar o deshabilitar punto de interrupción
F9 + Mayús + CTRL	Habilitar o deshabilitar punto de seguimiento
F11	Paso a paso por instrucciones
F11 + Mayús	Paso a paso para salir
F11 + CTRL	Paso a paso por procedimientos
F11 + Alt	Iniciar el depurador

9.2 Generador de perfiles XSLT y XQuery

Sitio web de Altova: Generador de perfiles XSLT, Generador de perfiles XQuery
El generador de perfiles XSLT/XQuery es una herramienta que sirve para analizar los tiempos de ejecución de las hojas de estilos XSLT (1.0 o 2.0) y de documentos XQuery desde XMLSpy. El generador informa del tiempo que tarda cada instrucción de la hoja de estilos XSLT o del documento XQuery en ejecutarse. Esta información puede ser muy útil a la hora de optimizar el tiempo de ejecución de estos archivos.

El generador de perfiles se utiliza para buscar las instrucciones que presentan el mayor tiempo de ejecución total y poder optimizarlas. Si una instrucción tiene un tiempo de ejecución alto es porque:

- la instrucción requiere demasiado tiempo
- la instrucción se evalúa demasiado a menudo (tiene un recuento alto)
- ambas razones

Generación de perfiles con contador y con gráfico de llamadas

El generador de perfiles puede generar perfiles con *contador* o con *gráfico de llamadas*. Ambos tipos de generación de perfiles muestran estadísticas sobre el tiempo de ejecución de cada instrucción.

Si desea optimizar una instrucción, lo normal es usar la generación de perfiles con contador, que muestra una línea en la ventana del generador de perfiles por cada instrucción.

La generación de perfiles con gráfico de llamadas muestra el historial de ejecución completo de una transformación XSLT o una ejecución de XQuery. Es decir, muestra a qué plantillas o funciones se llamó y en qué orden se les llamó durante la transformación. En los resultados de la generación de perfiles con gráfico de llamadas se muestra una línea por cada vez que se llamó a una instrucción, en lugar de una línea por cada instrucción.

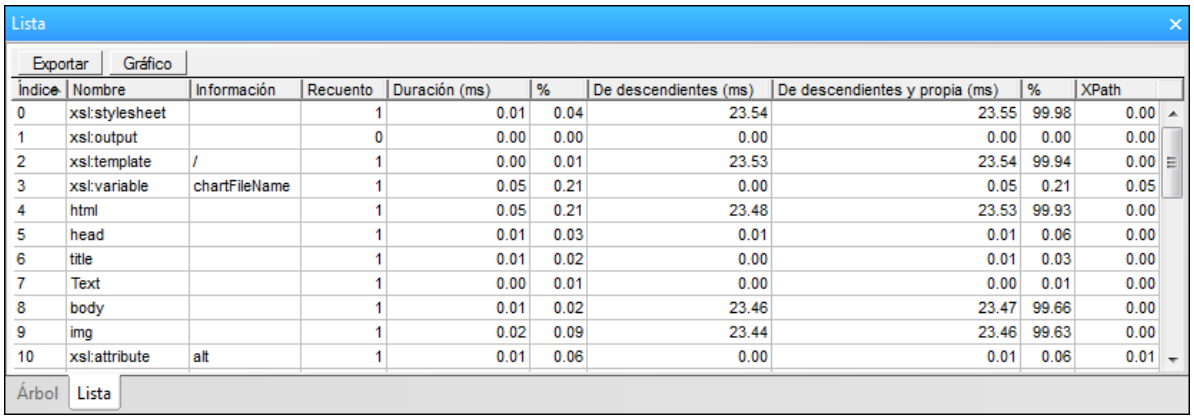
Para usar el generador de perfiles XSLT/XQuery, consulte los apartados Generación de perfiles XSLT 563 y Generación de perfiles XQuery 567.

Vistas del generador de perfiles

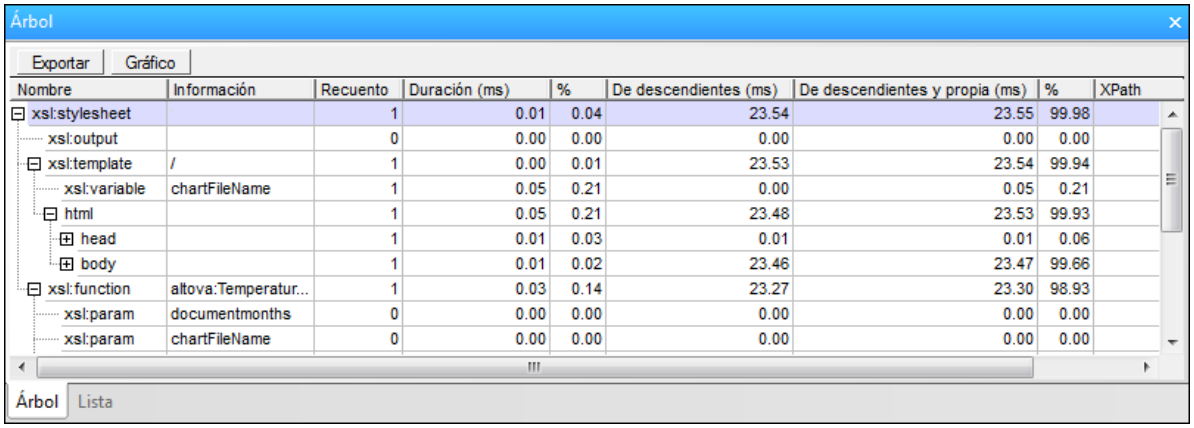
Los resultados del análisis se pueden ver en dos vistas diferentes, dependiendo de la pestaña seleccionada en la ventana de resultados:

- **Vista en forma de lista:** las estadísticas de la generación de perfiles se presentan en forma de lista. La lista se puede ordenar en base a varios criterios (p. ej. en base a la duración de la ejecución de la instrucción o de la ejecución de la instrucción y sus descendientes).

© 2016-2022 Altova GmbH



- **Vista en forma de árbol:** las estadísticas se presentan en una estructura en forma de árbol en la que por ejemplo puede ver cuánto tardó en ejecutarse una función y expandir el árbol de la función para ver cuánto tardó en ejecutarse cada instrucción de la función y cuántas veces se ejecutó.



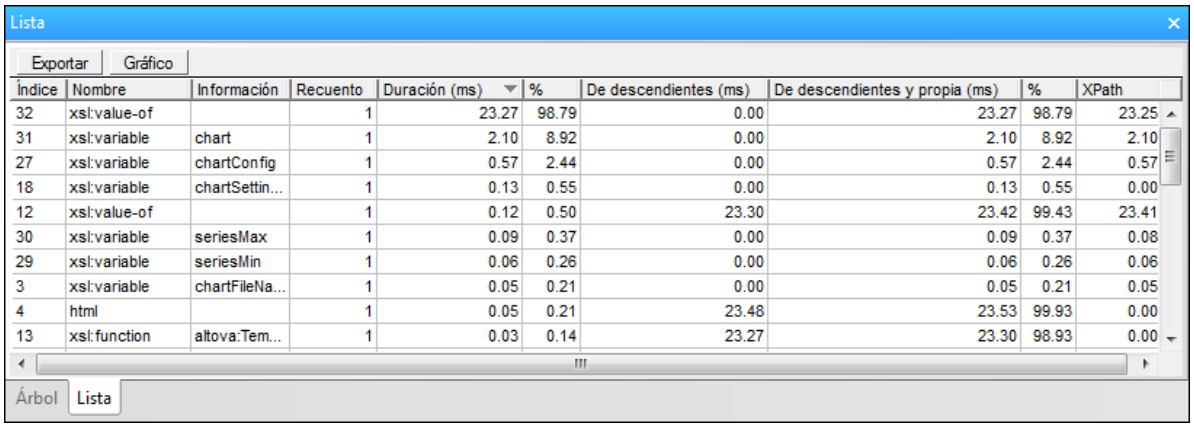
Ordenar los resultados

Después de ejecutar el generador de perfiles puede ordenar los resultados en base a varios criterios, como el tiempo que tardó en ejecutarse una función o cuántas veces se llamó a una instrucción.

Para ordenar la información del generador de perfiles:

1. Haga clic en la pestaña **Lista**.
2. Haga clic en el encabezado de la columna que desea ordenar (p. ej. haga clic en **Recuento** para ordenar los resultados en base al número de veces que se llamó a las instrucciones o en **Duración** para ordenar los resultados en base al tiempo que tarda en ejecutarse cada instrucción).

   En esta captura de pantalla del generador de perfiles los resultados se ordenaron en base a la duración de la instrucción y en orden descendente.



Optimizar sus hojas de estilos y documentos XQuery

Cuando optimice el tiempo de ejecución de las instrucciones de sus documentos XQuery y XSLT tenga en cuenta estos aspectos:

- La inicialización de variables requiere demasiado tiempo. Por tanto, evite usar variables en una instrucción si la variable se utiliza una sola vez.

  Los fragmentos de código XSLT que aparecen a continuación son un ejemplo de cómo puede optimizar el código si quita variables innecesarias. Ambos códigos hacen lo mismo, pero el segundo lo hace sin usar las variables `name` ni `containsResult`:

  Fragmento de código nº 1:

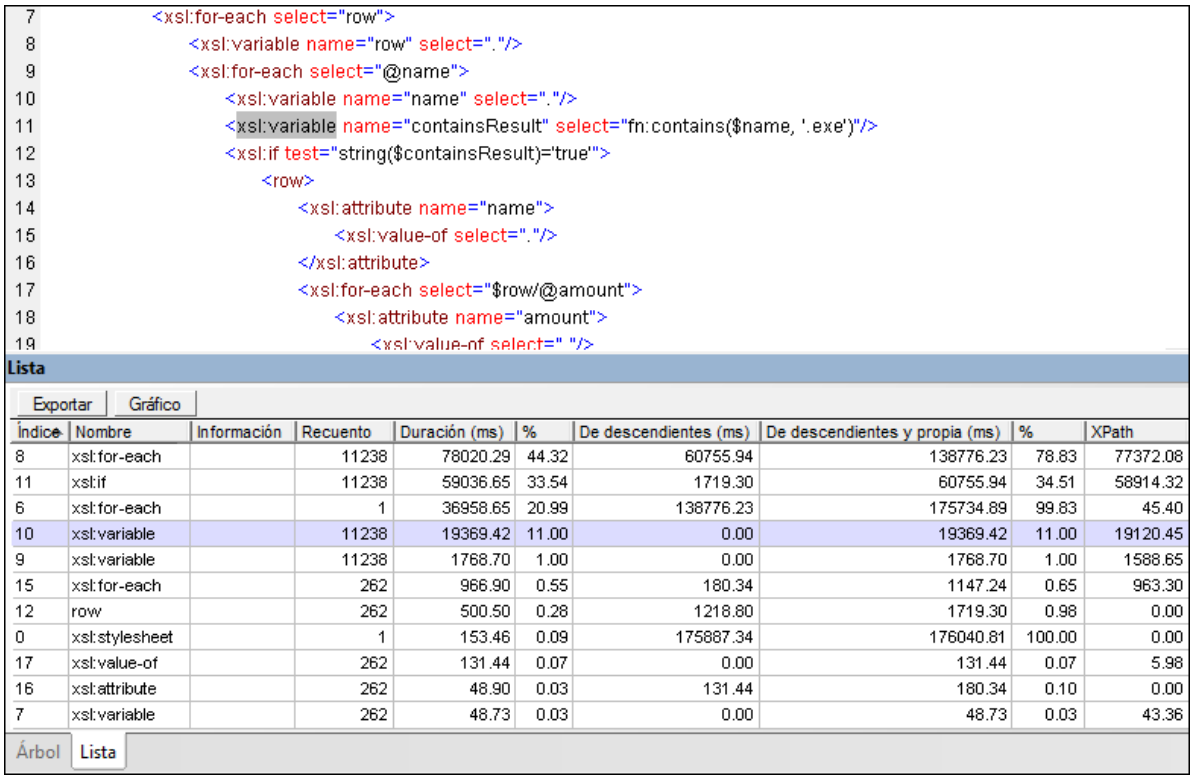
```
<xsl:for-each select="row">
	<xsl:variable name="row" select="."/>
	<xsl:for-each select="@name">
		<xsl:variable name="name" select="."/>
		<xsl:variable name="containsResult" select="fn:contains($name,
'.exe')"/>
		<xsl:if test="string($containsResult)='true'">

		...

		</xsl:if>
	</xsl:for-each>
</xsl:for-each>
```

  La imagen que aparece a continuación muestra los resultados que se obtienen al analizar el archivo que contiene este fragmento de código, ordenados en base a la duración de las instrucciones. La instrucción en la que se inicializa la variable `containsResult` tarda en total 19 segundos en ejecutarse.

 © 2016-2022 Altova GmbH



```
 7            <xsl:for-each select="row">
 8                <xsl:variable name="row" select="."/>
 9                <xsl:for-each select="@name">
10                    <xsl:variable name="name" select="."/>
11                    <xsl:variable name="containsResult" select="fn:contains($name, '.exe')"/>
12                    <xsl:if test="string($containsResult)='true'">
13                        <row>
14                            <xsl:attribute name="name">
15                                <xsl:value-of select="."/>
16                            </xsl:attribute>
17                            <xsl:for-each select="$row/@amount">
18                                <xsl:attribute name="amount">
19                                    <xsl:value-of select="."/>
```

Lista

Exportar | Gráfico

Índice	Nombre	Información	Recuento	Duración (ms)	%	De descendientes (ms)	De descendientes y propia (ms)	%	XPath
8	xsl:for-each		11238	78020.29	44.32	60755.94	138776.23	78.83	77372.08
11	xsl:if		11238	59036.65	33.54	1719.30	60755.94	34.51	58914.32
6	xsl:for-each		1	36958.65	20.99	138776.23	175734.89	99.83	45.40
10	xsl:variable		11238	19369.42	11.00	0.00	19369.42	11.00	19120.45
9	xsl:variable		11238	1768.70	1.00	0.00	1768.70	1.00	1588.65
15	xsl:for-each		262	966.90	0.55	180.34	1147.24	0.65	963.30
12	row		262	500.50	0.28	1218.80	1719.30	0.98	0.00
0	xsl:stylesheet		1	153.46	0.09	175887.34	176040.81	100.00	0.00
17	xsl:value-of		262	131.44	0.07	0.00	131.44	0.07	5.98
16	xsl:attribute		262	48.90	0.03	131.44	180.34	0.10	0.00
7	xsl:variable		262	48.73	0.03	0.00	48.73	0.03	43.36

Árbol | Lista

Esta otra imagen muestra los resultados en la vista **Árbol**. Aquí podemos ver que la instrucción `If` que utiliza la variable `containsResult` tarda en total 50 segundos en ejecutarse:

```
7           <xsl:for-each select="row">
8               <xsl:variable name="row" select="."/>
9               <xsl:for-each select="@name">
10                  <xsl:variable name="name" select="."/>
11                  <xsl:variable name="containsResult" select="fn:contains($name, '.exe')"/>
12                  <xsl:if test="string($containsResult)='true'">
13                      <row>
14                          <xsl:attribute name="name">
15                              <xsl:value-of select="."/>
16                          </xsl:attribute>
17                          <xsl:for-each select="$row/@amount">
18                              <xsl:attribute name="amount">
```

Árbol

Exportar | Gráfico

Nombre	Información	Recuento	Duración (ms)	%	De descendientes y propia (ms)	%
xsl:stylesheet		1	11.00	0.01	76960.70	100.00
xsl:output		0	0.00	0.00	0.00	0.00
xsl:template		1	0.01	0.00	76911.74	99.94
root		1	82.97	0.11	76911.73	99.94
xsl:attribute		1	54.04	0.07	54.06	0.07
xsl:for-each		1	2614.55	3.40	76774.70	99.76
xsl:variable		262	44.36	0.06	44.36	0.06
xsl:for-each		11238	22196.25	28.84	74160.15	96.36
xsl:variable		11238	1619.42	2.10	1619.42	2.10
xsl:variable		11238	13286.12	17.26	13286.12	17.26
xsl:if		11238	50441.78	65.54	51963.90	67.52
row		262	199.53	0.26	1522.12	1.98

Árbol | Lista

En total la transformación XSLT tarde unos 74 segundos:

Árbol

Exportar | Gráfico

Nombre	Información	Recuento	Duración (ms)	%	De descendientes (ms)	De descendientes y propia (ms)	%	XPath
xsl:stylesheet		1	20.33	0.03	73882.64	73902.97	100.00	0.00
xsl:output		0	0.00	0.00	0.00	0.00	0.00	0.00
xsl:template		1	0.01	0.00	73856.08	73856.09	99.94	0.00

Árbol | Lista

Fragmento de código nº 2:

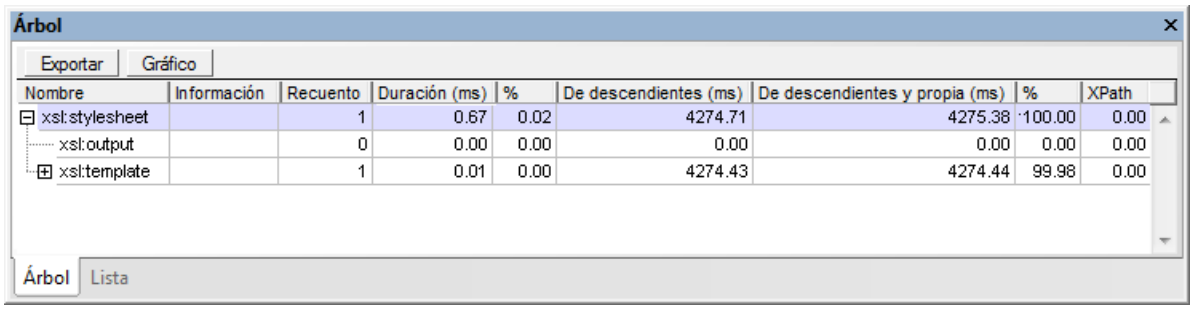
```
<xsl:for-each select="row">
       <xsl:variable name="row" select="."/>
       <xsl:for-each select="@name">
              <xsl:if test="fn:contains(., '.exe')">

              ...
```


© 2016-2022 Altova GmbH

```
            </xsl:if>
        </xsl:for-each>
</xsl:for-each>
```

Después de volver a escribir la hoja de estilos y eliminar las variables, el tiempo total de ejecución es unos 4,3 segundos:



- Es recomendable usar variables si un valor o expresión se usa repetidas veces.
- Intente no crear variables constantes locales dentro de una función. Es preferible crear variables globales.
- Intente no crear fragmentos de árbol de constantes dentro de una función. Es preferible crearlos globalmente.
- El filtrado con predicados se evalúa por separado por cada nodo. Por tanto, es recomendable reducir el uso de predicados. Puede reducir el número de llamadas a predicados si realiza filtros previos con nombres. En este ejemplo, `*` se usa con dos predicados:

  ```
  //*[node-name()=Book][author="Steve"]
  ```

  En esta otra instrucción equivalente a la anterior, se usan el nombre `Book` y un solo predicado:

  ```
  //Book[@Author="Steve"]
  ```

- Es recomendable dividir las instrucciones de modo que la parte de la instrucción que se debe ejecutar una sola vez se ejecute una sola vez. También es preferible crear variables globales a partir de las partes que dependen sólo del contexto global.

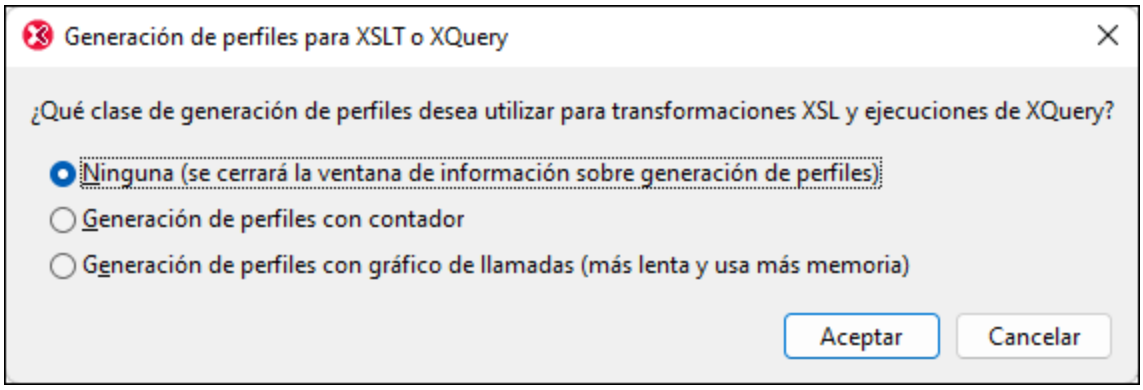
9.2.1 Generación de perfiles XSLT

Iniciar el generador de perfiles

Los resultados del tiempo de ejecución que aparecen en el generador de perfiles puede verse afectado por otras aplicaciones que se estén ejecutando en el equipo. Cuando analice los archivos con el generador de perfiles, es recomendable cerrar todas las aplicaciones excepto XMLSpy.

Para analizar una hoja de estilos XSLT:

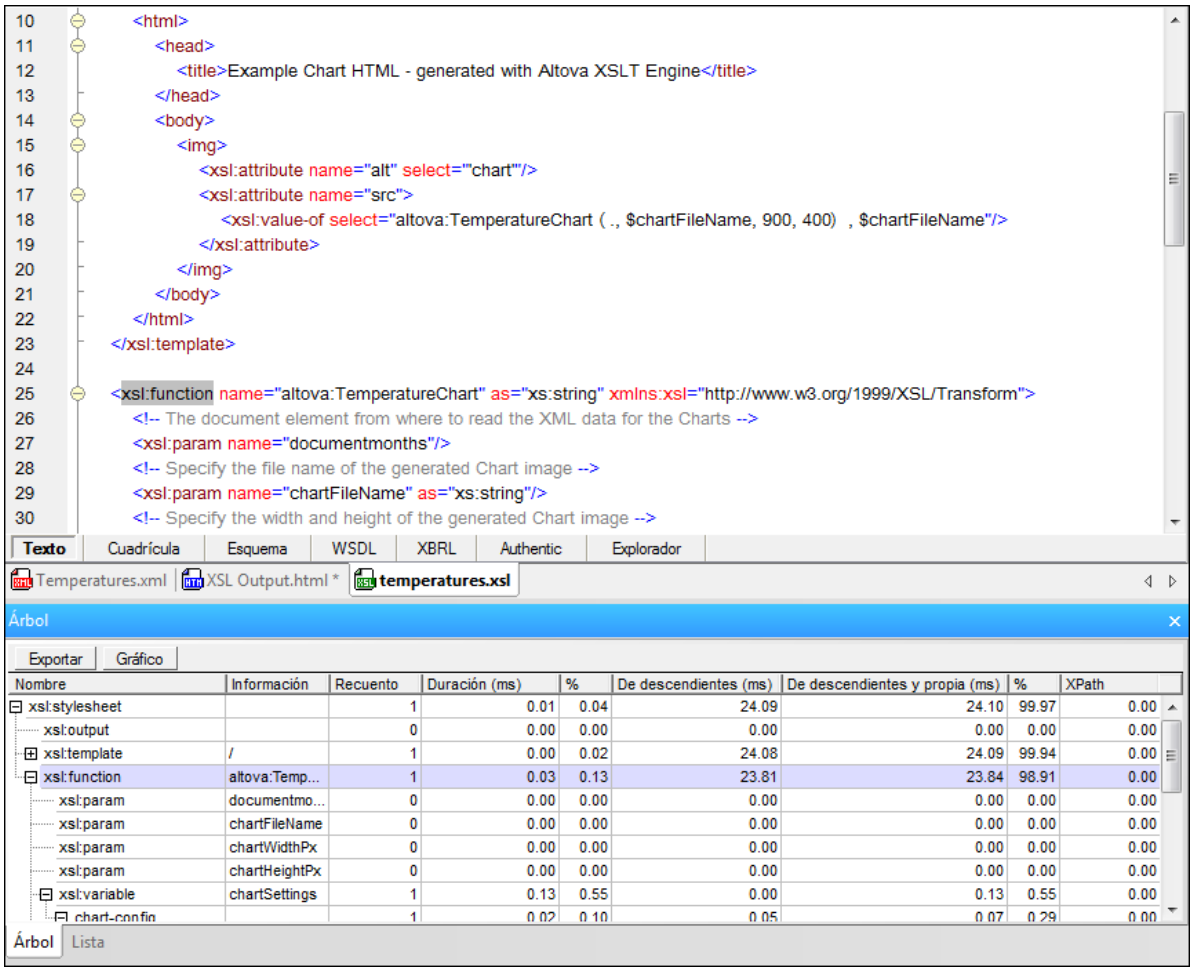
1. En XMLSpy abra el archivo XML que se usará como datos de entrada para la transformación XSLT.
2. Seleccione el comando **XSL/XQuery | Generación de perfiles para XSLT o XQuery**. Aparece este cuadro de diálogo:



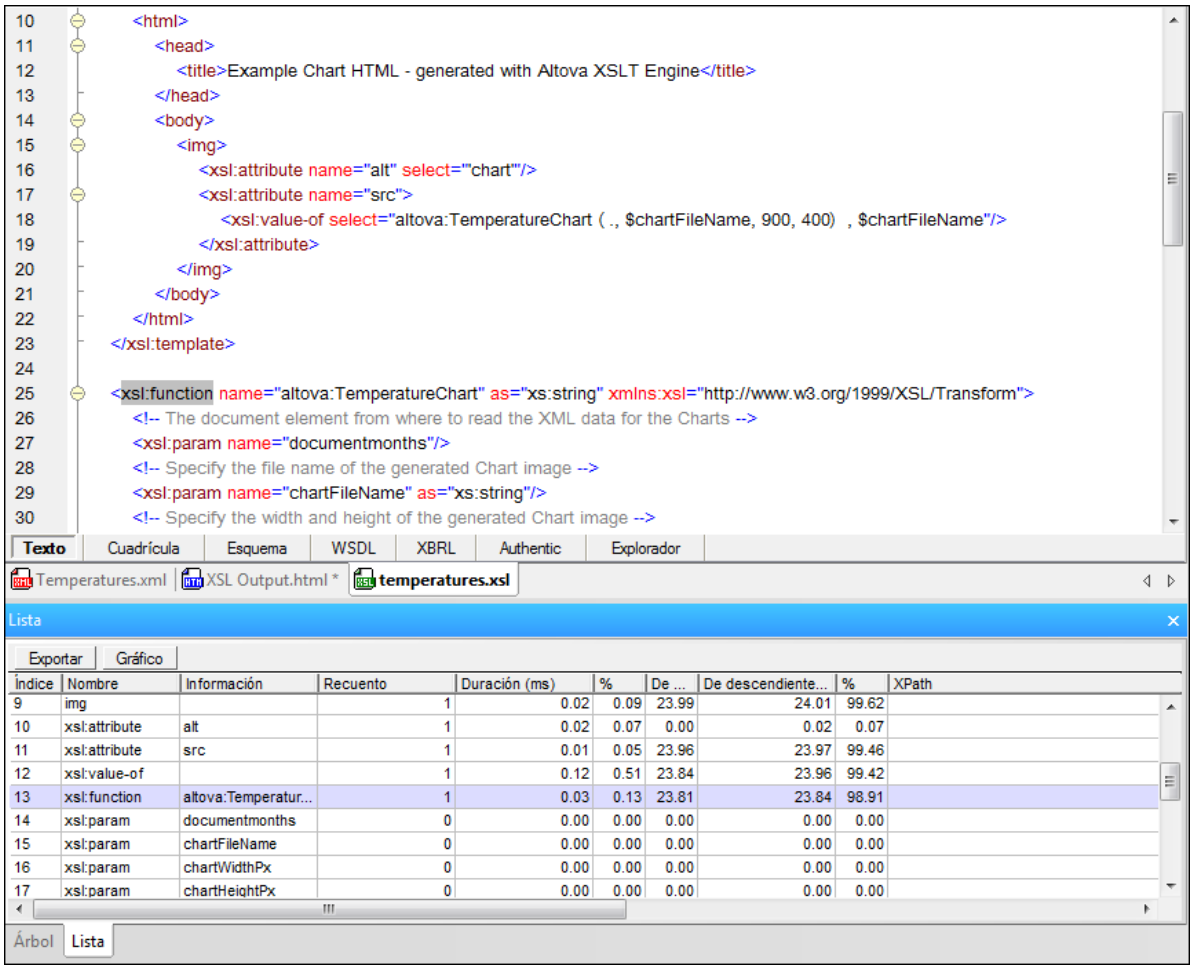
3. Seleccione la opción **Generación de perfiles con contador** o **Generación de perfiles con gráfico de llamadas** y haga clic en **Aceptar**. Aparece una ventana vacía del generador de perfiles.
4. Ejecute la transformación XSL (**XSL/XQuery | Transformación XSL**). Aparece un cuadro de diálogo donde debe seleccionar la ruta de acceso de la hoja de estilos que desea analizar. Una vez terminada la transformación, el generador de perfiles muestra las estadísticas del tiempo de ejecución.
5. Haga clic en los iconos + para expandir las filas de la vista **Árbol** del generador de perfiles para ver las estadísticas del tiempo de ejecución de la hoja de estilos XSLT (*ver imagen siguiente*). Tenga en cuenta que en las imágenes que aparecen en este apartado se seleccionó la **generación de perfiles con contador**.Haga clic en una fila del generador de perfiles para ver la instrucción correspondiente en el archivo que se analizó.

Esta captura de pantalla muestra la vista **Árbol** del generador de perfiles:

© 2016-2022 Altova GmbH



Esta captura de pantalla muestra la misma hoja de estilos XSLT en la vista **Lista** del generador de perfiles:



Cómo usar la información del generador de perfiles

Por cada instrucción de la hoja de estilos XSLT, el generador de perfiles ofrece esta información:

- **Índice:** el número que se asigna a cada instrucción dependiendo de cuándo se llamó a la instrucción (es decir, si es la primera instrucción a la que se llamó, la segunda, etc.)
- **Nombre:** el nombre de la instrucción XSLT.
- **Recuento:** el número total de veces que se llamó a la instrucción durante la transformación.
- **Duración (ms)** y **%:** el número de milisegundos que tardó en ejecutarse la instrucción, sin tener en cuenta el tiempo de ejecución de sus descendientes, y el porcentaje con respecto al tiempo total de ejecución.
- **De descendientes (ms):** la duración en milisegundos de la ejecución de los descendientes de la instrucción.
- **De descendientes y propia (ms)** y **%:** la duración en milisegundos de la ejecución de la instrucción y de sus descendientes, así como el porcentaje con respecto al tiempo total de ejecución.
- **XPath:** si la instrucción contiene una instrucción XPath, esta columna indica cuánto tardó en ejecutarse esa instrucción.

Nota: en la generación de perfiles con recuento la duración indicada en la ventana del generador es la suma total del tiempo de ejecución de todos las llamadas a esa instrucción. La generación de perfiles con

 © 2016-2022 Altova GmbH

gráfico de llamadas, por su parte, muestra por separado cada llamada a la instrucción y por tanto la duración indicada en la ventana del generador en ese caso es la duración de una sola ejecución de la instrucción.

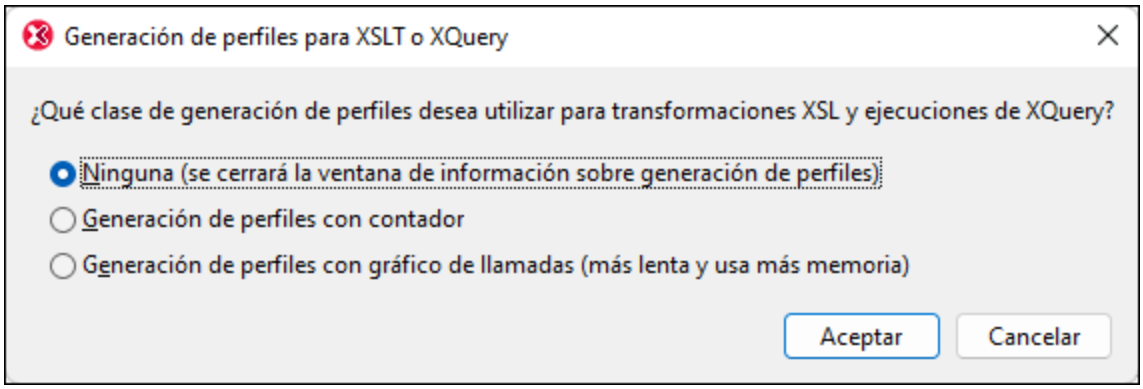
9.2.2 Generación de perfiles XQuery

Iniciar el generador de perfiles

Los resultados del tiempo de ejecución que aparecen en el generador de perfiles puede verse afectado por otras aplicaciones que se estén ejecutando en el equipo. Cuando analice los archivos con el generador de perfiles, es recomendable cerrar todas las aplicaciones excepto XMLSpy.

Para analizar un documento XQuery:

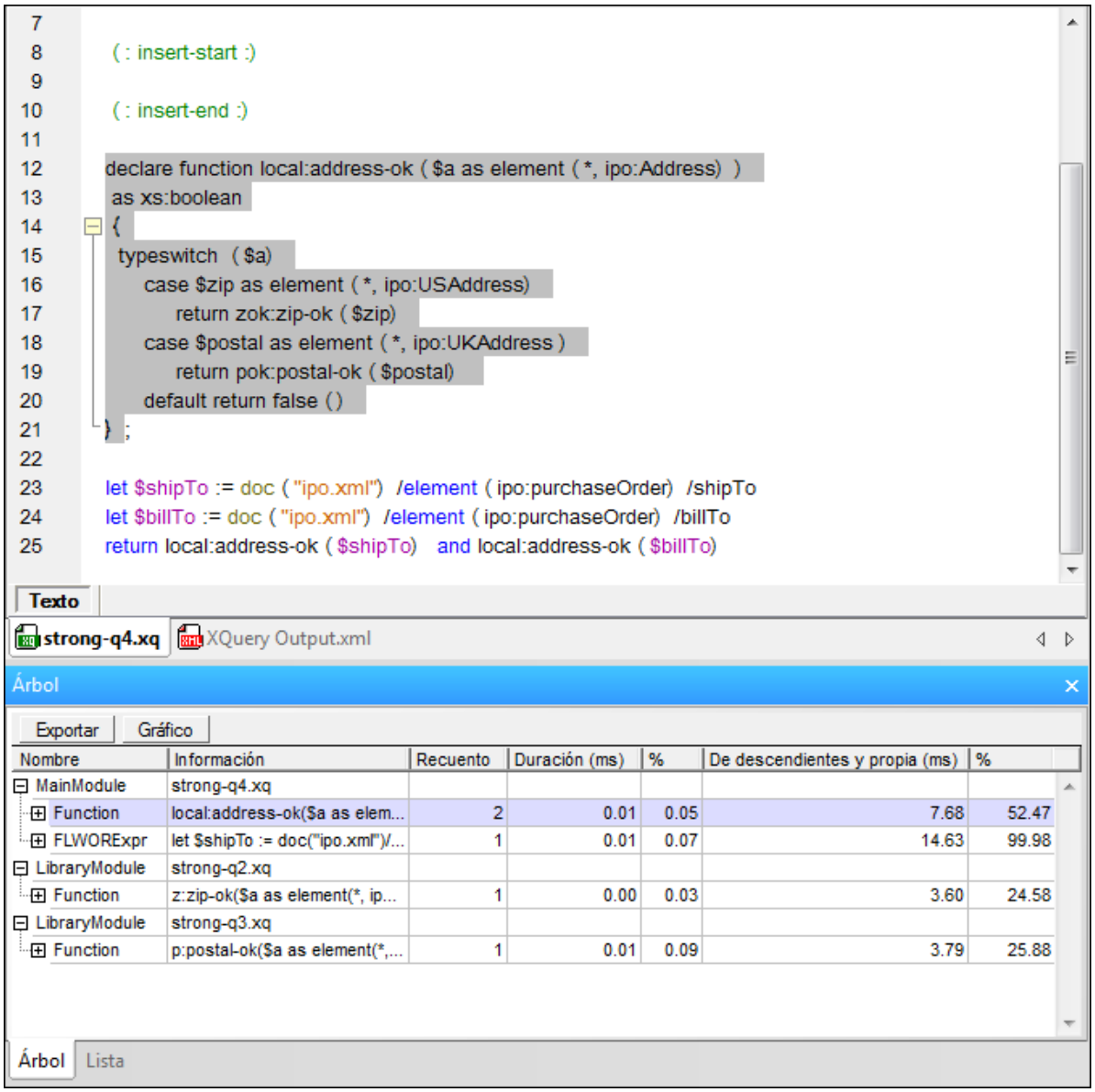
1. En XMLSpy abra el documento XQuery que desea analizar.
2. Seleccione el comando **XSL/XQuery | Generación de perfiles para XSLT o XQuery**. Aparece este cuadro de diálogo.



3. Seleccione la opción **Generación de perfiles con contador** o **Generación de perfiles con gráfico de llamadas** y haga clic en **Aceptar**. Aparece una ventana vacía del generador de perfiles.
4. Ejecute el documento XQuery (**XSL/XQuery | Ejecutar XQuery**). Una vez finalizada la ejecución aparecen las estadísticas del tiempo de ejecución en el generador de perfiles.
5. Haga clic en los iconos + para expandir las filas de la vista **Árbol** del generador de perfiles para ver las estadísticas del tiempo de ejecución de las instrucciones del documento XQuery (*ver imagen siguiente*). Tenga en cuenta que en las imágenes que aparecen en este apartado se seleccionó la **generación de perfiles con contador**.

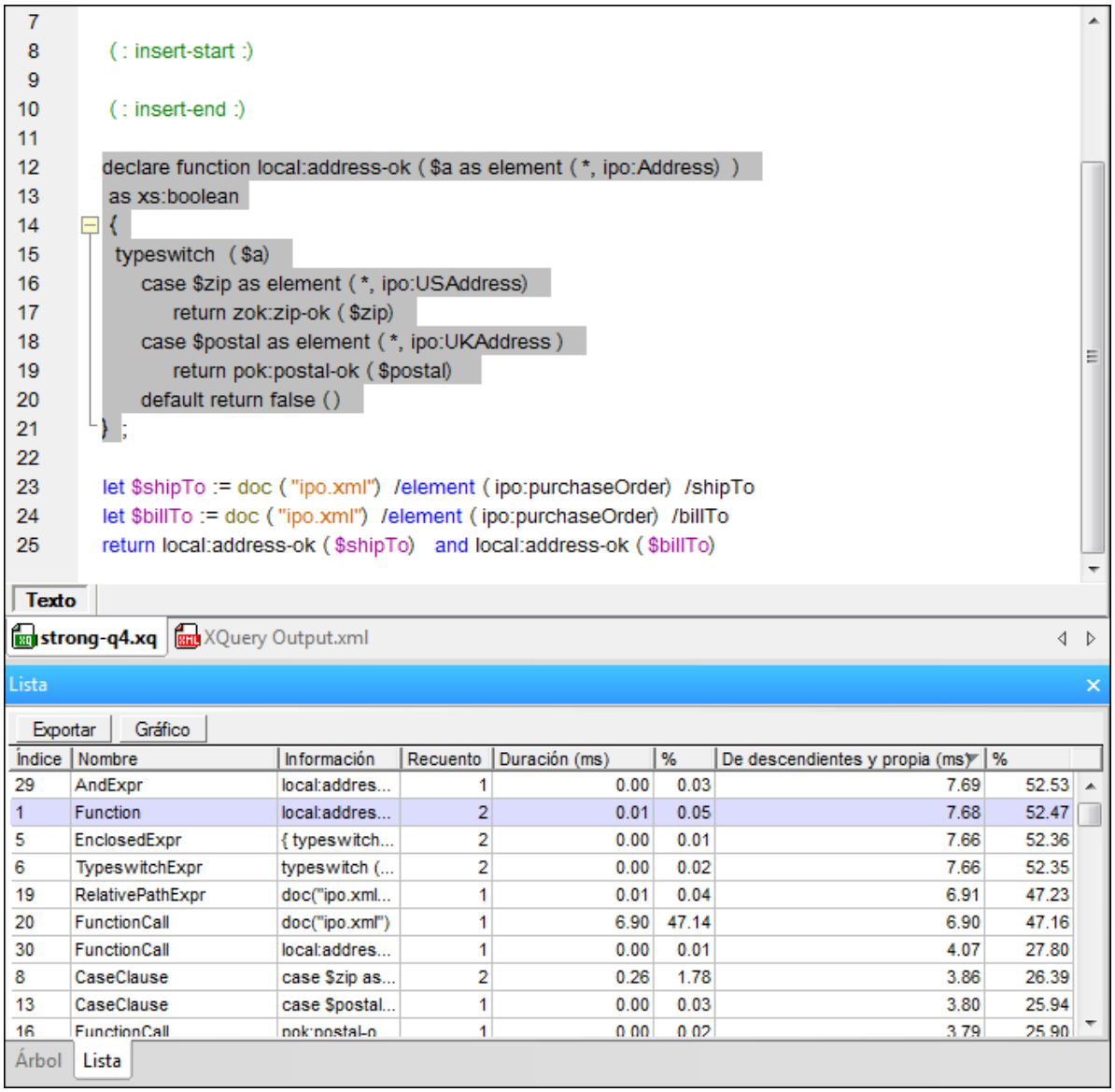
Haga clic en una fila del generador de perfiles para ver la instrucción correspondiente en el archivo que se analizó.

Esta captura de pantalla muestra la vista **Árbol** del generador de perfiles:



Esta captura de pantalla muestra el mismo documento XQuery en la vista **Lista** del generador de perfiles:

© 2016-2022 Altova GmbH



Cómo usar la información del generador de perfiles

Por cada instrucción del documento XQuery, el generador de perfiles ofrece esta información:

- **Índice:** el número que se asigna a cada instrucción dependiendo de cuándo se llamó a la instrucción (es decir, si es la primera instrucción a la que se llamó, la segunda, etc.)
- **Nombre:** el nombre de la instrucción XQuery.
- **Información:** información sobre la instrucción. Por ejemplo, si la instrucción es una declaración de variable, esta columna contiene el nombre de la variable y su valor. Si es una función, la columna contiene el nombre y los parámetros de la función.
- **Recuento:** el número total de veces que se llamó a la instrucción durante la ejecución.
- **Duración (ms)** y **%:** el número de milisegundos que tardó en ejecutarse la instrucción, sin tener en cuenta el tiempo de ejecución de sus descendientes, y el porcentaje con respecto al tiempo total de ejecución.

- **De descendientes (ms):** la duración en milisegundos de la ejecución de los descendientes de la instrucción.
- **De descendientes y propia (ms)** y **%:** la duración en milisegundos de la ejecución de la instrucción y de sus descendientes, así como el porcentaje con respecto al tiempo total de ejecución.
- **XPath:** si la instrucción contiene una instrucción XPath, esta columna indica cuánto tardó en ejecutarse esa instrucción.

Nota: en la generación de perfiles con recuento la duración indicada en la ventana del generador es la suma total del tiempo de ejecución de todos las llamadas a esa instrucción. La generación de perfiles con gráfico de llamadas, por su parte, muestra por separado cada llamada a la instrucción y por tanto la duración indicada en la ventana del generador en ese caso es la duración de una sola ejecución de la instrucción.

9.2.3 Resultados del generador de perfiles

Después de ejecutar el generador de perfiles XSLT o XQuery, puede exportar los resultados a un archivo XML o a un gráfico que se puede guardar a su vez como archivo de imagen.

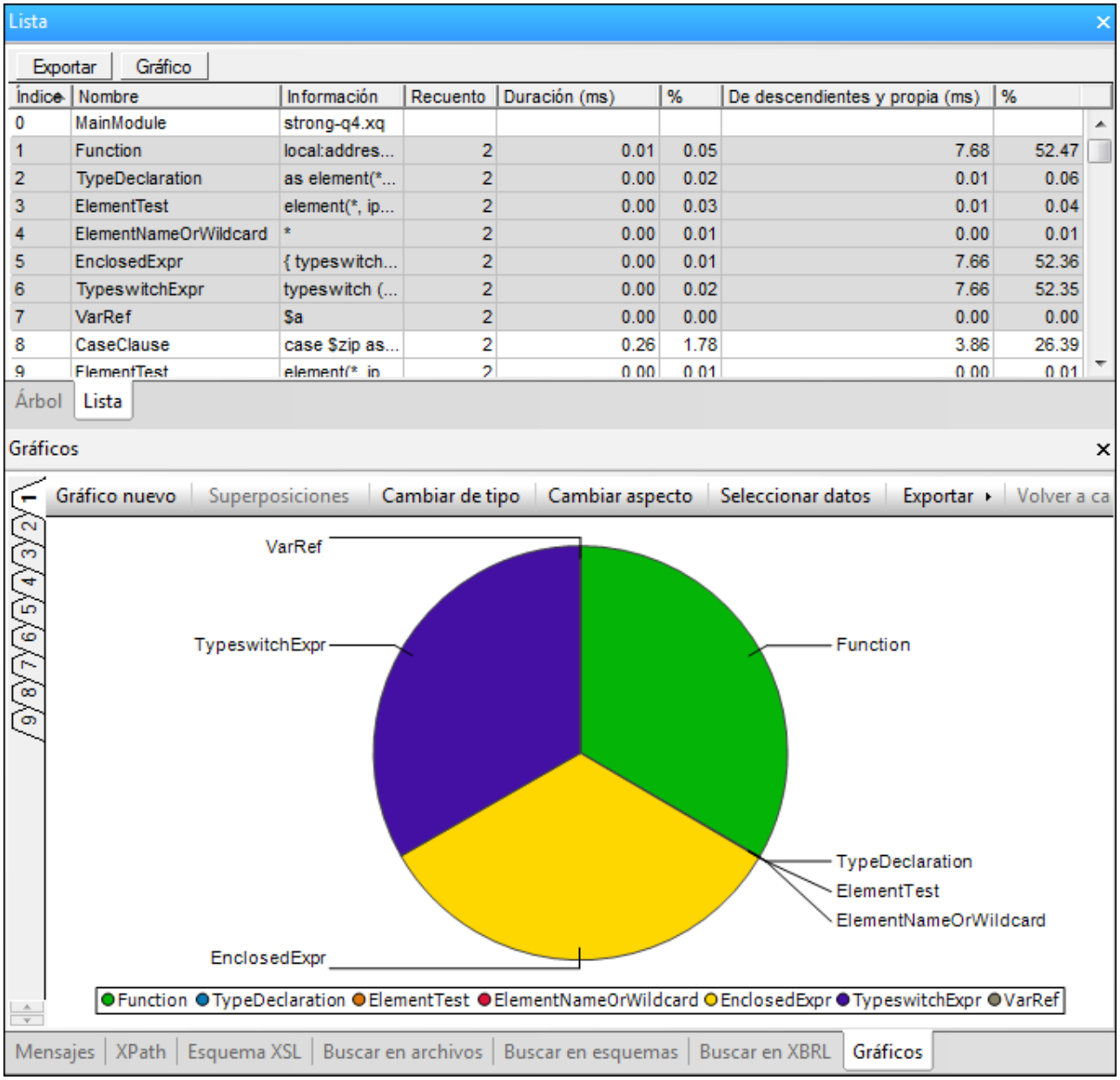
Exportación

Al hacer clic en el botón **Exportar** aparece un mensaje solicitando que seleccione una ubicación y un nombre de archivo para el archivo XML en el que se deben guardar los resultados del generador de perfiles. Para apreciar mejor la estructura de los datos, recomendamos que visualice el archivo XML en la vista Cuadrícula. Por ejemplo, al visualizar los resultados del generador de perfiles XSLT en la vista Cuadrícula, observará que la estructura del documento XML está formada por tres niveles jerárquicos, cada uno de los cuales se identifica con un elemento `node`. El primer elemento `node` representa la raíz del documento, el segundo elemento `node` representa el elemento `xsl:stylesheet` y el tercero representa los elementos globales (como `xsl:output` y `xsl:template`). Los datos del generador de perfiles se almacenan en los atributos de los elementos `node`.

Gráfico

Tras ejecutar el generador de perfiles XSLT/XQuery puede generar un gráfico a partir de los resultados del generador o de un subconjunto de resultados. En la ventana del generador (*ver imagen siguiente*), haga clic en el botón **Gráfico** para generar el gráfico en la ventana Gráficos (*ver imagen siguiente*).

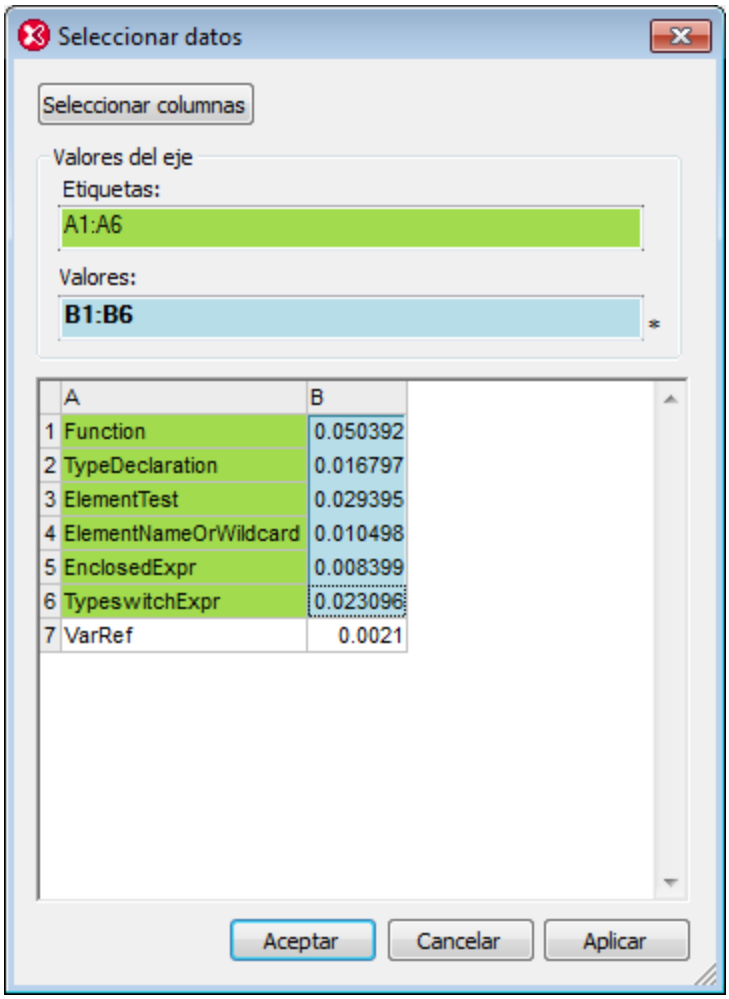
© 2016-2022 Altova GmbH



Es necesario tener en cuenta estos aspectos:

- En la ventana del generador (en la vista Árbol o Lista), puede marcar y seleccionar un subconjunto de resultados. Para marcas varios elementos, mantenga pulsada las teclas **Ctrl** o **Mayús** mientras selecciona los elementos. En la vista Lista los resultados se pueden ordenar según los valores de una columna. Para ello haga clic en el encabezado de la columna pertinente. Esta característica es muy práctica si desea, por ejemplo, ordenar los resultados en base a su duración y después selecciona los elementos que tarden más tiempo en ejecutarse. Para el gráfico de la imagen anterior se seleccionaron los elementos resaltados en la lista del generador de perfiles.
- Después de crear el gráfico, puede cambiar de tipo de gráfico (circular, de barras, de líneas, etc.) con el botón **Cambiar de tipo** 369 de la ventana Gráficos. Puede consultar una descripción de los diferentes tipos de gráficos en la sección Gráficos 346.

- Al hacer clic en el botón **Seleccionar datos** de la ventana Gráficos se abre el cuadro de diálogo "Seleccionar datos" (*imagen siguiente*). En este diálogo puede seleccionar los datos para el eje X e Y desde la tabla de datos que se genera como resultado de la selección de columnas. Para seleccionar los datos del eje X, haga clic en el cuadro de texto "Valores del eje" e introduzca el intervalo de valores de la tabla (p. ej. `A1:A6`) o arrastre el cursor desde el principio hasta el final del intervalo. Haga lo mismo para seleccionar los datos del eje Y.



  Para cambiar la selección de datos para la tabla de datos, haga clic en el botón **Seleccionar columnas**. Para más información sobre cómo funciona la selección de columnas, consulte los apartados XPath de origen[353], Selección del eje X[357] y Selección del eje Y[362].

Para más información sobre la función de creación de gráficos, consulte la sección Gráficos[346].

© 2016-2022 Altova GmbH

10 Expresiones XPath/XQuery

Las expresiones XPath y XQuery se usan para navegar estructuras XML. La versión 3.1 también es compatible con mapas y matrices, por lo que las expresiones XPath y XQuery 3.1 también se pueden usar para navegar estructuras JSON. XPath es un subconjunto de XQuery y cualquier expresión que sea válida en ambos lenguajes devolverá el mismo resultado en los dos. Para más información sobre estos dos lenguajes consulte las recomendaciones XPath 3.1 y XQuery 3.1.

Evaluar expresiones XPath/XQuery con XMLSpy

XMLSpy permite crear expresiones XPath/XQuery y evaluar y depurar expresiones con documentos XML y JSON. Esto permite generar y probar expresiones rápidamente con los documentos XML y JSON con los que pretende usarlas.

Estas características de análisis se encuentran en la ventana *XPath/XQuery*, que es una ventana de resultados situada por defecto entre el resto de ventanas de resultado 119, en la parte inferior de la interfaz de la aplicación. Las características de la ventana *XPath/XQuery* se describen en los apartados de esta sección.

Por lo general se siguen estos pasos:

1. Abra el documento XML o JSON para el que quiere generar o evaluar una expresión.
2. Introduzca la expresión XPath/XQuery en la ventana XPath/XQuery.
3. Ejecute el Evaluador o el Depurador para ver los resultados. El Evaluador muestra el resultado final, mientras que el Depurador permite ir recorriendo el proceso de evaluación y va mostrando partes del resultado en cada paso.

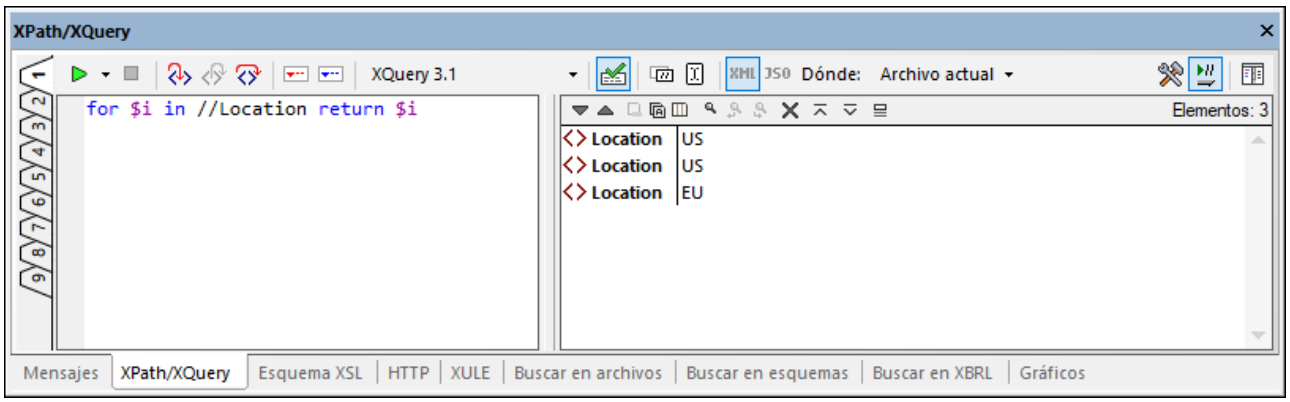
En esta sección

Esta sección describe las características de la ventana XPath/XQuery y cuenta con los siguientes apartados:

- La ventana XPath/XQuery 574
- Modo Evaluador 576: para evaular expresiones XPath/XQuery
- Modo Depurador 582: para depurar expresiones XPath/XQuery
- Generador de expresiones 590: para generar XPath/XQuery
- Expresiones XQuery para JSON 593
- Aspectos importantes 595

10.1 La ventana XPath/XQuery

La ventana XPath/XQuery[127] (*imagen siguiente*) permite generar, evaluar y depurar expresiones XPath y XQuery con respecto a documentos XML o JSON*. (Las consultas JSON no se introdujeron hasta la versión XPath/XQuery 3.1. Consulte también *Transformaciones JSON con XSLT/XQuery*[758].)

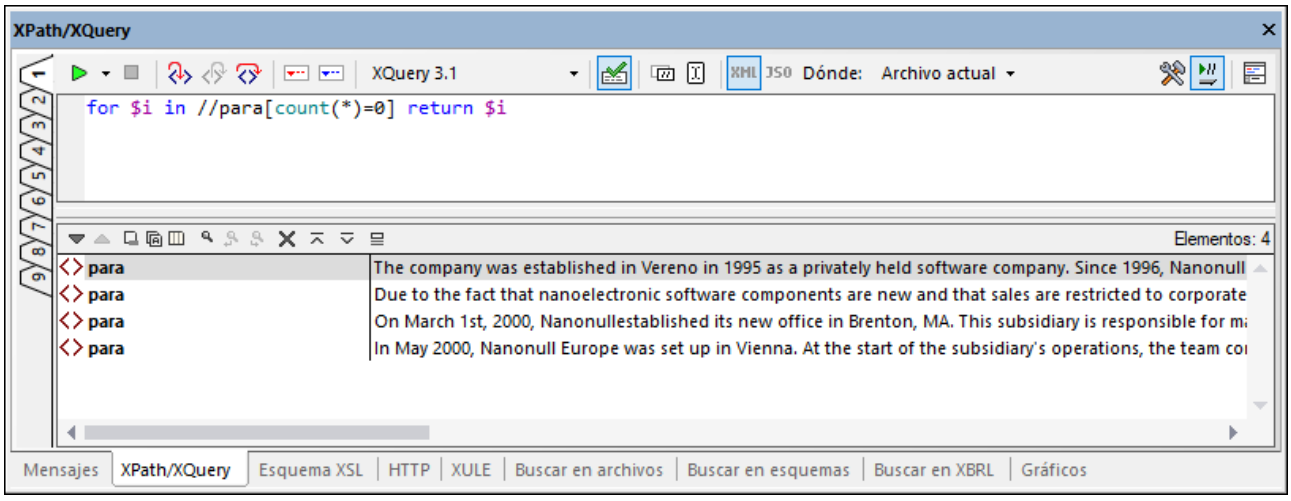


Para ver una descripción más amplia de esta ventana y su barra de herramientas consulte Ventana de resultados: XPath/XQuery[127].

Diseño vertical y horizontal

En la esquina superior derecha de la ventana (*imágenes anterior y siguiente*) encontrará un botón que sirve para cambiar del diseño horizontal y al diseño vertical y viceversa. Puede alternar entre los dos diseños y los dos modos en cualquier momento (*véanse más abajo el modo Evaluador*[575] *y el modo Depurador*[575]). La imagen anterior muestra el diseño vertical, que es útil si una expresión XPath/XQuery (*en el panel izquierdo de la imagen anterior*) tiene varias líneas.

El diseño horizontal (*imagen siguiente*) es útil en los casos en que el resultado contiene líneas continuadas muy largas.



 © 2016-2022 Altova GmbH

Nueve pestañas

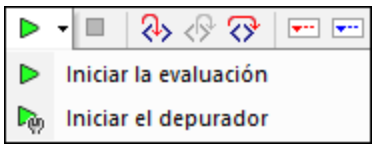
La ventana XPath/XQuery cuenta con nueve pestañas en su parte izquierda (*imagen anterior*). Tener varias pestañas permite trabajar con distintas expresiones en diferentes pestañas y comparar los resultados. Para ir a una pestaña basta con hacer clic en su tirador.

Modo Evaluador y modo Depurador

La ventana XPath/XQuery 127 se puede usar en dos modos:

- Modo Evaluador 576, en el que la expresión XPath o XQuery se evalúa con respecto a uno o más documentos XML/JSON. La expresión se introduce en el panel *Expresión* y el resultado se muestra en el panel adyacente *Resultados*. Puede hacer clic en nodos del resultado para ir a ese nodo en el documento XML o JSON.
- Modo Depurador 582, en el que puede depurar una expresión XPath/XQuery mientras esta se aplica al documento XML activo. Puede definir puntos de interrupción y puntos de seguimiento y seguir el proceso de evaluación paso a paso. En cada paso puede ver el contenido de las variables, así como definir expresiones de inspección para comprobar otros aspectos de la evaluación.

Para cambiar de un modo a otro, seleccione el comando correspondiente en el menú desplegable **Empezar evaluación/depuración**, ubicado en la parte izquierda de la barra de herramientas de la ventana (*imagen siguiente*).



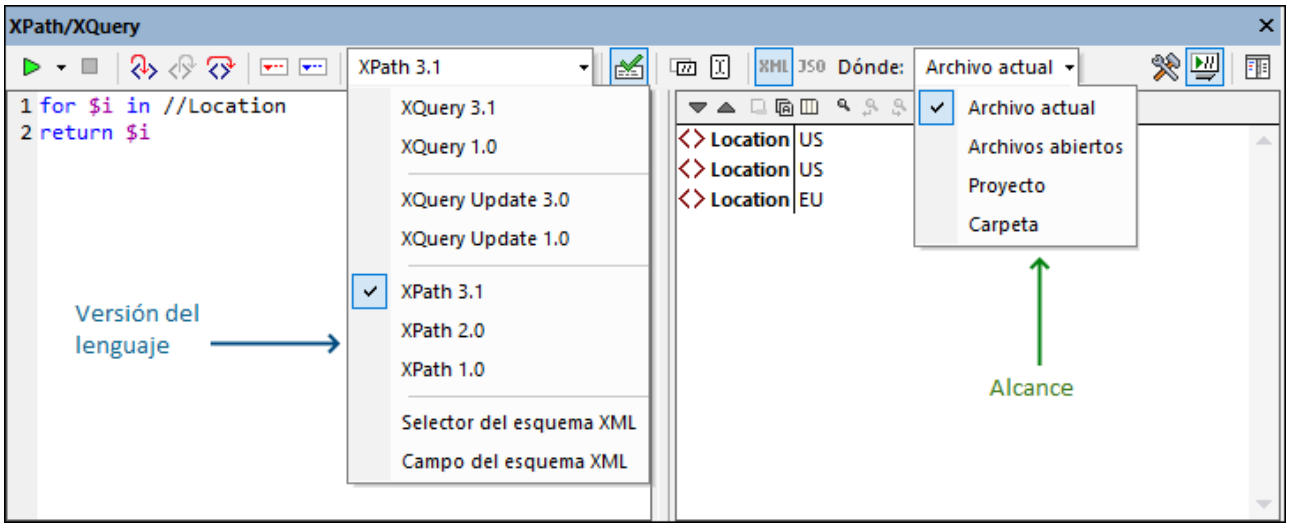
En los apartados de esta sección se explica cómo usar cada uno de estos dos modos.

Generador de expresiones XPath/XQuery

En los dos modos 575 se puede usar el generador de expresiones 590 como ayuda para construir expresiones con una sintaxis correcta. Active o desactive el generador de expresiones 590 con el botón **Cambiar al modo generador** de la barra de herramientas principal .

10.2 Modo Evaluador

La ventana XPath/XQuery permite generar expresiones XPath o XQuery (puede elegir la versión del lenguaje que quiere usar; *imagen siguiente*) y después evaluarla con un alcance que puede indicar en la opción *Dónde* (*imagen siguiente*). La expresión se puede evaluar en el archivo actual, además de en varios conjuntos de archivos: (i) todos los documentos abiertos, (ii) archivos del proyecto de XMLSpy[1060] activo o (iii) archivos de una carpeta en concreto. Si solamente quiere probar la expresión basta con que use un archivo. Sin embargo, la ventana XPath/XQuery también se puede usar para encontrar datos concretos en uno o más archivos, que aparecerán en el panel *Resultados*; en este caso el alcance debe comprender un conjunto de archivos.

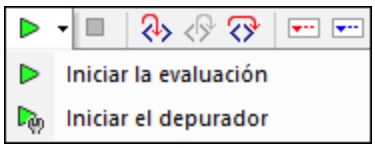


La ventana XPath/XQuery comprende una barra de herramientas y dos paneles: el panel *Expresión* (*a la izquierda de la imagen anterior*) y el panel *Resultados* (*a la derecha de la imagen anterior*).

Proceso de evaluación

Para evaluar una expresión XPath/XQuery siga estos pasos:

1. *Seleccione el modo Evaluador:* seleccione **Iniciar la evaluación** en el menú desplegable del comando **Iniciar evaluación/depuración (F5)**, que está en la parte izquierda de la barra de herramientas (*imagen anterior*).

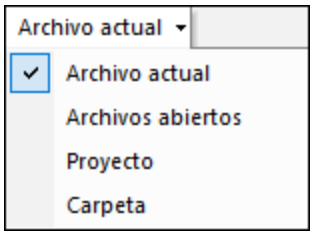


2. *Diseño horizontal/vertical:* para cambiar el diseño haga clic en el botón **Cambiar al diseño horizontal/vertical**, que está en la parte superior derecha de la barra de herramientas. El diseño predeterminado es el vertical.
3. *Seleccione la versión del lenguaje:* en la barra de herramientas seleccione la versión del lenguaje para la expresión que quiere evaluar; por ejemplo XPath 3.1. La versión predeterminada es XPath 3.1. Véase también más abajo la nota *Selección de esquema XML y el campo Esquema XML*[578].
4. *Introducir la expresión:* en el panel Expresión introduzca la expresión que se va a evaluar. Si necesita ayuda para generar la expresión puede usar los ayudantes de entrada del Generador de

© 2016-2022 Altova GmbH

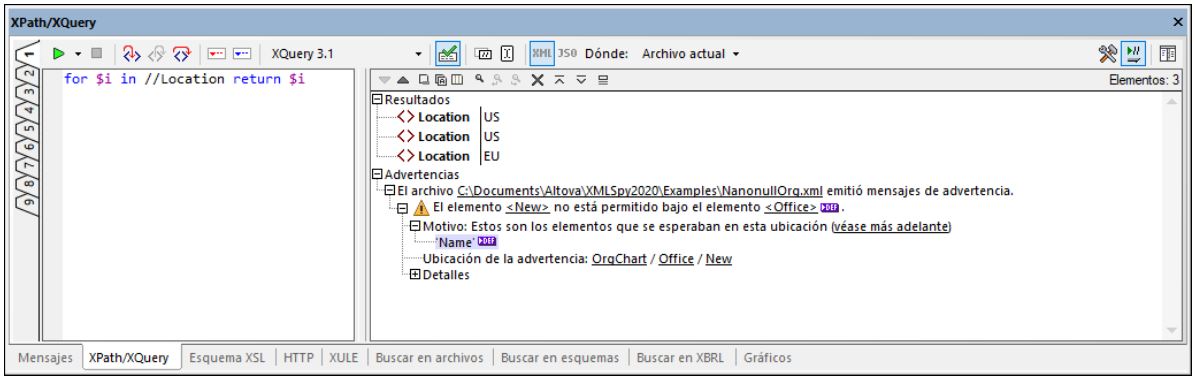
expresiones[590]. Para más información sobre la edición de las características del panel Expresión consulte también, más abajo, la nota *Editar en el panel Expresión*[578].

5. *Seleccionar el alcance de la evaluación:* en la opción *Dónde* de la barra de herramientas seleccione los archivos con los que quiere evaluar la expresión. Las opciones son: Archivos actuales; Archivos abiertos, Proyecto (de *XMLSpy*) o Carpeta. El valor predeterminado es *Archivo actual*.



Si selecciona *Archivo actual* se consultará el archivo que esté activo en ese momento. Si selecciona Archivos abiertos, la expresión se evaluará con respecto a todos los archivos que tenga abiertos en ese momento en XMLSpy. La opción *Proyecto* hace referencia al proyecto de XMLSpy[1060] que esté activo. (Puede omitir las carpetas externas de un proyecto de XMLSpy haciendo clic en el icono *Omitir carpetas externas*) La opción *Carpeta* permite navegar hasta una carpeta en concreto; la expresión XPath se evalúa con respecto a los archivos XML o JSON que contenga esa carpeta.

6. *Seleccione la evaluación XML o JSON:* si el alcance de la evaluación es el archivo actual, el modo de evaluación lo determina el tipo de conformidad del tipo de archivo del documento[1587] (modo JSON para documentos conforme a JSON y XML para el resto). El modo de detección automática no se puede cambiar, por lo que los botones correspondientes están deshabilitados. Si el alcance de la evaluación es una opción que comprende distintos tipos de archivos, entonces se habilitan esos botones y puede escoger el modo que prefiera. El modo predeterminado es el que se haya seleccionado la última vez.
7. *Definir el nodo de contexto:* el nodo de contexto puede ser: (i) el nodo raíz o (ii) la selección actual en el documento activo. Puede alternar entre las dos opciones con el botón de la barra de herramientas **Definir como origen para XPath/XQuery la selección actual en el documento**. El valor predeterminado es el nodo raíz.
8. *Validación XML:* si el botón de la barra de herramientas **Validar el documento XML** está activado (es la opción predeterminada), entonces los archivos XML que se evalúen también se validan. Si hay errores, estos se tratan como advertencias y aparecen en el panel *Resultados* (*imagen siguiente*), pero la evaluación no se detiene.



9. *Evaluar la expresión:* si se activa la opción de la barra de herramientas *Evaluar mientras se teclea*, entonces el resultado de la evaluación aparece en el panel *Resultados* conforme va escribiendo la expresión. Si esta opción no está seleccionada, entonces es necesario iniciar la evaluación de forma explícita haciendo clic en el comando **Iniciar evaluación/depuración (F5)**, en la parte izquierda superior de la barra de herramientas.

Botones de la barra de herramienta que se usan en el proceso de evaluación

	**Iniciar evaluación/el depurador (F5)**	Permite seleccionar el modo Evaluador e inicia la evaluación
	**Detener evaluación/Evaluador (Mayús+F5)**	Se habilita durante la evaluación. Se usa si la evaluación dura mucho o entra en bucle y necesita pararla.
	**Cambiar al diseño vertical/horizontal**	Alterna entre los diseños vertical y horizontal
	**Cambiar al modo generador**	Cambia al modo generador de expresiones, cuyos ayudantes de entrada contextuales ayudan a construir expresiones
	**Copiar la XPath de la selección actual al campo de edición**	Copia la ruta de acceso del nodo del documento XML en la última posición del cursor en el panel Expresión
	**Modo de evaluación XML/JSON** (alterna entre los modos de evaluación XML y JSON)	El icono que esté resaltad de los dos es la opción activa. Si el alcance de la evaluación son varios archivos, se habilitan los dos iconos para que seleccione uno de ellos. De lo contrario el modo de evaluación se selecciona automáticamente en función del tipo de archivo; en ese caso, el otro icono está deshabilitado.
	**Establecer la selección actual como contexto**	Alterna el contexto de la expresión entre el nodo raíz y la selección actual
	**Evaluar al teclear**	Este modo va evaluando las expresiones según las va tecleando
	**Validar XML**	Al activar esta opción se validan los documentos XML de destino

Selección de esquema XML y el campo Esquema XML

Las opciones *Selector del esquema XML* y *Campo del esquema XML* están destinadas a un subconjunto limitado de casos XPath 1.0 y son útiles si se han definido restricciones de identidad única en el esquema XML. Cuando se selecciona cualquiera de estas opciones, en la expresión XPath solo se permiten pruebas de nombres (y el comodín *) y no se permiten los predicados ni las funciones XPath. Además, en el caso de la opción *Selector del esquema XML* solo se permiten expresiones en el eje secundario; para la opción *Campo del esquema XML* se permiten expresiones en el eje secundario y en el eje de atributos. Para más información consulte la recomendación XML Schema: Structures Recommendation de W3C.

Editar en el panel Expresión

Tenga en cuenta lo siguiente a la hora de editar expresiones en el panel Expresión:

- para crear una expresión de varias líneas (para una lectura más fácil) use la tecla **Entrar**.
- para aumentar/reducir el tamaño del texto en el campo de la expresión, haga clic en el campo de la expresión y después pulse la tecla **Ctrl** mientras hace girar la rueda de desplazamiento del ratón. Tenga en cuenta que el cambio en el tamaño del texto tiene lugar también en el panel *Resultados*.
- puede usar XMLSpy para introducir manualmente la expresión de la ruta de acceso de un nodo en lugar de hacerlo usted mismo. Para ello: (i) coloque el cursor del ratón en el punto de la expresión

© 2016-2022 Altova GmbH

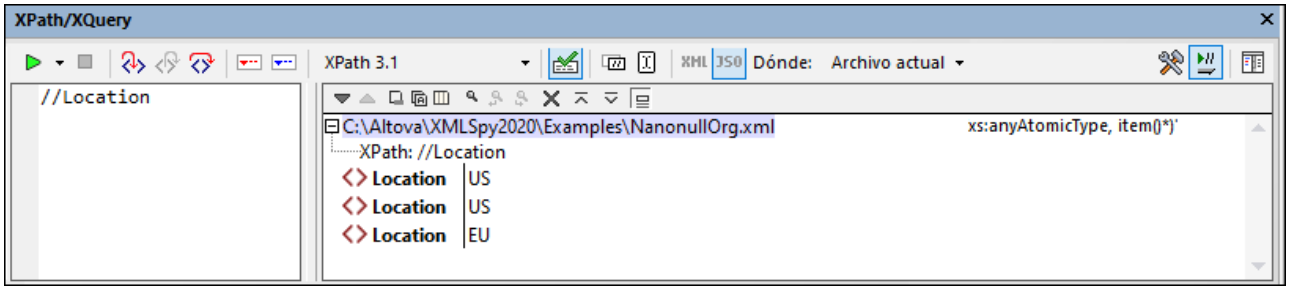
XPath donde quiere introducir la ruta de acceso; (ii) coloque el cursos dentro del nodo que quiere usar como destino; (iii) en la barra de herramientas haga clic en el botón **Copiar la XPath de la selección actual** para introducir en la expresión la ruta de acceso del nodo seleccionado. La ruta de acceso es absoluta y empieza en el nodo raíz del documento.

Evaluación XQuery y JSON

- para más información sobre evaluaciones XQuery consulte la sección Evaluar expresiones XQuery 514. (Los iconos XQ con para la evaluación XQuery 514; los iconos XQU son para ejecutar el comando Ejecutar XQuery/XQuery Update 519.)
- para una descripción de cómo hacer consultas en documentos JSON consulte Expresiones XQuery para JSON 593 y transformaciones JSON con XSLT/XQuery 706.

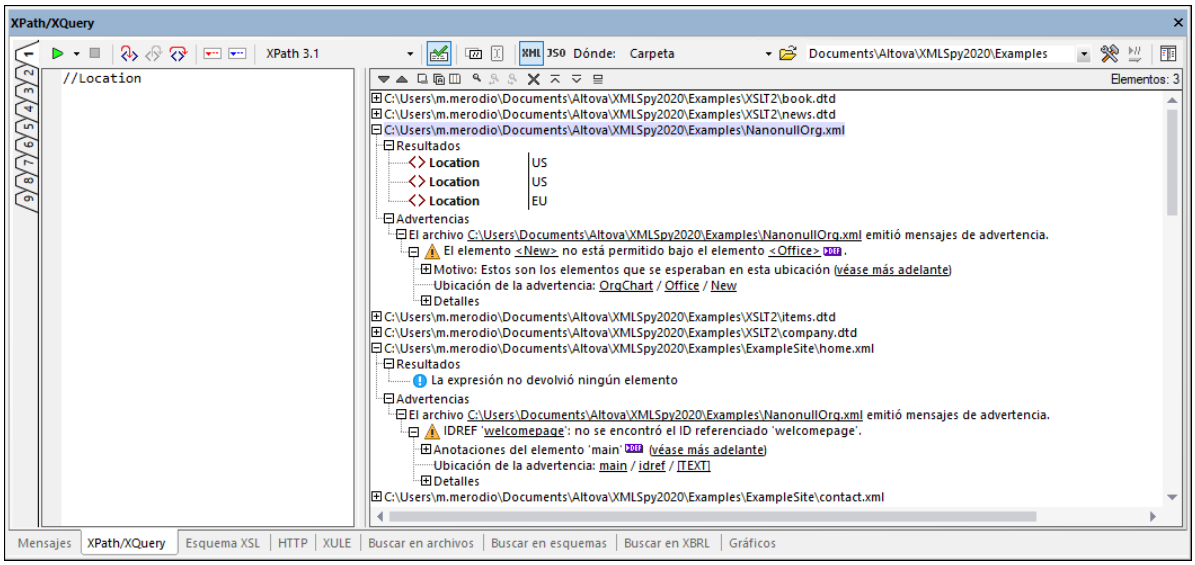
Panel Resultados

La imagen siguiente muestra el panel *Resultados*, a la derecha. Observe que este panel tiene su propia barra de herramientas.



En el panel Resultados puede llevar a cabo estas acciones:

- hacer clic en el icono **Mostrar encabezado en el resultado** si quiere que se vea la ubicación del archivo XML y la expresión XPath (*como en la imagen siguiente*).
- la lista de resultados consiste en dos columnas: (i) un nombre de nodo o tipo de datos y (ii) el contenido de ese nodo.
- si la expresión XPath devuelve nodos (como elementos o atributos), puede decidir si quiere que se muestre todo el contenido de un nodo como el valor de ese nodo activando el comando **Mostrar resultado completo**.
- si el resultado contiene un nodo (incluidos los nodos de texto), en lugar de literales generados por la expresión, al hacer clic en ese nodo en el panel *Resultados* se resalta el nodo correspondiente en el documento XML de la ventana principal.
- si la evaluación se hace en varios archivos (lo cual se indica en la opción *Dónde*), entonces los resultados de cada uno de los archivos se enumera por separado en la ruta de ese archivo (*imagen siguiente*). Si el modo de evaluación es XML, entonces se evalúan los archivos conforme a XML y el resto se omite. Si el modo de evaluación es JSON, entonces se evalúan los archivos conforme a JSON y se omite el resto.



- si los resultados incluyen varios archivos, al hacer clic en el nombre de un archivo en la lista de resultados, este se abre en XMLSpy y se convierte en el archivo activo.
- puede copiar las dos columnas de una sublínea del resultado o solamente la columna del valor. Para copiar todas las columnas haga clic con el botón derecho en una sublínea y active el comando **La copia incluye todas las columnas**. (Otra alternativa es activar/desactivar el comando con el icono de la barra de herramientas del panel *Resultados*) Después haga clic con el botón derecho en la sublínea que quiere copiar y seleccione el comando **Copiar sublínea** (para copiar solo esa sublínea) o **Copiar todo** (para copiarlas todas).

Barra de herramientas del panel Resultados

La barra de herramientas del panel *Resultados* contiene iconos para las funciones de navegación, búsqueda y copia. Estos iconos se describen en la siguiente tabla, empezando por el icono de la izquierda del todo. Los comandos correspondientes también están disponibles en el menú contextual de los elementos de la lista de resultados.

Icono	Acción
Siguiente, Anterior	Selecciona el elemento siguiente o anterior respectivamente de la lista de resultados.
Copiar la línea de texto seleccionada en el portapapeles	Copia la columna de valores del resultado seleccionado en el portapapeles. Para copiar todas las columnas debe activar el comando **La copia incluye todas las columnas** *(ver más abajo).*
Copiar todos los mensajes en el portapapeles	Copia la columna de valores de todos los resultados en el portapapeles, incluidos los valores que estén vacíos. Cada elemento se copia en una línea distinta.
La copia incluye todas las columnas	Alterna entre copiar todas las columnas y solamente la columna de valores. El separador de las columnas es un espacio.
Buscar	Abre el cuadro de diálogo "Buscar" para poder buscar cadenas, incluso caracteres especiales, en la lista de resultados.
Buscar anterior	Busca la instancia anterior del término de búsqueda introducido en el cuadro de

© 2016-2022 Altova GmbH

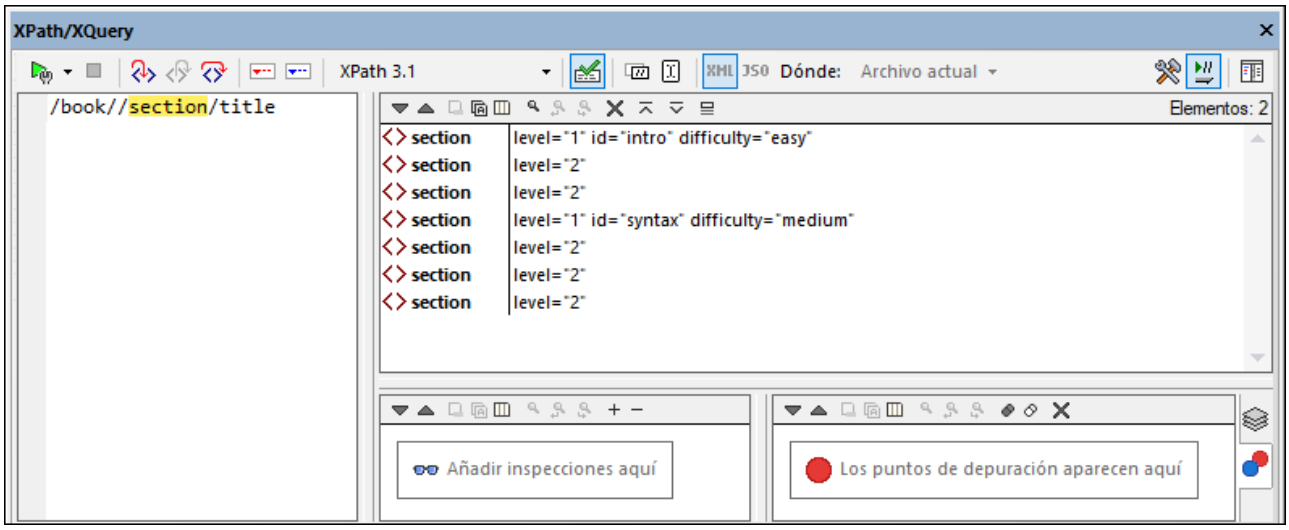
	diálogo "Buscar".
Buscar siguiente	Busca la instancia siguiente del término de búsqueda introducido en el cuadro de diálogo "Buscar".
Borrar	Borra la lista de resultados.
Contraer resultados multilínea en una sola línea	Si la columna de valores de un resultado contiene texto multilínea (texto con caracteres de línea nueva), puede alternar entre visualizar el texto en varias líneas o en una sola línea.
Mostrar resultado completo	Muestra todo el contenido de los nodos como el valor del nodo.
Mostrar encabezado en el resultado	Muestra la ubicación de los archivos XML y la expresión XPath.

10.3 Modo Depurador

Con el modo Depurador de la ventana XPath/XQuery (*imagen siguiente*) puede depurar expresiones XPath/XQuery con respecto al archivo activo.

En el modo Depurador se añaden otros dos paneles al panel *Resultados* (*imagen siguiente*).

- el panel *Pila de llamadas y puntos de depuración*, que se divide en dos pestañas.
- el panel *Variables y expresiones de inspección*; tanto las expresiones de inspección como las variables se muestran en el mismo panel.



Los paneles *Expresión* y *Resultados* se pueden colocar uno junto al otro en horizontal o en vertical. Para alternar entre estos diseños haga clic en el botón **Cambiar al diseño horizontal/vertical** (*en la parte superior derecha de la barra de herramientas de la ventana*).

Con el modo Depurador puede:

- acceder al proceso de evaluación XPath paso por paso para ver cómo se va evaluando la expresión XPath. Puede usar el botón **Paso a paso por instrucciones (F11)** para ello. En cada paso de la evaluación se resalta en amarillo la parte de la expresión que se está evaluando (*imagen anterior*) y el resultado de la evaluación de ese paso aparece en el panel *Resultados*. Por ejemplo, en la imagen anterior se han seleccionado todos los elementos `section` descendientes del elemento `book`.
- definir puntos de interrupción donde quiera que se detenga la evaluación para poder comprobar los resultados en esos puntos. Puede recorrer toda la evaluación deteniéndose solamente en los puntos de interrupción. Use el botón de la barra de herramientas **Iniciar el depurador (F5)** para ello. Esta opción es más rápida que la de detenerse en todos los pasos con **Paso a paso por instrucciones (F11)**.
- definir puntos de seguimiento que puede consultar después en un informe de los resultados de los pasos marcados como puntos de seguimiento. Con esta opción la evaluación no se detiene (excepto en los puntos de seguimiento) pero los resultados de los puntos de seguimiento aparecen en una lista en el panel *Resultados*.
- usar las expresiones de inspección para comprobar información (como datos de documentos o aspectos de la evaluación), lo cual resulta especialmente útil en los puntos de interrupción.

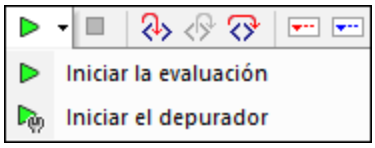
© 2016-2022 Altova GmbH

- mostrar las variables del alcance de la evaluación, incluyendo sus valores, en el panel *Variables y expresiones de inspección*.
- mostrar las llamadas realizadas por el procesador a un paso de la evaluación en la pestaña *Pila de llamadas* del panel *Pila de llamadas y puntos de depuración*.
- si ha definido puntos de interrupción y de seguimiento, entonces estos aparecen en la pestaña *Puntos de depuración* del panel *Pila de llamadas y puntos de depuración*.

Para más información sobre estas características consulte sus descripciones más abajo.

Configurar el modo Depurador

Para pasar al modo Depurador haga clic en **Iniciar el depurador** en el menú desplegable del comando **Iniciar evaluación/depuración (F5)**, que está en la parte superior izquierda de la barra de herramientas (*imagen siguiente*). Cuando se activa este modo aparecen os paneles *Variables y expresiones de inspección* y *Pila de llamadas y puntos de depuración*. Para iniciar el depurador de la expresión actual haga clic en **Iniciar el depurador** o pulse **F5**.



Tenga en cuenta que el modo Depurador solo funciona en el archivo activo y no se puede usar en varios archivos a la vez. Cuando se activa este modo, el valor de la opción *Dónde* cambia automáticamente a *Archivo actual*.

Botones para configurar el modo Depurador

	**Iniciar evaluación/el depurador (F5)**	Inicia el depurador
	**Cambiar al diseño vertical/horizontal**	Alternar entre el diseño horizontal y el vertical
	**Cambiar al modo generador**	Cambia al modo generador de expresiones, cuyos ayudantes de entrada contextuales ayudan a construir expresiones
	**Copiar la XPath de la selección actual**	Copia la ruta de acceso del nodo del documento XML en la última posición del cursor en el panel Expresión
	**Modo de evaluación XML/JSON**	El icono que esté resaltad de los dos es la opción activa. El modo de evaluación se selecciona automáticamente en función del tipo de archivo; en ese caso, el otro icono está deshabilitado.
	**Establecer la selección actual como contexto**	Alterna el contexto de la expresión entre el nodo raíz y la selección actual
	**Evaluar al teclear**	Este modo va evaluando las expresiones según las va tecleando

	**Validar XML**	Al activar esta opción se validan los documentos XML de destino

Ejecutar el depurador

A grandes rasgos, estos suelen ser los pasos que se siguen para depurar una expresión XPath/XQuery:

1. Convertir en activo el archivo XML/JSON en el que quiere ejecutar la expresión.
2. Introducir la expresión XPath/XQuery en el panel *Expresión*.
3. Establecer los puntos de interrupción y de seguimiento que quiera. Un punto de interrupción es un punto en el que se detiene la evaluación. Un punto de seguimiento es un punto en el que la evaluación queda grabada, lo que permite rastrear los resultados de la evaluación.
4. Si hace clic en **Iniciar el depurador**, la evaluación se ejecutará en un solo paso hasta el final, a no ser que haya marcado algún punto de interrupción en la expresión. Haga clic varias veces en **Iniciar el depurador** para avanzar entre los puntos de interrupción hasta el final de la evaluación.
5. Use la función **Paso a paso por instrucciones/Paso a paso para salir/Paso a paso por procedimientos** para avanzar por la evaluación paso a paso.

Botones para depurar

	**Iniciar el depurador (F5)**	Inicia el depurador. La evaluación va directamente al final y solo se detiene en los puntos de interrupción, si los hay.
	**Detener el depurador (Mayús+F5)**	Sale del modo Evaluador y detiene el depurador
	**Paso a paso por instrucciones (F11)**	Avanza por la evaluación, paso a paso. Se trata de un avance paulatino por la evaluación.
	**Paso a paso para salir (Mayús+F11)**	Sale del paso de evaluación actual y procede al paso progenitor
	**Paso a paso por procedimientos (Ctrl+F11)**	Recorre los pasos descendientes
	**Insertar o quitar punto de interrupción (F9)**	Inserta/elimina un punto de interrupción en el paso de la expresión en el que coloque el cursor
	**Insertar o quitar punto de seguimiento (Mayús+F9)**	Inserta/elimina un punto de seguimiento en el paso de la expresión en el que coloque el cursor

Paso a paso por instrucciones, para salir y por procedimientos

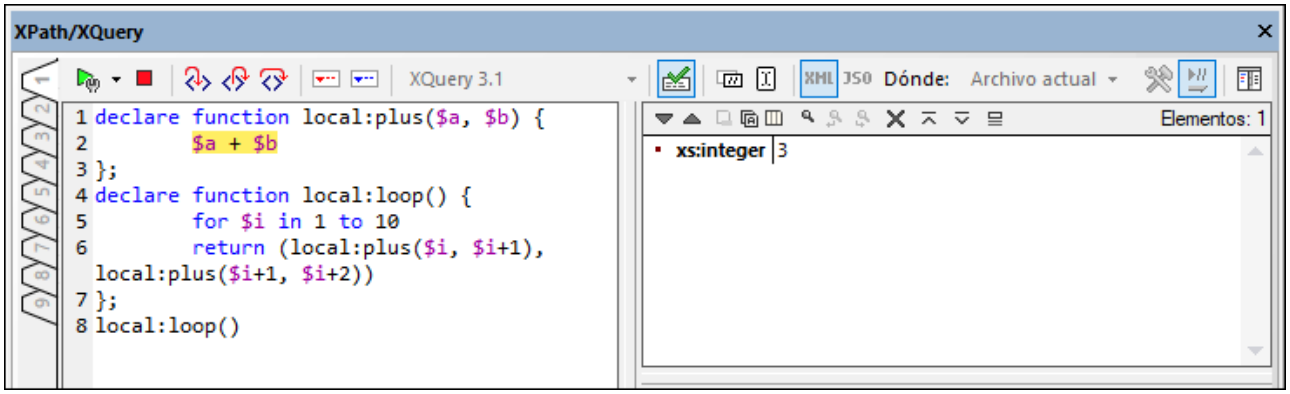
Con **Paso a paso por instrucciones** puede avanzar paso a paso por la evaluación. Cada vez que hace clic en este comando pasa al siguiente paso de la evaluación; el paso actual aparece resaltado en la expresión (*imagen siguiente*). El comando **Paso a paso para salir** va al paso de un nivel superior al del paso actual, mientras que el comando **Paso a paso por procedimientos** se salta los pasos de niveles inferiores y va directamente al siguiente paso del mismo nivel que el actual. Para probar estos comandos puede copiar la expresión XQuery 3.1 de más abajo en el panel Expresión y hacer clic en los tres botones de este comando para ver cómo funcionan.

© 2016-2022 Altova GmbH

- *Expresión XQuery 3.1 para probar las funciones Paso a paso por instrucciones, para salir y por procedimientos*

```
declare function local:plus($a, $b) {
        $a + $b
};
declare function local:loop() {
        for $i in 1 to 10
        return (local:plus($i, $i+1), local:plus($i+1, $i+2))
};
local:loop()
```

La imagen siguiente muestra la evaluación cuando se ha pausado el procesamiento al alcanzar el paso de suma `$a + $b` al pasar por primera vez por el bucle, es decir, cuando `$i=1`. En este paso de suma el resultado muestra `3` (como consecuencia de `1+2`).



Puntos de interrupción

Los puntos de interrupción son puntos en los que quiere que el Depurador se detenga después de haberlo iniciado con el comando **Iniciar el depurador**. Son útiles si tiene que analizar una parte concreta de la expresión. Cuando el depurador se detiene en el punto de interrupción, puede comprobar el resultado y podría usar la función **Paso a paso por instrucciones** para mostrar los resultados de los pasos siguientes de la evaluación. Para definir un punto de interrupción coloque el cursor en la expresión, en el punto en el que quiere introducir el punto de interrupción, y haga clic en el botón de la barra de herramientas **Insertar o quitar punto de interrupción (F9)**. El punto de interrupción se marcará con una línea roja discontinua. Para eliminar el punto de interrupción basta con seleccionarlo y volver a hacer clic en el comando **Insertar o quitar punto de interrupción (F9)**.

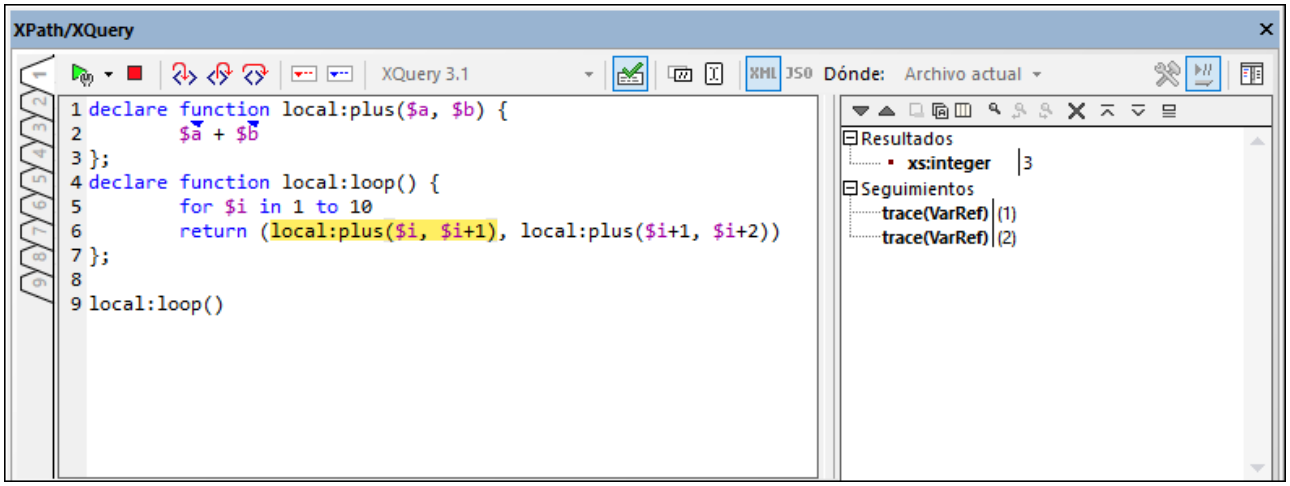
Véase también Puntos de depuración 587, más abajo.

Puntos de seguimiento

Los puntos de seguimiento son puntos en los que se graban los resultados. Estos resultados aparecen en la estructura *Seguimientos* de la pestaña *Resultados* (*imagen siguiente*). Esto permite ver todos los resultados de la evaluación en partes concretas de la expresión. Por ejemplo, en la imagen siguiente los puntos de seguimiento se pusieron en `$a`, `$b`, y `local:plus($i, $i+1)`; los resultados que corresponden a estos

puntos de seguimiento en la primera iteración del bucle aparecen en la estructura *Seguimientos*: `$a=1`, `$b=2`, and `local:plus($i, $i+1)=3`.

Para definir un punto de seguimiento coloque el cursor en el punto que quiera y haga clic en el botón de la barra de herramientas **Insertar o quitar punto de seguimiento (Mayús+F9)**.

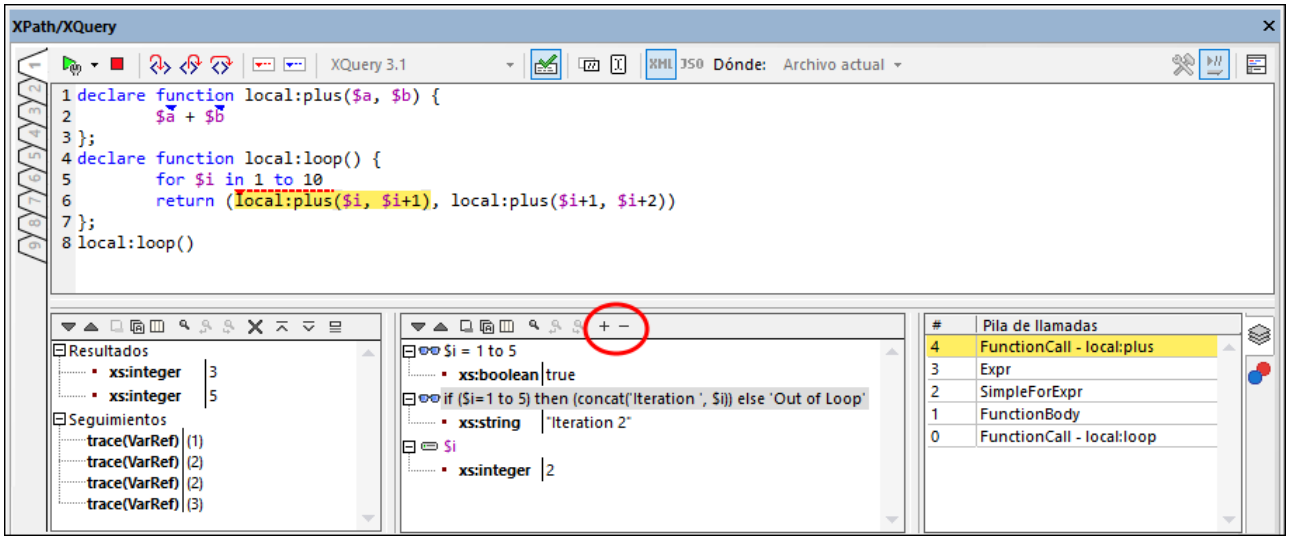


Nota: si en algún punto de la expresión hay un punto de interrupción y un punto se seguimiento, entonces la línea discontinua alterna los colores rojo y azul.

Véase también Puntos de depuración 587, más abajo.

Variables, expresiones de inspección y pila de llamadas

Las variables y las expresiones de inspección se muestran en el panel *Variables y expresiones de inspección* (*panel inferior en el centro en la imagen siguiente*).



 © 2016-2022 Altova GmbH

Variables
Las variables que se han declarado en la expresión y que están dentro del alcance del paso actual de la evaluación aparecen con sus valores correspondientes. Por ejemplo, en la imagen anterior el procesamiento ha alcanzado la llamada a la función **local:plus**. La variable **$i** está dentro del alcance de la función **local:loop** y acaba de incrementarse a **2**, por lo que **$i** se muestra con su valor actual. Las variables **$a** y **$b** no están en **local:loop**, por lo que tampoco entran dentro del alcance de la evaluación; en consecuencia, no se muestran.

Expresiones de inspección
Las expresiones de inspección son expresiones que puede introducir antes de que comience la evaluación o en un momento en que esta se detenga. Este tipo de expresiones se pueden usar para:

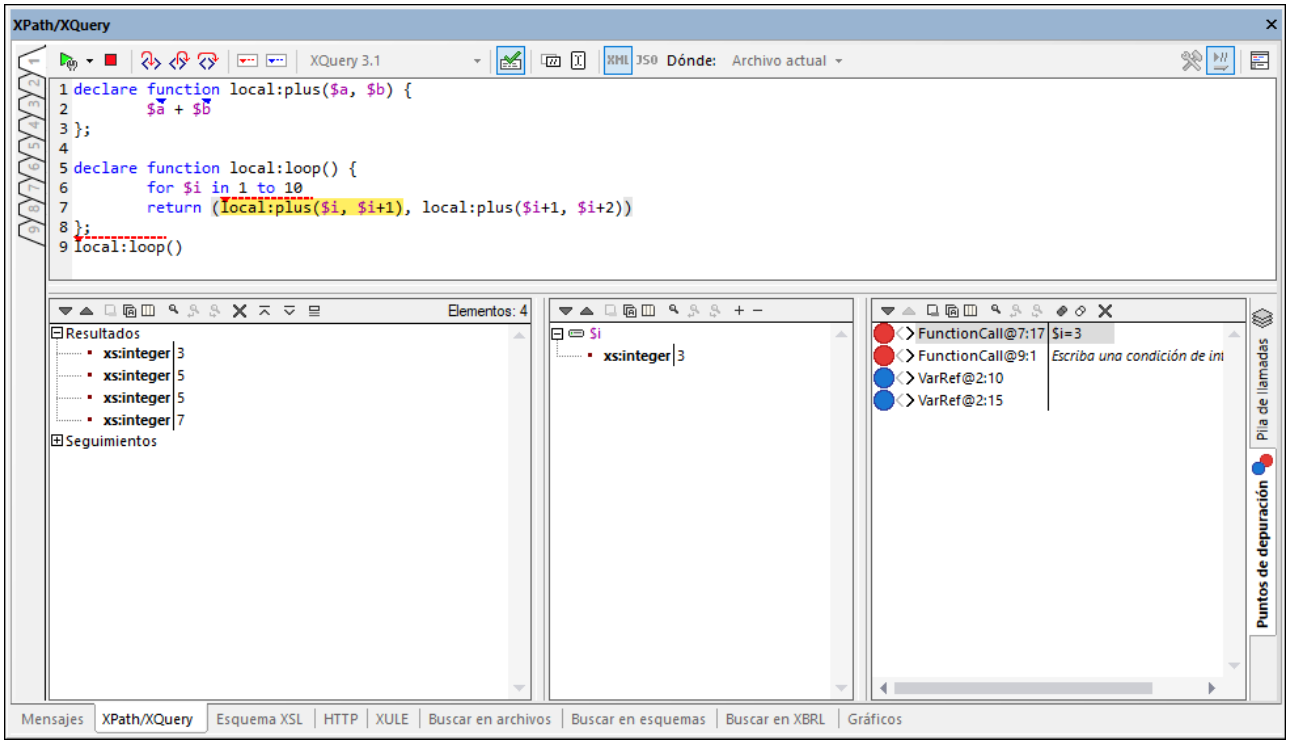
- probar ciertas condiciones. Por ejemplo, en la imagen anterior la expresión de inspección **$i=1 to 5** se usa para probar si la variable **$i** tiene un valor dentro del rango dado. El resultado `true` nos dice que este criterio se cumple.
- encontrar datos dentro de un contexto concreto. Por ejemplo, dentro del contexto de un elemento `Company` podríamos introducir la expresión de inspección **@id** para buscar el código de cliente de esa empresa en el documento XML de destino.
- generar datos adicionales. Por ejemplo, en la imagen anterior, la expresión de inspección **if ($i=1 to 5) then (concat("Iteration ", $i)) else "Out of Loop"** puede generar una cadena de texto que indique en qué iteración del bucle se encuentra en ese momento la evaluación.

Para introducir una expresión de inspección haga clic en **Agregar entrada de inspección** en la barra de herramientas del panel (*dentro del círculo rojo en la imagen anterior*), introduzca la expresión y pulse la tecla **Entrar** para terminar. Para quitar una expresión de inspección, selecciónela y haga clic en **Quitar la entrada de inspección seleccionada** en la barra de herramientas. Si durante la depuración la expresión no se puede evaluar correctamente por alguna razón (por ejemplo, si una de sus variables no entra dentro del alcance), entonces la expresión de inspección aparece en rojo.

Pila de llamadas
La pestaña *Pila de llamadas* del panel *Pila de llamadas y puntos de depuración* (*panel inferior derecho en la imagen anterior*) muestra las llamadas del procesador hasta ese punto de la depuración. La llamada actual del procesador está resaltada en amarillo. Tenga en cuenta que solo se muestran las llamadas que han llevado directamente al paso de evaluación actual. Por ejemplo, en la imagen anterior el paso de evaluación actual es una llamada a la función **local:plus**. Aunque se trata de la segunda iteración de **local:loop**, las llamadas del procesador de la primera iteración **no** aparecen porque están en un nivel paralelo al de la llamada actual a la función y no han llevado directamente a ella.

Puntos de depuración

La pestaña *Puntos de depuración* del panel *Pila de llamadas y puntos de depuración* (*panel inferior derecho en la imagen anterior* muestra los puntos de interrupción (círculos rojos) y los puntos de seguimiento (puntos azules) que haya colocado en la expresión. Cada punto de depuración aparece en la lista con su número de línea y de carácter. Por ejemplo, **FunctionCall@7:13** significa que hay un punto de depuración en la línea 7, carácter 13.



Algunas de las funciones que ofrece este modo son:

- En el caso de los puntos de interrupción, puede introducir una condición de interrupción; para ello haga doble clic en *Insertar condición de interrupción*, introduzca la expresión de la condición y pulse la tecla **Entrar**. Ese punto de interrupción se habilita solo si la condición evalúa en `true`. Por ejemplo, en la imagen anterior, la condición de interrupción `$i=3` habilita el punto de interrupción en esta llamada a la función solo si el valor de `$i` es `3`. La imagen muestra la evaluación pausada en este punto de interrupción.
- Puede activar o desactivar todos los puntos de depuración haciendo clic en los botones correspondientes de la barra de herramientas: **Habilitar todos los puntos de depuración** y **Deshabilitar todos los puntos de depuración** (*dentro de los círculos verdes en la imagen anterior*). Cuando se deshabilita un punto de depuración, este se desactiva para todas las evaluaciones hasta que se vuelva a habilitar.
- Puede habilitar/deshabilitar puntos de interrupción individuales en sus correspondientes menús contextuales.

Comandos de la barra de herramientas en los paneles

Los paneles de la ventana XPath/XQuery en el modo Depurador (*imagen anterior*) contienen botones para las funciones de navegación, búsqueda y copia. Estos botones se describen en la siguiente tabla, empezando por el botón de la izquierda del todo. Los comandos correspondientes también están disponibles en el menú contextual de los elementos de la lista de resultados.

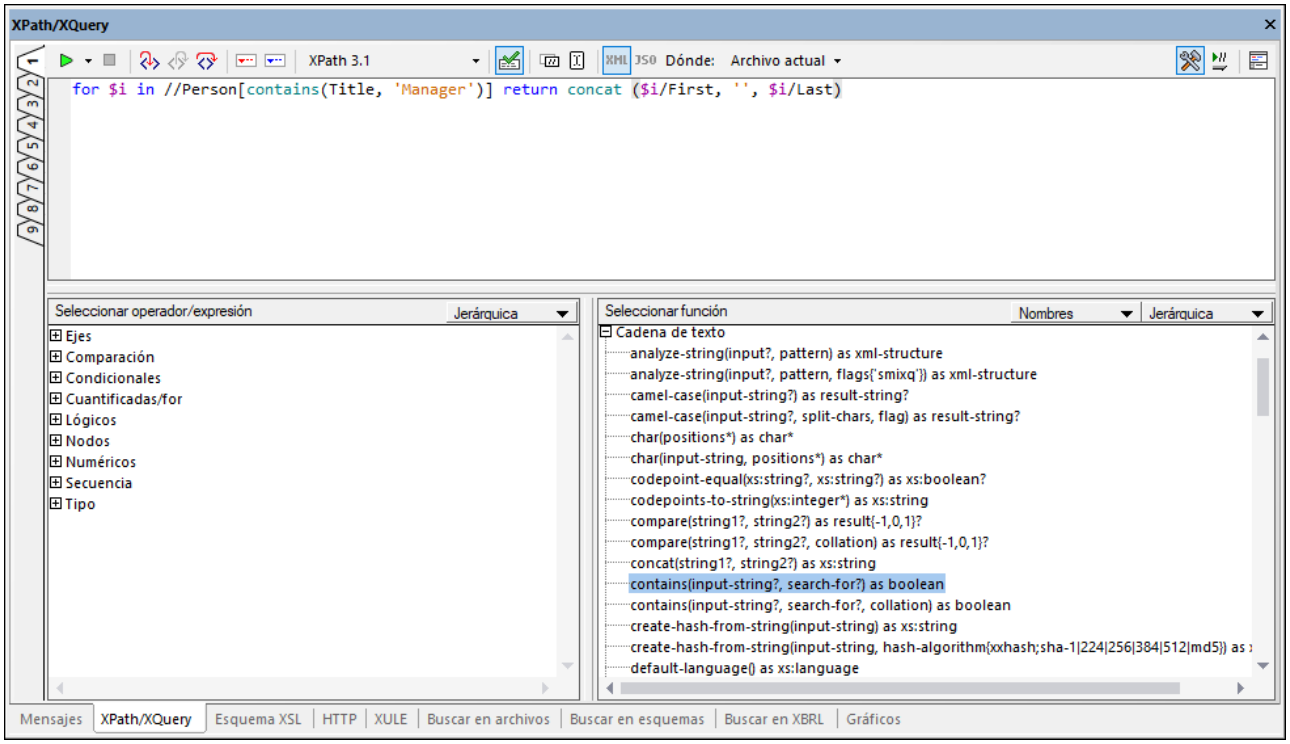
Icono	Acción
Siguiente, Anterior	Selecciona el elemento siguiente o anterior respectivamente de la lista de

© 2016-2022 Altova GmbH

	resultados.
Copiar la línea de texto seleccionada en el portapapeles	Copia la columna de valores del resultado seleccionado en el portapapeles. Para copiar todas las columnas debe activar el comando **La copia incluye todas las columnas** *(ver más abajo).*
Copiar todos los mensajes en el portapapeles	Copia la columna de valores de todos los resultados en el portapapeles, incluidos los valores que estén vacíos. Cada elemento se copia en una línea distinta.
La copia incluye todas las columnas	Alterna entre copiar todas las columnas y solamente la columna de valores. El separador de las columnas es un espacio.
Buscar	Abre el cuadro de diálogo "Buscar" para poder buscar cadenas, incluso caracteres especiales, en la lista de resultados.
Buscar anterior	Busca la instancia anterior del término de búsqueda introducido en el cuadro de diálogo "Buscar".
Buscar siguiente	Busca la instancia siguiente del término de búsqueda introducido en el cuadro de diálogo "Buscar".
Borrar	Borra la lista de resultados.
Contraer resultados multilínea en una sola línea	Si la columna de valores de un resultado contiene texto multilínea (texto con caracteres de línea nueva), puede alternar entre visualizar el texto en varias líneas o en una sola línea.
Mostrar resultado completo	Muestra todo el contenido de los nodos como el valor del nodo.
Mostrar encabezado en el resultado	Muestra la ubicación de los archivos XML y la expresión XPath.

10.4 Generador de expresiones

El generador de expresiones (o modo Generador) se activa o desactiva haciendo clic en el icono **Cambiar al modo generador** de la barra de herramientas principal  (*imagen siguiente*). El Generador de expresiones se puede activar tanto en modo modo Evaluador 574 como en modo Depurador 574. Este generador cuenta con dos ayudantes de entrada, cada uno con su panel: (i) para operadores y expresiones y (ii) para funciones (*imagen siguiente*). Los elementos de ambos paneles se pueden mostrar ordenados jerárquicamente o como una lista simple. Seleccione la opción que prefiera en la lista desplegable que hay en la parte superior de cada panel (*imagen siguiente*). En la imagen siguiente los dos paneles muestran sus elementos ordenados de forma jerárquica.



Características del Generador de expresiones:

- para ver la descripción de un elemento en cualquiera de los paneles de los ayudantes de entrada, pase el cursor del ratón por encima de ese elemento.
- cada función aparece en la lista con su firma (es decir, con sus argumentos, los tipos de datos de los argumentos y el tipo de datos de los resultados de la función).
- si un único nombre de función tiene más de una firma, cada firma aparece en la lista como una función distinta. (Estas variantes se conocen como sobrecargas de ese nombre de función.) En la imagen anterior, por ejemplo, la función `contains` aparece dos veces: una por cada una de sus dos firmas.
- los argumentos aparecen en la lista ordenados por nombre (si tienen uno) o por tipo de datos. Seleccione la opción que prefiera en la lista desplegable en la barra del título del panel Funciones (*dentro del círculo verde en la imagen anterior*).

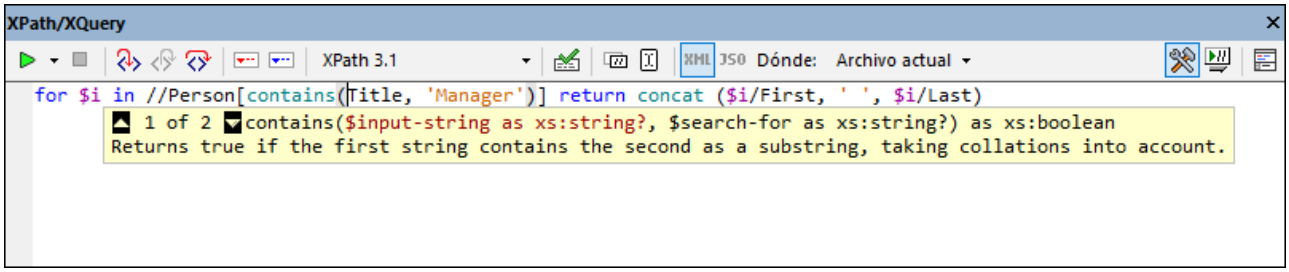
© 2016-2022 Altova GmbH

- al hacer doble clic en un elemento en cualquiera de los paneles (operador, expresión o función) se inserta ese elemento en el punto en que se encuentre el cursor del ratón en la expresión. Al insertar las funciones, se indican sus argumentos con marcadores de posición (con el símbolo **#**).
- si (i) se selecciona texto en el campo de edición de la expresión y (ii) se hace doble clic en un operador, una expresión o una función que contiene un marcador de posición, entonces el texto seleccionado se inserta en lugar de ese marcador de posición. Esta es una forma rápida de insertar texto largo (como una expresión de ruta de acceso) en un operador, una expresión o una función.
- Puede insertar la ruta en un nodo en el documento activo seleccionando ese nodo y después haciendo clic en el botón de la barra de herramientas de la ventana principal.
- Puede insertar la ruta de acceso a un nodo en el documento activo seleccionando ese nodo y haciendo clic en el botón **Copiar la XPath de la selección actual** de la barra de herramientas principal de la ventana.

Botones de la barra de herramientas del generador de expresiones

	**Cambiar al diseño vertical/horizontal**	Alternar entre el diseño horizontal y el vertical
	**Cambiar al modo generador**	Cambia al modo generador de expresiones, cuyos ayudantes de entrada contextuales ayudan a construir expresiones
	**Copiar la XPath de la selección actual**	Copia la ruta de acceso del nodo del documento XML en la última posición del cursor en el panel Expresión
	**Establecer la selección actual como contexto**	Alterna el contexto de la expresión entre el nodo raíz y la selección actual
	**Cargar/Guardar snippet de XPath/XQuery**	Carga o guarda un snippet de XPath/XQuery de/en un archivo XQuery

Tras insertar una función en la expresión, pase el puntero por encima del nombre de la función para ver su firma y una descripción. Si una función tiene varias firmas, esto se señala en la parte inferior de la pantalla con un factor de sobrecarga. Si coloca el cursor entre los paréntesis de la función y pulsa **Ctrl+Mayús+Barra espaciadora**, podrá ver las firmas de la sobrecarga de funciones (*imagen siguiente*).



Abrir y guardar snippets de XPath/XQuery de/en archivos

Puede guardar expresiones XPath/XQuery o recortes más largos que haya usado en la ventana XPath/XQuery, junto con la configuración de la ventana, en un archivo XQuery, y también cargar snippets, también llamados recortes, de XPath/XQuery desde un archivo XQuery. Para ello utilice los iconos correspondientes (Cargar/Guardar recorte) de la barra de herramientas de la ventana XPath/XQuery.



Una vez haya guardado el snippet en un archivo, puede volver a cargarlo en la ventana XPath/XQuery de cualquier instancia de XMLSpy (a partir de la versión 2022). Esto puede ser útil si quiere usar ese snippet en otro equipo, pasárselo a otro usuario o simplemente usarlo más tarde en ese mismo equipo. Cuando se carga la expresión en la ventana XPath/XQuery desde un archivo XQuery, la configuración de la ventana cambia automáticamente a la configuración que tenía la ventana cuando se guardó el snippet.

Guardar un snippet en un archivo
Para guardar una expresión XPath/XQuery o un snippet en un archivo siga estos pasos:

1. Defina la configuración de la ventana: (i) el lenguaje de evaluación; (ii) el campo *Dónde* y (iii) en caso de que en lugar de con *Dónde* esté trabajando con *Proyecto*, defina si las carpetas externas se deben omitir o no.
2. Introduzca la expresión XPath/XQuery o el snippet que quiere guardar.
3. Haga clic en **Guardar snippet**.
4. Cuando aparezca el cuadro de diálogo *Guardar como*, seleccione el archivo o introduzca el nombre de un archivo nuevo en el que se guardará el snippet, y haga clic en **Guardar**.

El snippet se guarda en el archivo y la configuración de la ventana aparece como comentario en la primera línea del archivo. A continuación puede ver un ejemplo.

```
(: {"language":"XQuery3","where":"CurrentFile"} :)
for $i in //para[count(*)=0] return $i
```

Nota: puede editar este archivo, así como el comentario que contiene la configuración de la ventana.

Cargar un snippet desde un archivo
Para cargar un snippet de XPath/XQuery desde un archivo:

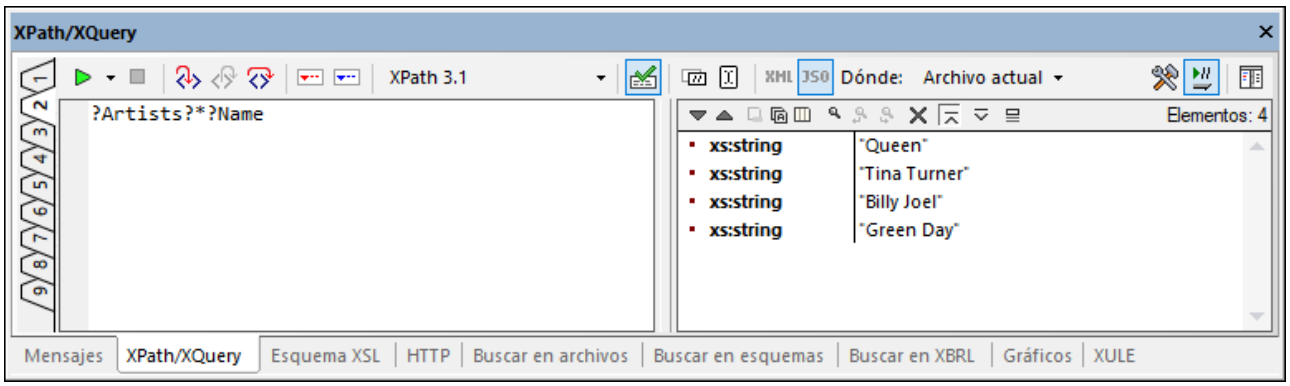
1. En la pestaña donde quiere cargar la expresión haga clic en **Cargar snippet**.
2. En el cuadro de diálogo *Abrir* que aparece navegue hasta el archivo XQuery desde el que quiere cargar el snippet y haga clic en **Abrir**.

Así carga el snippet y la configuración de la ventana XPath/XQuery cambiará a la que estuviera guardada en el archivo. Si no había ninguna configuración guardada estas opciones no cambian. Si la opción *Dónde* no se definió correctamente se usará la opción por defecto *Archivo actual*.

© 2016-2022 Altova GmbH

10.5 Expresiones XQuery para JSON

Para hacer consultas en un documento JSON introduzca una expresión de consulta XPath/XQuery 3.1 en la ventana de resultados XPath/XQuery 127 (*imagen siguiente*).



Para evaluar una expresión en un documento JSON siga estos pasos:

1. Seleccione el icono **XPath 3.1** o el icono **XQuery 3.1**.
2. Asegúrese de que la ventana está en modo de evaluación JSON.
3. Introduzca la expresión XPath 3.1 o XQuery 3.1.
4. Haga clic en **Iniciar evaluación** (*a la izquierda en la barra de herramientas*).

Expresiones XQuery 3.1 para JSON

Las estructuras de datos JSON suelen usar objetos y matrices y usan el operador de búsqueda XQuery 3.1 `?`, que se usa para ubicar nodos dentro de objetos (que básicamente son asignaciones desde la perspectiva de XQuery) y matrices JSON. Esta forma de ubicar nodos es distinta a la que usa expresiones de ruta de acceso para ubicar nodos en documentos XML. Es este caso, el operador barra `/` se usa para conectar pasos en una expresión de ruta de acceso (por ejemplo: `items/*`). En las expresiones XQuery para JSON, el operador barra no se usa para ubicar nodos.

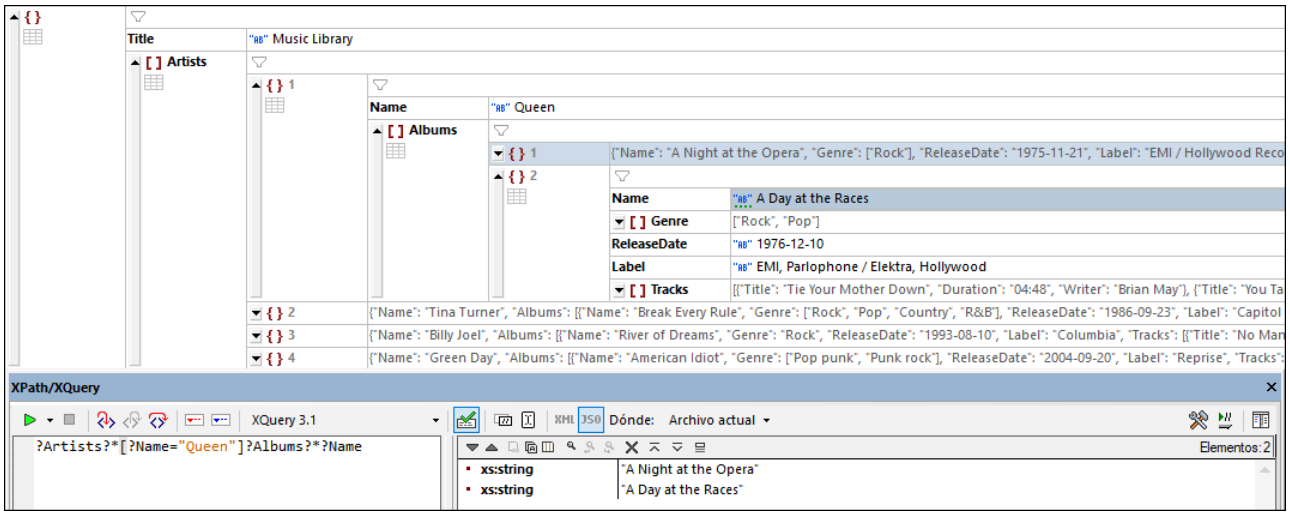
Ejemplos de expresiones XQuery para JSON

`?items?*`
Significa: buscar el nodo secundario **`items`** y después buscar todos sus nodos secundarios. Observe que se espera que **`items`** sea un nodo secundario del nodo de contexto.

`?Artists?1?Albums?2?Name`
Significa: buscar el nodo secundario **`Artists`** y después buscar su primer nodo secundario. Dentro de ese nodo, buscar el nodo secundario **`Albums`** y después buscar el segundo nodo secundario de este. Devolver el nodo **`Name`** de ese segundo nodo secundario.

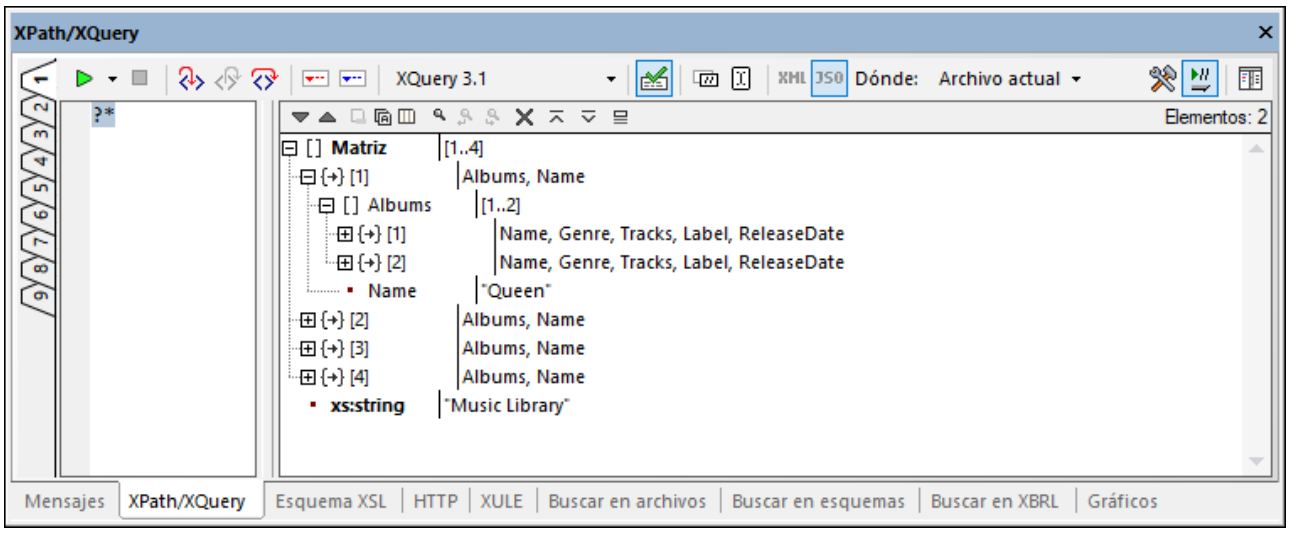
`?Tracks?*[contains(?Writer, 'Brian')]`
Significa: buscar el nodo secundario **`Tracks`** y después buscar todos sus secundarios. Mientras se buscan los secundarios, buscar los nodos secundarios de cada secundario **`Writer`** y seleccionar solamente los que contengan la cadena de texto **`'Brian'`**. Observe que en esta expresión hay tres operadores de búsqueda. Cada uno se usa en un paso nuevo en el que se busca en un conjunto de nodos.

`?Artists?*[?Name="Queen"]?Albums?*?Name`
Significa: dentro del objeto raíz, buscar el nodo secundario **`Artists`** y después buscar todos sus secundarios cuyo nodo **`Name`** tiene el valor **`"Queen"`**. Dentro de estos nodos, buscar todos los nodos secundarios **`Album`** y sus secundarios. Dentro de esos secundarios, buscar (y devolver) los respectivos nodos **`Name`**.En la imagen siguiente esta expresión se muestra en la ventana XPath/XQuery [573] junto con la representación de la vista Cuadrícula JSON [679] del documento JSON de destino.



Resultados para la evaluación JSON

En la parte izquierda del panel *Resultados* se muestran los componentes JSON seleccionados en negrita y el valor de cada componente aparece en la parte derecha del panel. En la imagen siguiente se muestran los resultados en negrita. La matriz está expandida para que se vea su contenido.



Los mapas y las matrices se muestran en un formato más o menos abreviado en función de si el icono de la barra de herramientas *Mostrar resultado completo* está activado o desactivado. Para expandir los mapas y las matrices de la parte izquierda del panel haga clic en sus respectivos iconos del signo "+".

 © 2016-2022 Altova GmbH

10.6 Aspectos importantes

Expresiones XPath 1.0

- Las funciones XPath 1.0 no pueden incluir el prefijo del espacio de nombres.
- Se admiten los cuatro tipos de pruebas de nodos: `node()`, `text()`, `comment()` y `processing-instruction()`.

Expresiones XPath 2.0 y 3.1

- Se admite el uso de literales de cadena (p. ej. "Hola") y literales numéricos (p. ej. 256). Para crear otros literales basados en tipos de XML Schema, se utiliza un constructor con el prefijo del espacio de nombres (p. ej. `xs:date('2004-09-02')`). El prefijo del espacio de nombres que utilice para tipos de XML Schema debe estar enlazado al espacio de nombres de XML Schema: `http://www.w3.org/2001/XMLSchema`. Este espacio de nombres debe estar declarado en su archivo XML.
- Las funciones XPath 2.0 y 3.0 utilizadas por el evaluador de XPath pertenecen al espacio de nombres `http://www.w3.org/2005/xpath-functions`. Por convención, el prefijo `fn:` está enlazado a este espacio de nombres. Sin embargo, puesto que este espacio de nombres es el espacio de nombres de funciones predeterminado utilizado por el evaluador de XPath, no es necesario especificar un prefijo en las funciones. Si utiliza un prefijo, asegúrese de que el prefijo está enlazado al espacio de nombres de XPath Functions, que se debe declarar en el documento XML. Ejemplos de cómo usar funciones: `current-date()` (sin declarar el espacio de nombres de Funciones en el documento XML); `fn:current-date()` (sin declarar el espacio de nombres de Funciones en el documento XML o con el espacio de nombres declarado en el documento XML y enlazado al prefijo `fn:`). Puede omitir el prefijo del espacio de nombres incluso si el espacio de nombres de Funciones se declaró en el documento XML con o sin un prefijo. Esto se debe a que en el espacio de nombres predeterminado (el espacio de nombres predeterminado para funciones) hay una función muy usada en una expresión XPath.
- Las extensiones XPath de Altova están en el espacio de nombres **`http://www.altova.com/xslt-extensions`**.

Nota de resumen: si utiliza constructores o tipos del espacio de nombres de XML Schema, es necesario declarar el espacio de nombres de XML Schema en el documento XML y usar los prefijos del espacio de nombres correctos en la expresión XPath. No es necesario usar un prefijo para las funciones XPath.

Tipos de datos en XPath 2.0 y 3.1

Si está evaluando una expresión XPath 2.0 o 3.1 para un documento XML que hace referencia a un esquema XML y es válido conforme a ese esquema, es necesario construir o convertir los tipos de datos que no se convierten implícitamente en el tipo de datos necesario mediante una operación. En los modelos de datos XPath 2.0 y 3.1 usado por el motor XPath integrado, a todos los valores de nodo atomizados desde el documento XML se les asigna el tipo de datos `xs:untypedAtomic`. El tipo `xs:untypedAtomic` funciona bien con conversiones de tipos implícitas. Por ejemplo, la expresión `xs:untypedAtomic("1") + 1` da como resultado un valor de 2 porque el valor `xs:untypedAtomic` se asciende implícitamente al tipo `xs:double` mediante el operador de suma. Los operadores aritméticos ascienden implícitamente los operandos a `xs:double`. Los operadores de comparación ascienden los operandos a `xs:string` antes de realizar la comparación.

En algunos casos, no obstante, es necesario convertir los tipos de datos explícitamente al tipo de datos necesario. Por ejemplo, si tiene los dos elementos `startDate` y `endDate`, que en el esquema XML se definieron para tener el tipo `xs:date`, recibirá un mensaje de error si utiliza la expresión XPath 2.0 `endDate - startDate`, por ejemplo. Por otro lado, si usa `xs:date(endDate) - xs:date(startDate)` o `(endDate cast as xs:date) - (startDate cast as xs:date)`, la expresión dará como resultado una secuencia singleton de tipo `xs:dayTimeDuration`.

Nota: Los motores XPath que utiliza el evaluador de XPath también son utilizados por el motor XSLT de Altova. Por tanto, las expresiones XPath 2.0 o 3.1 en hojas de estilos XSLT que no se conviertan implícitamente al tipo de datos necesario se deben construir o convertir explícitamente al tipo de datos necesario.

Longitud de cadena en referencias de caracteres y entidad

Cuando se utilizan referencias de caracteres y entidad como cadenas de entrada para la función `string-length()`, las referencias no se pueden resolver y se devuelve la longitud de la cadena de texto no resuelta. Dentro del entorno XSLT, no obstante, estas referencias tendrían significado y se devuelve la longitud de la cadena resuelta.

Compatibilidad con funciones XPath 2.0 y 3.1

Consulte los anexos 2120 de esta documentación.

© 2016-2022 Altova GmbH

11 Authentic

La vista Authentic (*imagen siguiente*) ofrece una representación gráfica de sus documentos XML. Esta vista permite ver los documentos XML sin marcado e incluye funciones de formato y entrada de datos, como campos de entrada de datos y botones de opción. Los datos introducidos por el usuario en la vista Authentic se introducen también en el archivo XML.



Para poder ver y editar un documento XML en la vista Authentic, es necesario que el documento XML tenga asociada una hoja de estilos **SPS** (**StyleVision Power Stylesheet**), que se crea con la aplicación StyleVision de Altova. Una SPS (archivo `.sps`) es básicamente una hoja de estilos XSLT que especifica una presentación de salida para un archivo XML que puede incluir mecanismos de entrada de datos. Por tanto, los usuarios de la vista Authentic pueden escribir datos directamente en el archivo XML o en la BD. Una SPS se basa en un esquema y le "pertenece". Si desea usar una SPS para editar un archivo XML en la vista Authentic, la SPS debe estar basada en el mismo esquema que el archivo XML.

Trabajar con la vista Authentic

- Si está abierto un archivo XML, puede cambiar a la vista Authentic pulsando el botón Authentic situado en la parte inferior de la ventana principal. Si el archivo XML no tiene asignada una SPS, se le solicitará que le asigne una. La SPS debe estar basada en el mismo esquema que el archivo XML.
- Para crear un archivo nuevo y verlo en la vista Authentic seleccione el comando **Archivo | Nuevo** y haga clic en en el botón Seleccionar una hoja de estilos de StyleVision. Este nuevo archivo es una plantilla asociada a la SPS seleccionada. La plantilla puede tener algunos datos iniciales. Estos datos iniciales están en un archivo XML (un archivo de plantilla XML) que se pueden asociar o no a la SPS.

Después, cuando el archivo XML se presente en la vista Authentic, podrá introducir los datos en el archivo y guardarlo.

- También puede abrir una SPS con el comando **Authentic | Documento nuevo**. Si se asignó un archivo de plantilla XML a la SPS, los datos del archivo de plantilla XML se usan como datos iniciales de la plantilla de documento XML creada en la vista Authentic.

Contenido de esta sección

Esta sección incluye un tutorial que le enseñará a usar la vista Authentic y una descripción detallada sobre las diferentes funciones de edición de la vista Authentic.

Más información sobre la vista Authentic

Para más información sobre la vista Authentic, consulte (i) el apartado Vistas de edición | Vista **Authentic** 597 de este manual, que describe la ventana de edición de la vista Authentic, así como (ii) la sección Menú Authentic 1411 de la sección Comandos de menú.

© 2016-2022 Altova GmbH

11.1 Tutorial de la vista Authentic

En la vista Authentic puede editar documentos XML en una interfaz gráfica WYSIWYG (*ver imagen siguiente*) como si se tratase de una aplicación de procesador de textos como Microsoft Word. De hecho, lo único que tiene que hacer es introducir los datos. No tiene que preocuparse por el formato del documento, puesto que este ya está definido en la hoja de estilos que controla la vista Authentic del documento XML. La hoja de estilos StyleVision Power Stylesheet (en adelante SPS) la crea un diseñador de hojas de estilos con Altova StyleVision.



Para editar un documento XML en la vista Authentic el usuario puede hacer dos cosas: (i) editar la estructura del documento (por ejemplo, agregar/eliminar párrafos y títulos) e (ii) introducir datos (el contenido de los párrafos y títulos, por ejemplo).

Este tutorial describe:

- Cómo se abre un documento XML en la vista Authentic. El principal requisito para editar documentos XML en la vista Authentic es que los documentos estén asociados a un archivo SPS.
- La interfaz de la vista Authentic y cómo funcionan los mecanismos de edición en general.
- Cómo se edita la estructura de un documento insertando y eliminando nodos.
- Cómo se introducen datos en un documento XML.
- Cómo se introducen valores de atributo desde el ayudantes de entrada Atributos y valores de entidades.
- Cómo se imprime el documento.

Recuerde que este tutorial es solo una introducción y, por tanto, es bastante simple y directo. Si desea más información consulte el material de referencia de la sección Interfaz de la vista Authentic 613.

Material necesario para este tutorial

Todos los archivos necesarios para este tutorial se encuentran en la carpeta `...\Examples` de la carpeta de aplicación. Los archivos necesarios para este tutorial son:

- `NanonullOrg.xml` (el documento XML)
- `NanonullOrg.sps` (la hoja de estilos SPS a la que está vinculado el documento XML)
- `NanonullOrg.xsd` (el esquema XML en el que se basan y al que están vinculados el documento XML y la hoja de estilos SPS)
- `nanonull.gif` y `Altova_right_300.gif` (dos archivos de imagen usados en el tutorial)

Nota: en algunas secciones del tutorial le pediremos que mire el texto XML del documento XML (a diferencia de la vista Authentic del documento). Si la edición del producto de Altova que utiliza no incluye la vista Texto (si utiliza por ejemplo Authentic Desktop o Authentic Browser), entonces puede utilizar un editor de texto corriente como Wordpad o Notepad para ver el texto del documento XML.

Advertencia: le recomendamos que haga una copia del archivo NanonullOrg.xml y la utilice para este tutorial. Así podrá consultar siempre que quiera el documento original.

11.1.1 Abrir un documento en la vista Authentic

En la vista Authentic puede editar un documento XML ya existente o crear uno nuevo y editarlo. En esta sección del tutorial abrirá un documento XML en la vista Authentic y en las secciones siguientes aprenderá a editarlo. Esta sección también explica cómo se crea un documento XML nuevo para editarlo en la vista Authentic.

Abrir un documento XML ya existente

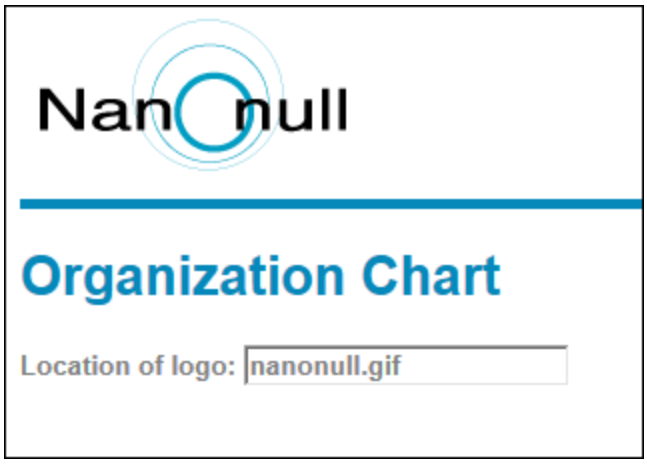
El archivo que abrirá es `NanonullOrg.xml`. Se encuentra en la carpeta `Examples` de su aplicación de Altova. Hay dos maneras de abrir el archivo:

- Haga clic en **Archivo | Abrir**, busque el archivo `NanonullOrg.xml` en el cuadro de diálogo que aparece y haga clic en **Abrir**.
- Utilice el explorador de Windows para localizar el archivo, haga clic con el botón derecho en el archivo y seleccione su producto de Altova en el submenú **Abrir con**.

El archivo `NanonullOrg.xml` se abre directamente en la vista Authentic (*imagen siguiente).* Hay dos razones por las que esto ocurre:

1. El archivo ya tiene asignada una hoja de estilos SPS (StyleVision Power Stylesheet)
2. Y en el cuadro de diálogo "Opciones" (**Herramientas | Opciones**), en la sección *Vista*, está marcada la opción para abrir archivos XML en la vista Authentic si tienen asignado un archivo SPS. (De lo contrario el archivo se abre en la vista Texto.)

© 2016-2022 Altova GmbH



Nota: el archivo que define y controla cómo se presenta el documento XML en la vista Authentic es el archivo SPS. Si no existe un SPS, el documento no se puede mostrar en la vista Authentic.

Crear un documento XML basado en una SPS

También puede crear un documento XML nuevo basado en una SPS. Hay dos maneras de hacerlo: con el comando de menú **Archivo | Nuevo** o con el comando de menú **Authentic | Documento nuevo**. En ambos casos es necesario seleccionar un archivo SPS.

Opción 1: Archivo | Nuevo

1. Seleccione **Archivo | Nuevo** y en el cuadro de diálogo "Crear documento nuevo" seleccione el tipo de archivo XML.
2. Haga clic en el botón **Seleccione una hoja de estilos de StyleVision** y busque el archivo SPS.

Si se ha asignado un archivo de plantilla XML al archivo SPS, los datos de ese archivo de plantilla XML se usan como base para crear la plantilla del documento XML en vista Authentic.

Opción 2: Authentic | Documento nuevo

1. Seleccione **Authentic | Documento nuevo**.
2. En el diálogo "Crear documento nuevo" busque el archivo SPS.

Si se asignó un archivo de plantilla XML al SPS, entonces los datos de la plantilla XML se usarán como datos de partida del documento XML que se crea en la vista Authentic.

11.1.2 La interfaz de la vista Authentic

La interfaz de edición de la vista Authentic está compuesta por una ventana principal, en la que se introducen y editan los datos del documento, y tres ayudantes de entrada. Editar documentos en la vista Authentic es fácil. Si desea ver el marcado del documento, active las etiquetas de marcado. Después puede empezar a escribir en el contenido del documento. Para cambiar la estructura del documento puede usar el menú contextual o el ayudante de entrada Elementos.

Mostrar las etiquetas de los nodos XML (marcado del documento)

Un documento XML es básicamente una jerarquía de nodos. Por ejemplo:

```
<DocumentRoot>
    <Person id="ABC001">
        <Name>Alpha Beta</Name>
        <Address>Some Address</Address>
        <Tel>1234567</Tel>
    </Person>
</DocumentRoot>
```

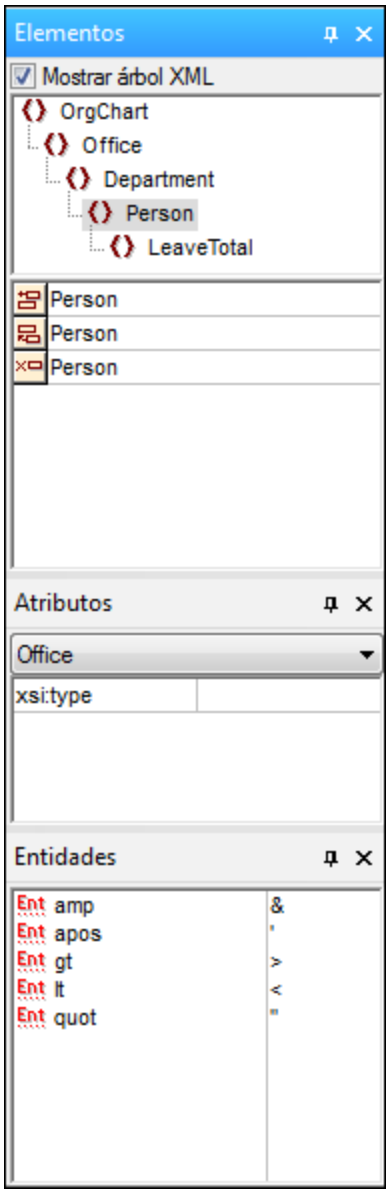
La opción predeterminada de la vista Authentic es no mostrar las etiquetas de los nodos. Para activar las etiquetas de los nodos, seleccione el comando de menú **Authentic | Mostrar marcado en tamaño grande** (o haga clic en el icono  de la barra de herramientas). Las etiquetas en tamaño grande muestran el nombre del nodo. También puede ver el marcado en tamaño pequeño (las etiquetas no incluyen el nombre del nodo) o como marcado mixto (definido por el diseñador de la hoja de estilos. En el marcado mixto predeterminado todas las etiquetas están desactivadas).

Puede ver el texto del documento XML en la vista Texto de su producto de Altova o en un editor de texto.

Ayudantes de entrada

Hay tres ayudantes de entrada en la interfaz de la vista Authentic (*imagen siguiente*), situados a la derecha de la ventana de la aplicación. Son los ayudantes de entrada Elementos, Atributos y Entidades.

© 2016-2022 Altova GmbH



Ayudante de entrada Elementos
Este ayudante de entrada muestra los elementos que se pueden insertar y quitar dependiendo de la posición del cursor o del texto que está seleccionado en la ventana principal. Tenga en cuenta que el ayudante de entrada es contextual y su contenido cambia dependiendo de la posición del cursor o del texto que está seleccionado. Además, cuando se selecciona un nodo del árbol XML del ayudante de entrada, los elementos relacionados con ese nodo aparecen en el ayudante de entrada. Para ver el árbol XML en el ayudante de entrada Elementos, marque la casilla "Mostrar árbol XML" situada en la parte superior del ayudante (*ver imagen anterior*). El árbol XML muestra la jerarquía de nodos a partir del elemento de nivel superior y hasta llegar al nodo que está seleccionado en la ventana principal.
Ayudante de entrada Atributos
Este ayudante de entrada muestra los atributos del elemento que está seleccionado en la ventana principal. Los valores de los atributos se pueden introducir y editar en este ayudante. El ayudante también incluye un cuadro combinado donde puede seleccionar todos los elementos de la jerarquía, desde el elemento de nivel superior hasta el elemento seleccionado. Al seleccionar un elemento en el cuadro combinado aparecen los atributos de ese elemento en el ayudante de entrada, donde pueden editarse.

Ayudante de entrada Entidades
Este ayudante de entrada no es contextual y muestra todas las entidades que se declararon para el documento. Al hacer doble clic en una entidad, esta se inserta en la posición del cursor. Para más información sobre cómo añadir entidades en un documento, consulte la sección Interfaz de la vista Authentic[613].

Menú contextual

Al hacer clic con el botón derecho en un documento abierto en la vista Authentic aparece un menú contextual con opciones relacionadas con la posición (el nodo) del documento donde se hizo clic. El menú contextual incluye comandos para:

- Insertar nodos en la posición del nodo, antes o después. Los submenús incluyen todos los nodos permitidos en la posición donde se hizo clic.
- Quitar el nodo seleccionado (si el esquema lo permite) o los elementos antecesores que se puedan quitar. Los nodos que se pueden quitar (de acuerdo con el esquema) aparecen en un submenú.
- Insertar entidades y secciones CDATA. Las entidades declaradas para el documento aparecen en un submenú. Las secciones CDATA sólo se pueden insertar dentro de texto.
- Cortar, copiar, pegar (como texto o como XML) y eliminar contenido del documento.

Nota: para más información sobre la interfaz, consulte la sección Interfaz de la Vista Authentic[613]

11.1.3 Operaciones con nodos

En la vista Authentic de un documento XML existen dos tipos principales de nodos: **nodos de elemento** y **nodos de atributo**. Estos nodos están marcados con etiquetas, que puede activar[601] si lo desea. El documento también puede incluir otros nodos, como nodos de texto (que no están marcados) y nodos de sección CDATA (que están marcados para separarlos del texto que los rodea).

Las operaciones de nodo descritas en este apartado se refieren únicamente a los nodos de elemento y de atributo. Recomendamos que tenga activado el marcado en tamaño grande[601] mientras prueba las operaciones descritas en este apartado.

Nota: es importante recordar que antes y después del elemento seleccionado sólo se pueden insertar **elementos del mismo nivel o de nivel superior**. Por ejemplo, los **elementos del mismo nivel** de un elemento párrafo serían otros elementos párrafos, pero también listas, una tabla, una imagen, etc. Los elementos del mismo nivel pueden aparecer antes o después de un elemento. Los elementos de nivel superior son los elementos **antecesores** y los elementos del mismo nivel de los antecesores. Por ejemplo, los elementos antecesores de un elemento párrafo serían un elemento sección, capítulo, artículo, etc. Un párrafo de un archivo XML válido ya tendría elementos antecesores. Por tanto, al añadir un elemento de nivel superior en la vista Authentic, el elemento nuevo que se añade se crea al mismo nivel que el antecesor correspondiente. Por ejemplo, si se inserta una sección después de un párrafo, se crea como elemento del mismo nivel que la sección en la que se encuentra dicho párrafo.

Realizar operaciones con nodos

Para realizar operaciones con nodos seleccione un comando del menú contextual[604] o haga clic en la opción correspondiente del ayudante de entrada Elementos[602]. En algunos casos se pueden añadir elementos o atributos haciendo clic en el hipervínculo Agregar nodo[605] de la vista Authentic del documento. Hay casos especiales en los que los elementos se definen como párrafos o elementos de lista. Si coloca el cursor dentro de dichos elementos y pulsa la tecla Entrar[606], se crea un elemento del mismo nivel de ese tipo. Este

© 2016-2022 Altova GmbH

apartado también describe cómo se crean y eliminan nodos con los mecanismos Aplicar elemento[606], Quitar nodo[606] y Borrar elemento[606].

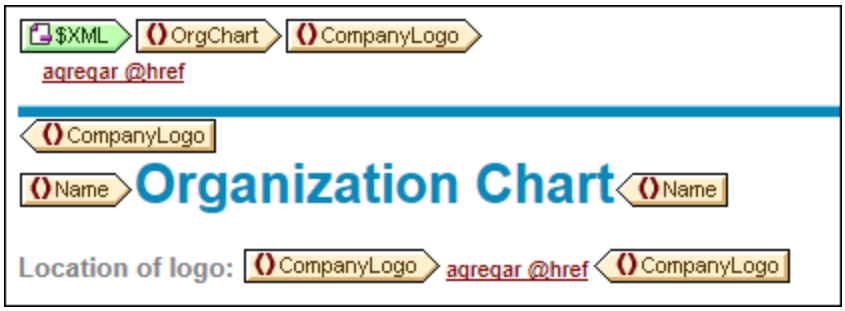
Insertar elementos

Puede insertar elementos en estas posiciones:

- Dentro del nodo de elemento, en la posición donde está situado el cursor. Los elementos que se pueden insertar en la posición seleccionada aparecen en un submenú del comando **Insertar** del menú contextual. En el ayudante de entrada Elementos los elementos que se pueden insertar en la posición seleccionada llevan el icono . En el documento `NanonullOrg.xml` coloque el cursor dentro del elemento `para` y cree los elementos `bold` e `italic` usando tanto el menú contextual como el ayudante de entrada Elementos.
- Antes o después del elemento seleccionado o de sus antecesores, si el esquema lo permite. Seleccione el elemento que desea insertar en los submenús del menú contextual. En el ayudante de entrada Elementos los elementos que se pueden insertar antes o después del elemento seleccionado llevan los iconos y respectivamente. Recuerde que desde el ayudante de entrada Elementos no se pueden insertar elementos antes ni después del elemento antecesor, sino antes o después del elemento seleccionado. Pruebe este comando primero dentro del elemento `para` y después dentro de la tabla que incluye los empleados de la compañía.

Hipervínculo Agregar nodo

Si un elemento o atributo está incluido en el diseño pero no está presente en el documento XML, entonces aparece el hipervínculo Agregar nodo en la posición del documento donde se especifica dicho nodo. Para ver este hipervínculo, vaya a la línea que tiene el texto *Location of logo*, seleccione el nodo `@href` del elemento `CompanyLogo` y elimínelo (pulsando la tecla **Suprimir**). Paso seguido aparece el hipervínculo `agregar @href` dentro del elemento `CompanyLogo` que acaba de cambiar (*ver imagen siguiente*). Al hacer clic en el hipervínculo se añade el nodo `@href` al documento XML. El cuadro de texto situado dentro de las etiquetas de `@href` aparece porque el diseño indica que el nodo `@href` se debe añadir de esta manera. Es decir, todavía tiene que introducir el valor (o el contenido) del nodo `@href`. Escriba `nanonull.gif` en el cuadro de texto.



Si el modelo de contenido de un elemento es ambiguo, por ejemplo, si especifica que una secuencia de elementos secundarios puede aparecer en cualquier orden, entonces aparece el hipervínculo `agregar...`. (sin el nombre del nodo). Al hacer clic en este hipervínculo aparece una lista de elementos que se pueden insertar.

Nota: el hipervínculo `agregar Nodo` aparece directamente en la plantilla del documento. No hay una opción equivalente ni en el menú contextual ni en el ayudante de entrada Elementos.

Crear elementos nuevos con la tecla Entrar

Hay casos en los que el diseñador de la hoja de estilos dio formato de párrafo o elemento de lista a un nodo. Si sitúa el cursor dentro de un nodo de este tipo y pulsa la tecla **Entrar**, entonces se inserta otro nodo del mismo tipo después del nodo seleccionado. Pruebe este mecanismo en el documento `NanonullOrg.xml`. Para ello sitúe el cursor al final de un nodo `para` (es decir, justo antes de su etiqueta final) y pulse la tecla **Entrar**.

Aplicar elementos

En los elementos de contenido mixto (es decir los que incluyen tanto texto como elementos secundarios) puede seleccionar parte del texto y aplicarle un elemento secundario. Como resultado el texto pasa a ser el contenido del elemento que se aplicó. Para aplicar elementos, seleccione **Aplicar** en el menú contextual y después seleccione uno de los elementos disponibles. (Si no se puede aplicar ningún elemento al texto seleccionado, entonces el comando **Aplicar** no aparece en el menú contextual.) En el ayudante de entrada Elementos los elementos que se pueden aplicar al texto seleccionado llevan el icono . En el documento `NanonullOrg.xml` seleccione parte del texto del contenido del elemento `para` y pruebe esta operación, aplicando los elementos `bold` e `italic`.

El diseñador de la hoja de estilos también puede crear un icono en la barra de herramientas para aplicar un elemento. Es el caso del documento `NanonullOrg.xml`, cuyos elementos `bold` e `italic` se pueden aplicar con solo hacer clic en los iconos **B** e ***I*** de la barra de herramientas de Authentic.

Quitar nodos

Puede quitar un nodo siempre y cuando al quitarlo el documento siga siendo válido. Cuando se quita un nodo se elimina también su contenido. Para quitar un nodo seleccione el comando **Quitar** del menú contextual. Cuando el comando **Quitar** está resaltado, aparece un submenú que contiene todos los nodos que se pueden quitar, empezando por el nodo seleccionado hasta llegar al nodo de nivel superior del documento. Para seleccionar el nodo que desea quitar, coloque el cursor dentro del nodo o resalte el nodo (o parte del nodo). En el ayudante de entrada Elementos los nodos que se pueden quitar llevan el icono . También puede hacer clic en un nodo y pulsar la tecla **Suprimir**. Pruebe esta operación en el documento `NanonullOrg.xml` y quite varios nodos usando los diferentes mecanismos descritos en este párrafo. Para deshacer los cambios pulse **Ctrl+Z**.

Borrar elementos

Puede borrar los elementos secundarios de elementos de contenido mixto (elementos que tienen texto y elementos secundarios). Para borrar el elemento entero, seleccione el nodo o coloque el cursor dentro del nodo. Para borrar un fragmento de texto, resalte el fragmento de texto. Cuando termine de seleccionar el nodo o fragmento de texto que desea borrar, seleccione la opción **Borrar** del menú contextual y después seleccione el elemento que desea borrar. En el ayudante de entrada Elementos los elementos que se pueden borrar en la posición seleccionada llevan el icono  (si la selección se hizo colocando el cursor en un punto de inserción) o el icono  (si seleccionó un intervalo). Pruebe este tipo de operación en el documento `NanonullOrg.xml`. Por ejemplo, intente borrar los elementos secundarios `bold` e `italic` del nodo `para` (que tiene contenido mixto).

Tablas y estructura de las tablas

En la vista Authentic hay dos tipos de tablas:

© 2016-2022 Altova GmbH

- *Tablas SPS (estáticas y dinámicas).* La estructura general de una tabla SPS la define el diseñador de la hoja de estilos. Dentro de la estructura general de la tabla, los únicos cambios que se pueden realizar son cambios de contenido. Por ejemplo, puede añadir filas nuevas en una tabla SPS dinámica.

- *Tablas XML.* Puede decidir si presenta el contenido de un nodo en forma de tabla. Si el diseñador de la hoja de estilos habilitó la posibilidad de crear este nodo en forma de tabla, entonces puede definir su estructura y editar su contenido. Las tablas XML se explican en detalle en el apartado Tablas en la vista Authentic 632.

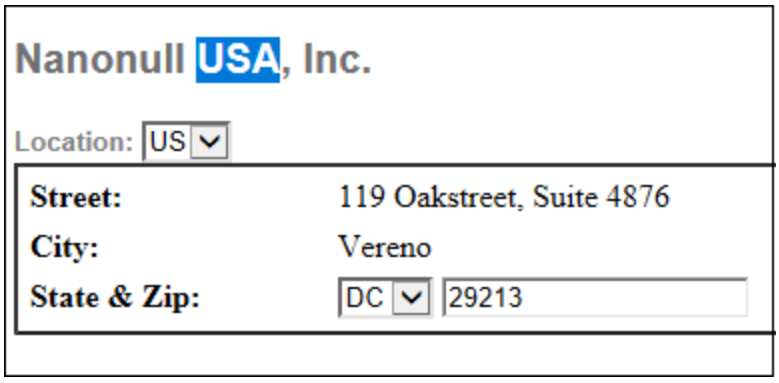
11.1.4 Introducir datos en la vista Authentic

En la vista Authentic los datos se introducen en el documento XML desde la ventana principal. Los valores de los atributos también se pueden introducir en el ayudante de entrada Atributos 609. Los datos se pueden introducir de dos maneras: (i) escribiendo el texto o (ii) mediante un dispositivo de entrada de datos, que se asigna a una entrada de texto predefinida.

Agregar contenido de texto

En la ventana principal de la vista Authentic puede introducir el contenido de los elementos y el valor de los atributos directamente en forma de texto. Para insertar contenido, sitúe el cursor en la posición deseada y teclee el texto. También puede copiar texto y pegarlo en el documento. También puede usar las teclas **Mayús** y **Suprimir**. Por ejemplo, puede resaltar el texto que desea editar y teclear el texto nuevo con la tecla **Mayús** activada.

Por ejemplo, para cambiar el nombre de la compañía, en el campo `Name` de `Office`, sitúe el cursor después de *Nanonull* y teclee *USA* para cambiar el nombre de la compañía a *Nanonull USA, Inc*.



Si el texto se puede editar, entonces podrá situar el cursor en el texto y resaltarlo. De lo contrario no podrá resaltar el texto con el cursor. Por ejemplo, intente cambiar el **nombre de los campos** (no su valor) *Street*, *City* o *State/Zip* del bloque `Address`. Verá que no puede colocar el cursor en este texto porque no es contenido XML, sino que viene dado por la hoja de estilos StyleVision Power Stylesheet.

Insertar caracteres especiales y entidades

En la vista Authentic los caracteres especiales y las entidades se gestionan de manera especial:

- *Caracteres especiales que se usan para marcado XML (&, ', >, <, ")*. Estos caracteres son entidades integradas[610] y se puede introducir en el documento con solo hacer doble clic en ellos en el ayudante de entrada Entidades. Si estos caracteres aparecen con frecuencia (en fragmentos de código, por ejemplo) puede incluirlos en secciones CDATA. Para introducir una sección CDATA, haga clic con el botón derecho en la posición elegida y seleccione **Insertar sección CDATA** en el menú contextual. El procesador XML ignora todos los caracteres de marcado que están dentro de secciones CDATA. Si desea introducir un carácter especial dentro de una sección CDATA, introduzca el carácter, no su referencia de entidad.
- *Caracteres especiales que no se pueden introducir con el teclado.* Para introducir estos caracteres cópielos y péguelos en la posición deseada.
- Puede definir como entidades[648] las *cadenas de texto que se usan frecuentemente*. Las entidades aparecen en el ayudante de entrada Entidades. Para insertar una entidad[610] sitúe el cursor en la posición elegida y haga doble clic en la entidad en el ayudante de entrada. Esta función es muy útil porque el valor de la cadena de texto se guarda en una sola ubicación y si necesita cambiar el valor de la cadena, basta con cambiar la definición de entidad.

Nota: en la vista Authentic es fácil pasar por alto los elementos vacíos si el marcado XML está oculto. Por eso recomendamos que active el marcado en tamaño grande o pequeño[601].

Pruebe a insertar caracteres especiales y entidades tal y como acabamos de explicar.

Agregar contenido mediante dispositivos de entrada de datos

Hasta ahora aprendió a introducir datos tecleando el texto directamente en el documento. En la vista Authentic también puede introducir **contenido en los elementos** (o valores de atributo) mediante dispositivos de entrada de datos.

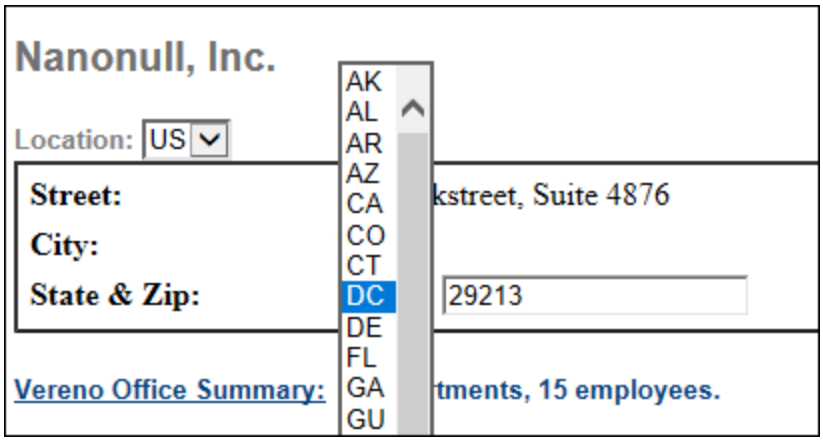
A continuación mostramos una lista de los dispositivos de entrada de datos de la vista Authentic, con una breve explicación de cada uno de ellos.

Dispositivo	Datos en el archivo XML
Campo de entrada (cuadro de texto)	Texto introducido por el usuario
Campo de entrada multilínea	Texto introducido por el usuario
Cuadro combinado	La opción elegida por el usuario tiene asignado un valor
Casilla de verificación	La opción elegida por el usuario tiene asignado un valor
Botón de opción	La opción elegida por el usuario tiene asignado un valor
Botón	La opción elegida por el usuario tiene asignado un valor

Por ejemplo, en la tabla estática que contiene los campos de dirección (*imagen siguiente*), hay dos dispositivos de entrada de datos: un campo de entrada para el campo *Zip* y un cuadro combinado para el campo *State*. Los valores que introduzca en el campo de entrada se introducen en el elemento

© 2016-2022 Altova GmbH

correspondiente del documento XML. En el cuadro combinado cada opción tiene asignado ya un valor y en el documento XML se introduce el valor de la opción seleccionada por el usuario.



El texto XML de la vista Authentic de la imagen anterior sería:

```
<Address>
   <ipo:street>119 Oakstreet, Suite 4876</ipo:street>
   <ipo:city>Vereno</ipo:city>
   <ipo:state>DC</ipo:state>
   <ipo:zip>29213</ipo:zip>
</Address>
```

Observe que la opción `DC` del cuadro combinado tiene asignado el valor `DC`. Por el contrario, el valor del campo `Zip` lo introduce el usuario libremente.

11.1.5 Introducir valores de atributo

Un atributo es una propiedad de un elemento y un elemento puede tener varios atributos. Los atributos tienen valores. A veces puede ser necesario introducir datos XML como valor del atributo. En la vista Authentic hay dos maneras de introducir valores de atributo:

- en la ventana principal, si así lo decidió el diseñador de la hoja de estilos y
- en el ayudante de entrada Atributos.

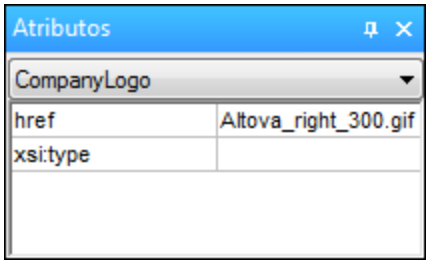
Valores de atributo en la ventana principal

Para introducir el valor de un atributo puede escribir texto, escribir texto en un campo de entrada o seleccionar una opción de entrada que tiene asignado un valor XML. Para más información consulte el apartado Introducir datos en la vista Authentic 607. En estos casos la distinción entre el contenido del elemento y el valor del atributo se vale del archivo StyleVision Power Stylesheet
In such cases, the distinction between element content and attribute value is made by the StyleVision Power Stylesheet and the data is handled appropriately.

Valores de atributo en el ayudante de entrada Atributos

También puede introducir un valor para un atributo o cambiar el valor de un atributo desde el ayudante de entrada Atributos. Primero seleccione el nodo de atributo en la vista Authentic. Después introduzca o edite el valor del atributo en el ayudante de entrada Atributos. En el documento `NanonullOrg.xml` la ubicación del logotipo viene dada por el valor del atributo `href` del elemento `CompanyLogo`. Para cambiar el logotipo de la compañía utilizado en el documento siga estos pasos:

1. Haga clic en la etiqueta `CompanyLogo` para seleccionar el elemento `CompanyLogo`. Los atributos del elemento `CompanyLogo` aparecen en el ayudante de entrada Atributos.
2. En el ayudante de entrada Atributos cambie el valor `nanonull.gif` del atributo `href` por `Altova_right_300.gif` (una imagen de la carpeta `Examples`).

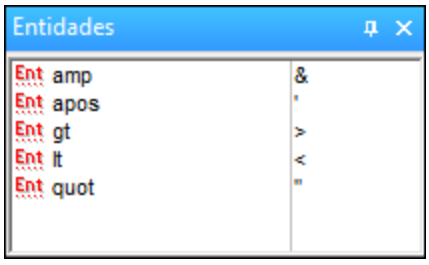


Como resultado el logotipo Nanonull desaparece y lo sustituye el logotipo de Altova.

Nota: en el ayudante de entrada Atributos no se pueden introducir entidades.

11.1.6 Añadir entidades

En la vista Authentic las entidades suelen ser datos XML (aunque no siempre), como un solo carácter, una cadena de texto o incluso un fragmento de un documento XML. Las entidades también pueden ser archivos binarios, como archivos de imagen por ejemplo. Todas las entidades que se pueden usar en el documento aparecen en el ayudante de entrada Entidades (*imagen siguiente*). Para insertar una entidad, sitúe el cursor en la posición deseada y haga doble clic en la entidad en el ayudante de entrada Entidades. Recuerde que en el ayudante de entrada Atributos no se pueden insertar entidades.

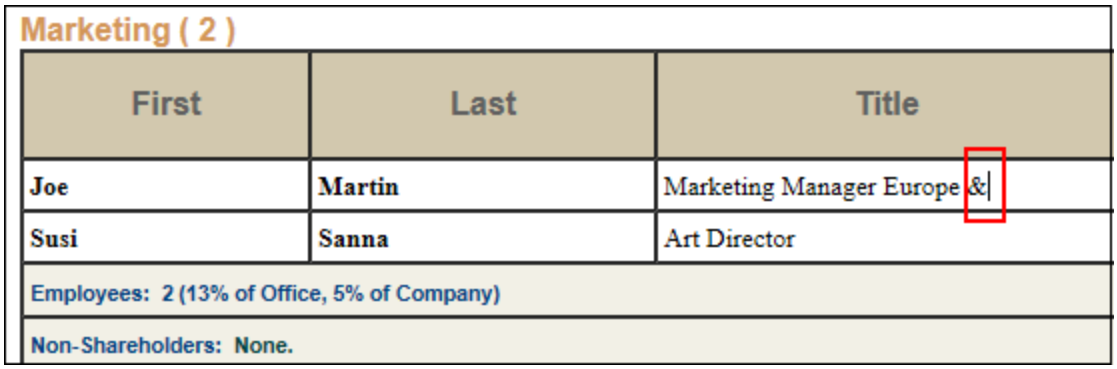


El carácter et (&) es de especial importancia en XML (al igual que los caracteres ', >, < y "). Para insertar estos caracteres, se utilizan entidades y así no se confunden con caracteres XML. Estos caracteres son entidades predeterminadas en la vista Authentic.

© 2016-2022 Altova GmbH

En el documento `NanonullOrg.xml` cambie el puesto de (*Title*) de Joe Martin (del departamento de Marketing) por *Marketing Manager Europe & Asia*. Para ello siga estos pasos:

1. Sitúe el cursor en la posición donde debe insertarse el carácter et (&).
2. En el ayudante de entrada Entidades, haga doble clic en la entidad `amp`. Como resultado se inserta el carácter et (&).



Nota: el ayudante de entrada Entidades no es contextual y muestra todas las entidades disponibles independientemente de la posición del cursor. Esto no significa que las entidades se puedan introducir en cualquier parte del documento. Por ello, después de insertar una entidad, es muy importante validar el documento: **XML | Validar el documento XML** (**F8**).

Definir entidades propias

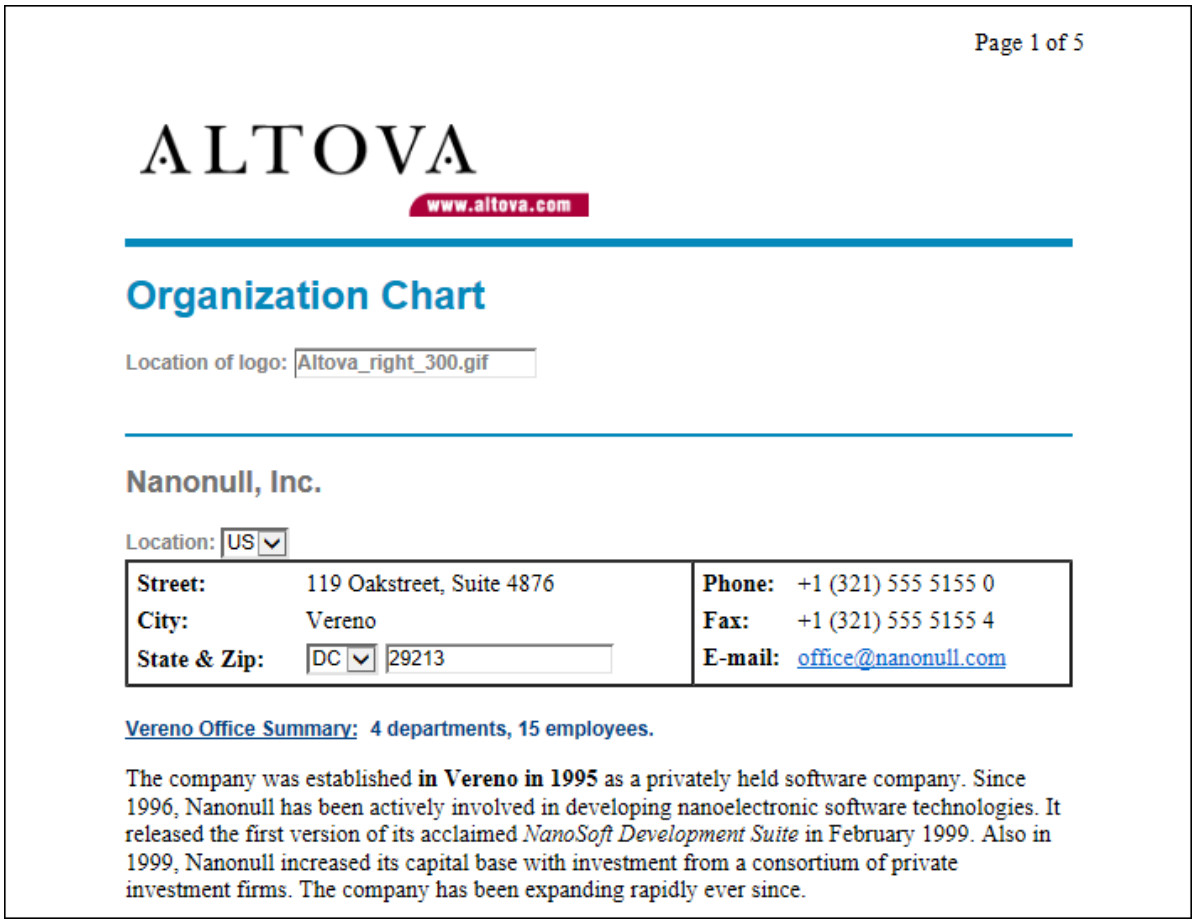
Si lo desea también puede editar sus propias entidades. Para aprender a hacerlo consulte el apartado Definir entidades en vista Authentic 648.

11.1.7 Imprimir el documento

Si lo desea puede imprimir la vista Authentic de su documento.

Para imprimir la vista Authentic de `NanonullOrg.xml` siga estos pasos:

1. Cambie al modo **Ocultar marcado** (es necesario desactivar el marcado si no quiere que se imprima).
2. Seleccione **Archivo | Vista previa de impresión** para obtener una vista previa de todas las páginas. A continuación aparece parte de la vista previa, reducida un 50%. Observe que el formato de la página es idéntico al que tiene en la vista Authentic.



3. Para imprimir el archivo, haga clic en **Archivo | Imprimir**.

Recuerde que también puede imprimir el documento con el marcado XML. Para ello, cambie la vista Authentic al modo **Mostrar marcado en tamaño grande** o **Mostrar marcado en tamaño pequeño** y después imprima el documento.

© 2016-2022 Altova GmbH

11.2 Interfaz de la vista Authentic

La vista Authentic se habilita haciendo clic en la pestaña Authentic del documento activo. Si no se asignó un archivo SPS al documento XML, se le pedirá que asigne uno. Para asignar un archivo SPS utilice el comando **Authentic | Asignar hoja de estilos de StyleVision**.

Esta sección incluye:

- Una presentación de la interfaz gráfica.
- Una descripción de los iconos de la barra de herramientas propios de la vista Authentic.
- Una descripción de los modos de visualización disponibles en la ventana principal de la vista Authentic.
- Una descripción de los ayudante de entrada e información sobre cómo usarlos.
- Una descripción de los menús contextuales disponibles en varios puntos de la vista Authentic del documento XML

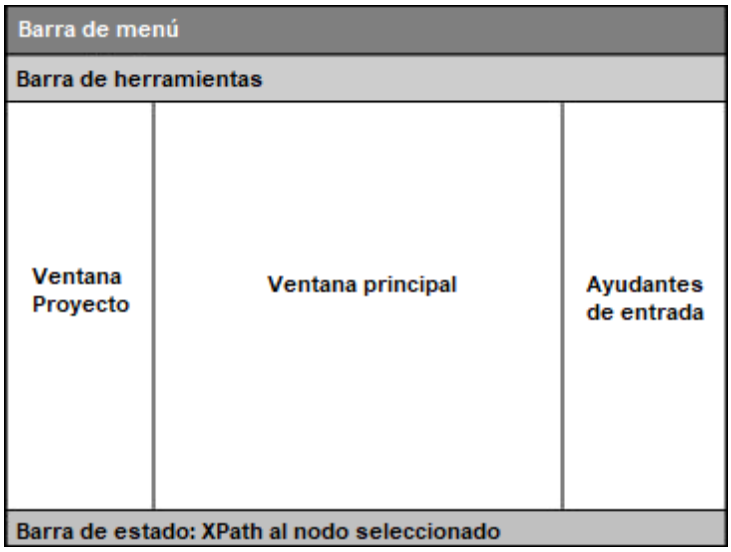
Para más información sobre la vista Authentic consulte estos recursos:

- El tutorial de la vista Authentic, que explica cómo usar la interfaz de la vista Authentic. Este tutorial está disponible en la documentación de Altova XMLSpy.
- La *Referencia del usuario* de su producto, que incluye una descripción detallada de los comandos de menú de la vista Authentic.

Sitio web de Altova: Edición de contenido XML, Creación de contenido XML

11.2.1 La interfaz gráfica del usuario (IGU)

La vista Authentic tiene una barra de menú y una barra de herramientas situadas en la parte superior de la ventana y tres secciones: la ventana Proyecto, la ventana principal y la ventana de los ayudantes de entrada. La imagen que aparece a continuación muestra todos estos componentes.



Barra de menú

Los menús de la barra de menú se describen en detalle en la *Referencia del usuario* del producto.

Barra de herramientas

Los símbolos e iconos de la barra de herramientas se describen en la sección iconos de la barra de herramientas de la vista Authentic 614.

Ventana Proyecto

Puede agrupar en un proyecto todos los archivos XML, XSL, XSD y de entidad. Para crear o modificar la lista de archivos del proyecto, use los comandos del menú Proyecto (descritos en la *Referencia del usuario* de su producto). La lista de archivos del proyecto aparece en la ventana Proyecto. Para tener acceso a los archivos de la ventana Proyecto, haga doble clic en el archivo que le interese.

Ventana principal

Esta es la ventana donde se visualiza y edita el documento XML . Para más información, consulte la sección ventana principal de la vista Authentic 617.

Ayudantes de entrada

La vista incluye tres ayudantes de entrada: Elementos, Atributos y Entidades. Las opciones de entrada de los ayudantes Elementos y Atributos son contextuales, es decir, dependen de la posición del cursor en el documento. Para introducir un elemento o entidad en el documento, haga doble clic en ellos en su ayudante de entrada. El valor de un atributo se introduce en el campo de valor del atributo en el ayudante de entrada Atributos. Para más información consulte la sección ayudantes de entrada de la vista Authentic 619.

Barra de estado

La barra de estado muestra la XPath al nodo que esté seleccionado en cada momento.

Menús contextuales

Los menús contextuales son los menús que aparecen al hacer clic con el botón derecho en la ventana principal. Los comandos son comandos de edición contextuales, es decir, permiten manipular la estructura y el contenido del nodo seleccionado. Por ejemplo, puede insertar, anexar o eliminar un nodo, añadir entidades o cortar y pegar contenido.

11.2.2 Iconos de la barra de herramientas de la vista Authentic

Los iconos de la barra de herramientas de la vista Authentic son básicamente accesos rápidos a los comandos. Algunos iconos le sonarán de otras aplicaciones de Windows o de otros productos de Altova, mientras que otros le serán totalmente nuevos. Esta sección describe los iconos propios y exclusivos de la vista Authentic.

Mostrar u ocultar marcado XML

En la vista Authentic puede elegir si se presentan las etiquetas de todos los elementos o atributos XML, de algunos o de ninguno, ya sea con el nombre de la etiqueta (marcado de tamaño grande) o sin el nombre de la

© 2016-2022 Altova GmbH

etiqueta (marcado de tamaño pequeño). Los cuatro iconos de marcado de la barra de herramientas se corresponden con cuatro comandos del menú **Authentic**.





Ocultar marcado. Se ocultan todas las etiquetas XML excepto las que se contrajeran. Si pulsa el icono *Ocultar marcado* y después hace doble clic en una etiqueta contraída (la manera normal de expandirla), entonces se mostrará el contenido del nodo pero las etiquetas se ocultarán.



Mostrar marcado en tamaño pequeño. Las etiquetas de los atributos o elementos XML se muestran sin el nombre de etiqueta.



Mostrar marcado en tamaño grande. Las etiquetas de los atributos o elementos XML se muestran con el nombre de etiqueta.



Mostrar marcado mixto. En la StyleVision Power Stylesheet se puede especificar cómo se debe mostrar cada elemento o atributo XML (con marcado grande o pequeño) o si no se debe mostrar. Esto recibe el nombre de marcado mixto porque algunos elementos se mostrarán con marcado y otros no. Por tanto, en el modo de marcado mixto el usuario de la vista Authentic ve un tipo de marcado personalizado. No obstante, es necesario decir que quien configura estas opciones es la persona que diseña la StyleVision Power Stylesheet. El usuario de la vista Authentic no puede configurar estas opciones.

Editar estructuras de tablas dinámicas

Las filas de una **tabla dinámica SPS** son básicamente repeticiones de una estructura de datos. Cada fila representa una repetición de un único elemento. Cada fila, por tanto, tiene la misma subestructura XML que la fila siguiente.

Los comandos de edición de la tabla dinámica permiten manipular las filas de una tabla dinámica SPS. Es decir, puede cambiar el número de veces que aparecen los elementos y el orden en el que aparecen. Sin embargo, no puede editar las columnas de una tabla dinámica SPS porque eso supondría cambiar la subestructura de las diferentes repeticiones del elemento.

Los iconos de edición de tablas dinámicas están en la barra de herramientas y se corresponden con los comandos del menú **Authentic**.





Anexar una fila a la tabla



Insertar una fila en la tabla



Duplicar la fila actual (es decir, el contenido de la celda se duplica)



Subir la fila actual



Bajar la fila actual



Eliminar la fila actual

Nota: estos comandos corresponden sólo a las **tablas dinámicas SPS**. No se deben usar dentro de tablas SPS estáticas. Los distintos tipos de tablas utilizados en la vista Authentic se describen en la sección Usar tablas en la vista Authentic[632] de la presente documentación.

Crear y editar tablas XML

Si desea presentar sus datos en forma de tabla, puede insertar sus propias tablas. Estas tablas se insertan como tablas XML. Puede modificar la estructura de la tabla XML y dar formato a la tabla. Los iconos para crear y editar tablas XML están disponibles en la barra de herramientas (*ver imagen siguiente*). Estos iconos se describen en detalle en la sección Iconos de edición de tablas XML[637].



Los comandos de estos iconos **no están disponibles como opciones de menú**. Recuerde también que para poder usar tablas XML, esta función debe estar habilitada y configurada correctamente en la StyleVision Power Stylesheet.

Para más información sobre los tipos de tablas utilizados en la vista Authentic y sobre cómo crear y editar tablas XML, consulte la sección Usar tablas en la vista Authentic[632].

Iconos de formato de texto

Para dar formato al texto en la vista Authentic es necesario aplicar el elemento o atributo XML que contiene el formato deseado. Si el formato deseado ya se definió, el diseñador de la StyleVision Power Stylesheet puede incluir iconos para dicho formato en la barra de herramientas de la vista Authentic. Para dar formato al texto, seleccione el texto y pulse el icono correspondiente.

Iconos de navegación por filas de base de datos



Estos iconos significan, de izquierda a derecha: *Ir al primer registro de la BD*, *Ir al registro anterior*, Abrir el cuadro de diálogo *Ir al registro número*, *Ir al registro siguiente* e *Ir al último registro*.



Este icono abre el cuadro de diálogo *Editar la consulta de base de datos*, donde puede escribir una consulta. La vista Authentic mostrará los registros de la consulta.

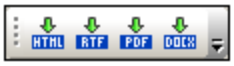
Edición de bases de datos XML

El comando **Seleccionar y editar una fila nueva con datos XML** permite seleccionar una fila nueva de la tabla pertinente en una base de datos XML, como IBM DB2. Esta fila aparece en la vista Authentic, donde se puede editar. Después puede guardar la fila otra vez en la base de datos.

Botones de la barra de herramientas para formularios XML portátiles (PXF)

La vista Authentic de XMLSpy incluye estos botones en la barra de herramientas para archivos PXF:

© 2016-2022 Altova GmbH



Al hacer clic en cada uno de estos botones, se genera un archivo de salida en formato HTML, RTF, PDF y DocX respectivamente.

Estos botones se habilitan cuando se abre un archivo PXF en la vista Authentic y de acuerdo con la configuración del archivo PXF. Por ejemplo, si el archivo PXF se configuró para contener una hoja de estilos XSLT para formatos de salida HTML y RTF, entonces se habilitarán los botones HTML y RTF, pero los botones PDF y DocX estarán deshabilitados.

11.2.3 Ventana principal de la vista Authentic

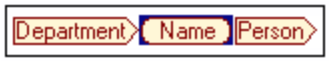
La vista Authentic ofrece cuatro modos de visualización: marcado grande, marcado pequeño, marcado mixto y marcado oculto. Estos modos permiten visualizar la información de marcado del documento de diferentes maneras. Para cambiar de un modo a otro, utilice los comandos del menú Authentic o los iconos de la barra de herramientas (ver el apartado anterior Iconos de la barra de herramientas de la vista Authentic 614).

Marcado de tamaño grande

Este modo muestra las etiquetas iniciales y finales de los elementos y atributos con los nombres del elemento o atributo:



El elemento `Name` de la imagen anterior está **expandido**, es decir, se muestra tanto las etiquetas iniciales y finales del elemento como su contenido. Los elementos y atributos se pueden **contraer**. Para ello haga doble clic en la etiqueta inicial o en la final. Para volver a expandir el elemento o atributo contraído, haga doble clic en la etiqueta contraída.

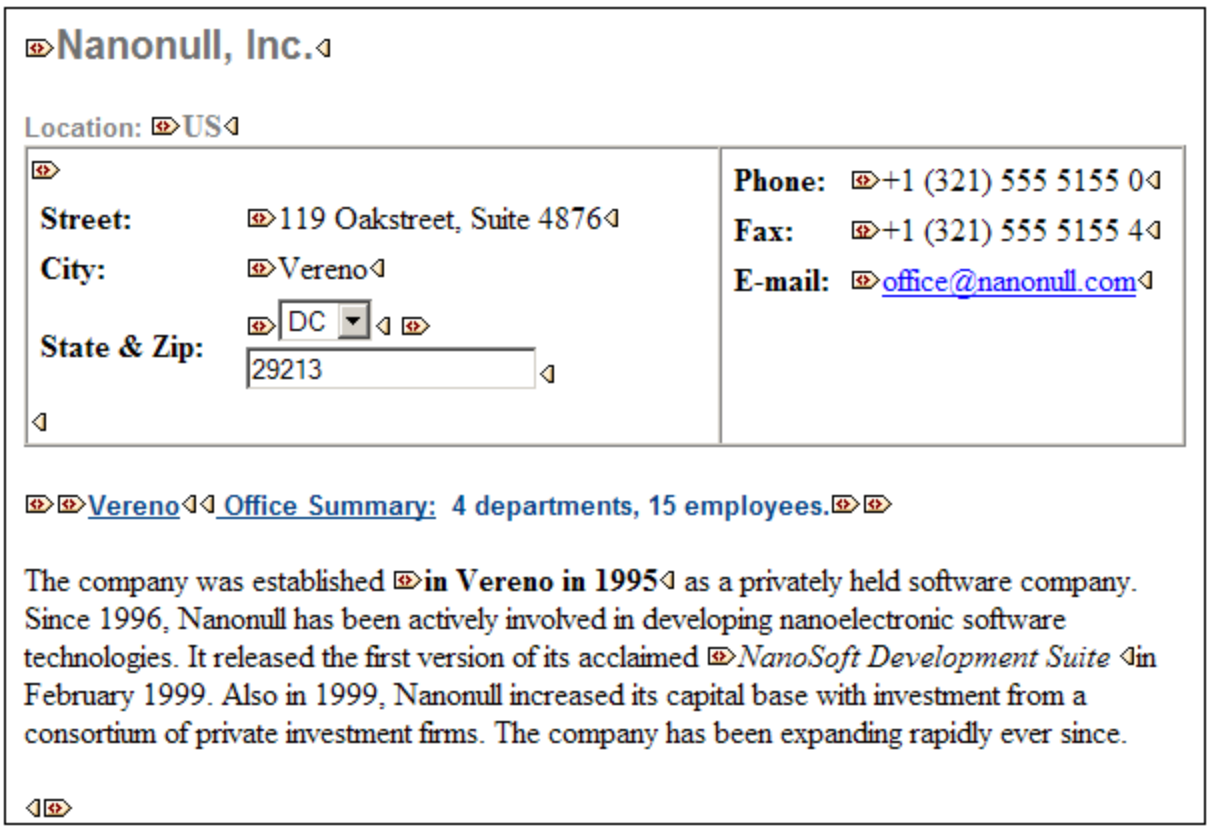


En el modo de marcado grande, los atributos se identifican porque las etiquetas iniciales y finales llevan el signo =:



Marcado de tamaño pequeño

Este modo muestra las etiquetas iniciales y finales de los elementos y atributos sin el nombre del elemento o atributo:



Observe que las etiquetas iniciales incluyen un símbolo, mientras que las etiquetas finales están vacías. Asimismo, las etiquetas de elementos llevan un símbolo en forma de corchetes angulares, mientras que las etiquetas de atributos llevan el signo = (*ver imagen siguiente*).



Para contraer o expandir un elemento o atributo, haga doble clic en su etiqueta. El ejemplo que aparece a continuación muestra un elemento contraído (resaltado en color azul). Observe la forma de la etiqueta del elemento contraído, comparada con la etiqueta inicial del elemento expandido situada a la izquierda.



Marcado mixto

El modo de marcado mixto muestra el marcado según las opciones de configuración elegidas por la persona que diseñó la StyleVision Power Stylesheet (por cada elemento y atributo del documento puede indicar si el elemento o atributo debe mostrarse con marcado grande, marcado pequeño o sin marcado). El usuario de la vista Authentic ve este marcado personalizado en el modo de marcado mixto.

Ocultar marcado

Se oculta el marcado XML. Puesto que el formato visible en la vista Authentic es el formato del documento impreso, este modo de visualización es una vista WYSIWYG del documento.

© 2016-2022 Altova GmbH

Visualización de contenidos

En la vista Authentic el contenido se puede visualizar de dos maneras:

- Texto sin formato. El usuario escribe el texto y este texto se convierte en el contenido del elemento o en el valor del atributo.



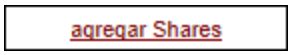
- Mecanismos de entrada de datos. La presentación incluye un campo de entrada de datos (un cuadro de texto), un campo de entrada multilínea, un cuadro combinado, una casilla de verificación o un botón de opción. En el caso de campos de entrada y campos de entrada multilínea, el texto que se escriba en el campo se convierte en el contenido XML del elemento o en el valor del atributo.



  En el caso de los demás mecanismos de entrada de datos, la opción seleccionada produce un valor XML equivalente, que viene especificado en la StyleVision Power Stylesheet. Por tanto, si en un cuadro combinado selecciona, por ejemplo, la opción "confirmado" (que estaría disponible en la lista desplegable del cuadro combinado) al elemento se le asignaría el valor XML "1" o el valor "confirmado", por ejemplo. Si la opción seleccionada fuese "no confirmado", se le asignaría el valor XML "0" o el valor "no confirmado", por ejemplo.

Nodos opcionales

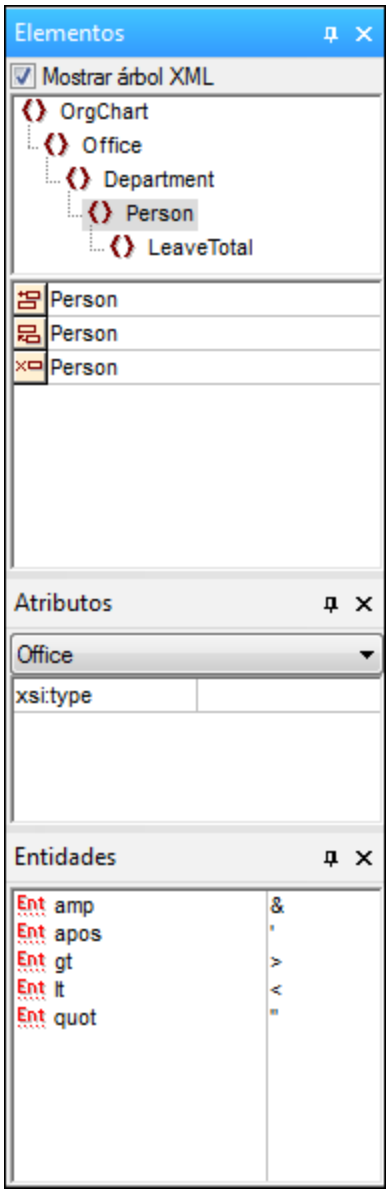
Cuando un elemento o atributo es opcional (de acuerdo con el esquema al que se hace referencia), aparece un aviso de tipo `agregar [nombre del elemento/atributo]`:



Al hacer clic en el aviso se agrega el elemento y el cursor se coloca en el campo de entrada de datos. Si hay varios nodos opcionales, aparece el aviso `agregar...` . Al hacer clic en el aviso, aparece un menú con los nodos opcionales.

11.2.4 Ayudantes de entrada de la vista Authentic

La vista Authentic ofrece tres ayudantes de entrada: Elementos, Atributos y Entidades. Los ayudantes de entrada están disponibles en tres ventanas situadas a la derecha de la interfaz de la vista Authentic (*imagen siguiente*).



Los ayudantes de entrada Elementos y Atributos son contextuales, es decir, sus opciones dependen de la posición del cursor en el documento. Las entidades del ayudante de entrada Entidades no son contextuales, es decir, el ayudante muestra todas las entidades permitidas en el documento, independientemente de la posición del cursor.

A continuación describimos los ayudantes de entrada uno por uno.

Ayudante de entrada Elementos

El ayudante de entrada Elementos está dividido en dos paneles:

- Un panel superior con el árbol XML (que se puede activar o desactivar haciendo clic en la casilla **Mostrar árbol XML**). El árbol XML muestra los antecesores del elemento seleccionado hasta la altura del elemento raíz del documento. Al hacer clic en un elemento del árbol XML, los elementos relevantes

 © 2016-2022 Altova GmbH

para ese elemento (que se describen en el siguiente punto) aparecen en el panel inferior del ayudante de entrada Elementos.
- Un panel inferior con los nodos que se pueden eliminar, insertar en la posición actual, insertar antes de la posición actual e insertar después de la posición actual. La lista del panel inferior también incluye los nodos que se pueden aplicar al elemento o intervalo de texto seleccionado y los que se pueden borrar del elemento o intervalo de texto en la vista Authentic. Los nodos de esta lista van acompañados por un icono que indica el tipo de acción que se puede realizar. A continuación describimos cada uno de estos iconos y su significado.

Para utilizar un nodo del ayudante de entrada Elementos, haga clic en su icono.



Insertar después del elemento
El elemento del ayudante de entrada se inserta después del elemento seleccionado y se anexa en el nivel jerárquico correcto. Por ejemplo, si el cursor está dentro de un elemento `//sect1/para` y anexa un elemento `sect1`, el elemento `sect1` nuevo no se anexará como elemento del mismo nivel que `//sect1/para` sino como elemento del mismo nivel que `sect1`, que es el elemento primario del elemento `para`.



Insertar antes del elemento
El elemento del ayudante de entrada se inserta antes del elemento seleccionado. Tenga en cuenta que, al igual que el comando **Insertar después del elemento**, el elemento se inserta en el nivel jerárquico correcto.



Quitar elemento
Quita el elemento y su contenido.



Insertar elemento
También puede insertar un elemento del ayudante de entrada dentro de otro elemento. Cuando el cursor está dentro de un elemento, entonces se pueden insertar los elementos secundarios permitidos de ese elemento. Recuerde que los elementos secundarios permitidos pueden formar parte de un modelo de contenido que sólo contiene elementos o de un modelo de contenido mixto (con elementos y texto).

Para insertar un elemento secundario permitido, seleccione un intervalo de texto o sitúe el cursor en una posición dentro del texto (el punto de inserción).

- Cuando se selecciona un intervalo de texto y se inserta un elemento, el intervalo de texto se convierte en el contenido del elemento insertado.
- Cuando se inserta un elemento en un punto de inserción, se inserta el elemento en ese punto.

Después de insertar el elemento, puede borrarlo con solo hacer clic en el icono **Borrar elemento** del ayudante de entrada Elementos, disponible para estos elementos insertados. El icono **Borrar elemento** puede ser de dos tipos, dependiendo de si selecciona un intervalo de texto o sitúa el cursor en el texto para indicar un punto de inserción (véase la descripción que aparece más abajo).



Aplicar elemento
Si selecciona un elemento del documento (active el modo de vista "Mostrar marcado en tamaño grande" y seleccione la etiqueta inicial o final del elemento) y ese elemento se puede sustituir por otro elemento (p. ej. en un elemento de contenido mixto como `para`, el elemento

`italic` se puede sustituir por el elemento `bold`), este icono indica que el elemento del ayudante de entrada se puede aplicar al elemento seleccionado (el elemento original). El comando **Aplicar elemento** también se puede aplicar a un intervalo de texto de un elemento de contenido mixto. El intervalo de texto se convertirá en el contenido del elemento que se aplicó.

- Si el elemento que se aplicó tiene **un secundario con el mismo nombre** que el secundario del elemento original y existe una instancia de este secundario en el elemento original, entonces se mantiene el secundario del elemento original en el contenido del elemento nuevo.
- Si el elemento que se aplicó **no tiene un secundario con el mismo nombre** que el secundario del elemento original y existe una instancia del secundario en el elemento original, entonces el secundario del elemento original se anexa como elemento del mismo nivel de los secundarios del elemento nuevo.
- Si el elemento que se aplicó tiene **un secundario para el que no hay un equivalente** en el modelo de contenido del elemento original, entonces este elemento secundario no se crea automáticamente, aunque la vista Authentic le da la opción de insertarlo.

Si en lugar de seleccionar un elemento, selecciona un intervalo de texto y pulsa el icono **Aplicar elemento**, entonces se crea el elemento del ayudante de entrada en esa posición y el texto seleccionado se convierte en el contenido del elemento recién creado. No se puede aplicar un elemento cuando el cursor está situado en un punto de inserción.



Borrar elemento (cuando se selecciona un texto)
Este icono aparece cuando se selecciona texto dentro de un elemento de contenido mixto. Si hace clic en el icono, se borra el elemento que rodea el texto seleccionado.



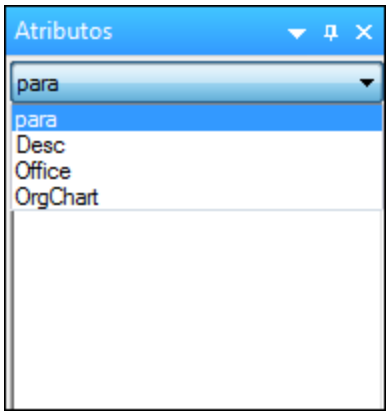
Borrar elemento (cuando se selecciona un punto de inserción)
Este icono aparece cuando se sitúa el cursor dentro de un elemento que es un secundario de un elemento de contenido mixto. Si hace clic en el icono, se borra el elemento alineado.

Ayudante de entrada Atributos

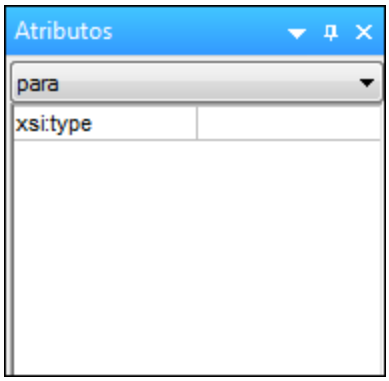
El ayudante de entrada Atributos contiene un cuadro combinado con una lista desplegable. Dentro del cuadro combinado aparece el elemento seleccionado (seleccione la etiqueta inicial o final de un elemento o sitúe el cursor dentro del contenido del elemento).

El ayudante de entrada que aparece en estas imágenes incluye un elemento `para` en el cuadro combinado. Al hacer clic en la punta de flecha del cuadro combinado, se despliega una lista con todos los ancestros del elemento `para` **hasta llegar el elemento raíz del documento**, en este caso `OrgChart`.

© 2016-2022 Altova GmbH



Justo debajo del cuadro combinado aparece una lista de atributos válidos para el elemento seleccionado en el cuadro combinado, en este caso el elemento `para`. Si un atributo es obligatorio en un elemento dado, el atributo aparece en negrita. (En la imagen siguiente, no hay atributos obligatorios, excepto el atributo integrado `xsi:type`.)



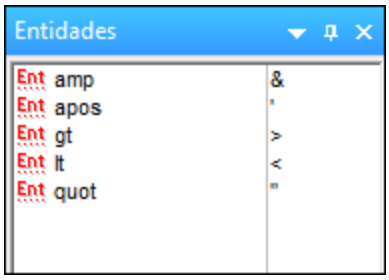
Para introducir el valor de un atributo, haga clic en el campo de valor del atributo y escriba el valor. Como resultado se crea el atributo y su valor en el documento XML.

En el caso del atributo `xsi:nil`, que aparece en el ayudante de entrada Atributos cuando se selecciona un elemento que puede tener valores nulos, entonces el valor del atributo `xsi:nil` sólo se puede introducir seleccionando los valores permitidos (`true` o `false`) de la lista desplegable del campo de valor del atributo.

Para cambiar el atributo `xsi:type` haga clic en el campo de valor del atributo y después (i) seleccione un valor de la lista desplegable que aparece o (ii) introduzca un valor. Los valores de la lista son los tipos abstractos disponibles definidos en el esquema XML en el que se basa el documento de la vista Authentic.

Ayudante de entrada Entidades

El ayudante de entrada Entidades permite insertar entidades en el documento. Las entidades se pueden usar para insertar caracteres especiales o fragmentos de texto que aparecen con frecuencia en el documento (como, por ejemplo, el nombre de una compañía). Para insertar una entidad, sitúe el cursor en la posición donde desea insertar la entidad y haga doble clic en la entidad en el ayudante de entrada Entidades.



Nota: se denomina *entidad interna* aquella entidad cuyo valor está definido dentro de una DTD. Por el contrario, *entidad externa* es aquella cuyo valor está en una fuente externa, p. ej. otro archivo XML. Tanto las entidades externas como las internas aparecen en el ayudante de entrada Entidades. Cuando se inserta una entidad, ya sea externa o interna, se inserta la entidad en el texto XML, no sólo su valor. Si la entidad es interna, la vista Authentic **muestra el valor de la entidad**. Si la entidad es externa, la vista Authentic muestra la entidad, pero no su valor. Esto significa que, por ejemplo, en la vista Authentic un archivo XML que sea una entidad externa se mostrará como entidad. Su contenido no sustituirá a la entidad en la vista Authentic.

En la vista Authentic también puede definir entidades nuevas, que aparecerán también en el ayudante de entrada. Para más información consulte el apartado Definir entidades 648 de la sección *Edición en la vista Authentic*.

11.2.5 Menús contextuales de la vista Authentic

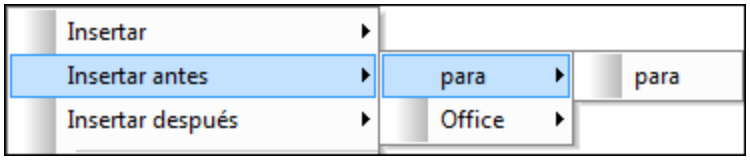
Al hacer clic con el botón derecho en un nodo del documento o en el contenido, aparece un menú contextual con comandos relacionados con la selección o la posición del cursor.

Insertar elementos

La imagen siguiente muestra el submenú Insertar, que incluye una lista con todos los elementos que se pueden insertar en la posición del cursor. El submenú Insertar antes ofrece una lista con todos los elementos que se pueden insertar antes del elemento seleccionado. El submenú Insertar después ofrece una lista con todos los elementos que se pueden insertar después del elemento seleccionado. En la imagen siguiente, el elemento seleccionado es para y dentro del elemento para se pueden insertar los elementos `ts:bold`, `ts:italic` y `ts:underline`.



Como muestra la imagen siguiente, antes del elemento `para` se pueden insertar los elementos `para` y `Office`.



 © 2016-2022 Altova GmbH

Los comandos de inserción de nodo, sustitución (**Aplicar**) y eliminación de marcado (**Quitar**) del menú contextual también están disponibles en los ayudantes de entrada de la vista Authentic[619] y se describen en detalle en esa sección.

Insertar entidad

Al pasar el puntero sobre el comando **Insertar entidad** aparece un submenú con una lista de todas las entidades declaradas. Al hacer clic en una entidad de la lista, la entidad se inserta en la posición seleccionada. Para más información sobre cómo definir entidades para sus documentos, consulte la sección Definir entidades[648].

Insertar sección CDATA

Este comando se habilita cuando se sitúa el cursor dentro de un texto. Al hacer clic en este comando, se inserta una sección CDATA en el punto de inserción del cursor. La sección CDATA está delimitada por una etiqueta inicial y otra final. Para ver las etiquetas, active el marcado en tamaño grande o tamaño pequeño. Dentro de las secciones CDATA se ignora el marcado XML y no se analizan caracteres. Los caracteres de marcado XML (como & ' > < ") no se analizan como marcado, sino como literales. Por tanto, las secciones CDATA son ideales a la hora de usar código de programa de muestra con caracteres XML en el documento.

Quitar nodo

Al pasar el puntero sobre el comando **Quitar** aparece un submenú con una lista que incluye el nodo seleccionado y todos sus ancestros que se puedan quitar (si al quitar estos ancestros el documento sigue siendo válido). Haga clic en el elemento que desea quitar. Tenga en cuenta que si hace clic en un ancestro, se eliminará el elemento y sus descendientes, incluido el elemento seleccionado.

Borrar

El comando **Borrar** elimina el marcado del elemento que envuelve la selección. Si se selecciona todo el nodo, entonces se elimina el marcado de todo el nodo. Si se selecciona un fragmento de texto, se elimina el marcado que envuelve sólo el fragmento seleccionado.

Aplicar

El comando **Aplicar** aplica un elemento seleccionado al nodo o texto seleccionado en la ventana principal. Para más detalles, consulte el apartado ayudantes de entrada de la vista Authentic[619].

Copiar, cortar y pegar

Son los comandos estándar de Windows. No obstante, tenga en cuenta que el comando **Pegar** pega el texto copiado bien como XML, bien como texto, dependiendo de lo especificado en la hoja de estilos SPS. Para más información sobre los comandos **Copiar como XML** y **Copiar como texto**, consulte la descripción del párrafo siguiente sobre el comando **Copiar como**.

Pegar como

El comando Pegar como ofrece la opción de pegar un fragmento XML de la vista Authentic (que se copió en el portapapeles) como XML o como texto. Si el fragmento copiado se pega como XML, se pega e incluye el marcado XML. Si se pega como texto, entonces se pega sólo el contenido de texto del fragmento copiado (sin el marcado XML). Se pueden dar estos dos casos:

- Se selecciona ***todo un nodo y sus etiquetas de marcado*** en la vista Authentic y se copia en el portapapeles. (i) El nodo se puede pegar como XML en cualquier posición, siempre que la posición sea válida. No se puede pegar en una posición no válida. (ii) Si el nodo se pega como texto, se pega sólo el *contenido de texto* del nodo (no su marcado). El texto se puede pegar en cualquier posición del documento XML que acepte texto.
- Se selecciona un ***fragmento de texto*** en la vista Authentic y se copia en el portapapeles. (i) Si este fragmento se pega como XML, las etiquetas de marcado XML del texto se pegan con el texto (aunque no se copiaran), pero sólo si el nodo XML es válido en la posición donde se pega el fragmento. (ii) Si el fragmento se pega como texto, entonces se puede pegar en cualquier posición del documento XML que acepte texto.

Nota: El texto se copiará en nodos que acepten texto y el usuario debería comprobar si, tras copiar el texto, el documento sigue siendo válido. Recuerde que el texto copiado:

(i) debería ser válido desde un punto de vista léxico en la posición nueva (por ejemplo, no sería válido pegar caracteres numéricos en un nodo de tipo numérico) y
(ii) no debería invalidar el nodo en la posición nueva (por ejemplo, no sería válido pegar cuatro dígitos en un nodo que acepta números con un máximo de tres dígitos).

Si el texto pegado invalida el documento, el texto aparecerá en color rojo a modo de aviso.

Eliminar

El comando **Eliminar** elimina el nodo seleccionado y su contenido. Para seleccionar el nodo que desea eliminar, sitúe el cursor dentro del nodo o haga clic en su etiqueta inicial o final.

© 2016-2022 Altova GmbH

11.3 Edición en la vista Authentic

En esta sección describimos con detalle algunas funciones de la vista Authentic que consideramos importantes. Estas funciones son importantes bien porque se utilizan con mucha frecuencia o bien porque implican una cierta complejidad y deben ser explicadas en más detalle.

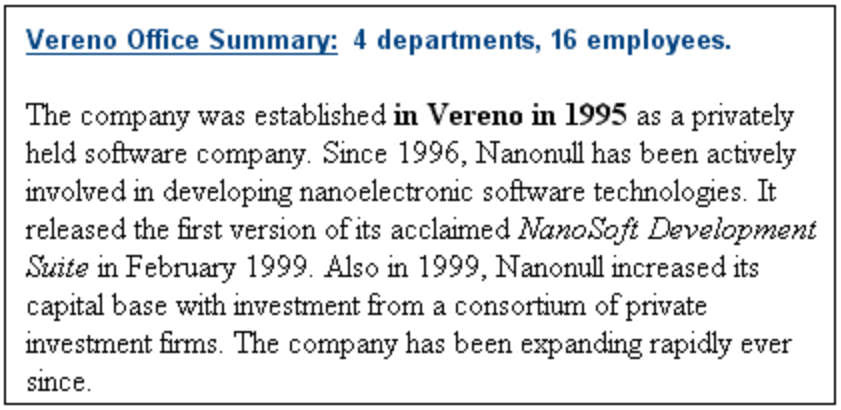
Los temas que describimos en esta sección son:

- En la vista Authentic se pueden usar tres tipos de tablas diferentes. El apartado Tablas en la vista Authentic 632 describe los tres tipos de tablas (tablas SPS estáticas, tablas SPS dinámicas y tablas XML) así como cuándo y cómo se utilizan. Este apartado empieza con una descripción general y después explica detalladamente el uso de las tablas.
- El selector de fechas es un calendario gráfico con el que puede introducir fechas en el formato XML correcto con solo hacer clic en una fecha. Consulte el apartado titulado Selector de fecha 646.
- Una entidad es una abreviación de un carácter especial o una cadena de texto. En la vista Authentic puede definir entidades propias para poder insertar estos caracteres o cadenas de texto especiales. Para más información consulte el apartado Definir entidades 648.
- En las ediciones Enterprise de los productos de Altova, el usuario de la vista Authentic puede firmar los documentos XML con firmas XML digitales 650 y verificar las firmas.
- Qué formatos de imagen 651 pueden visualizarse en la vista Authentic.

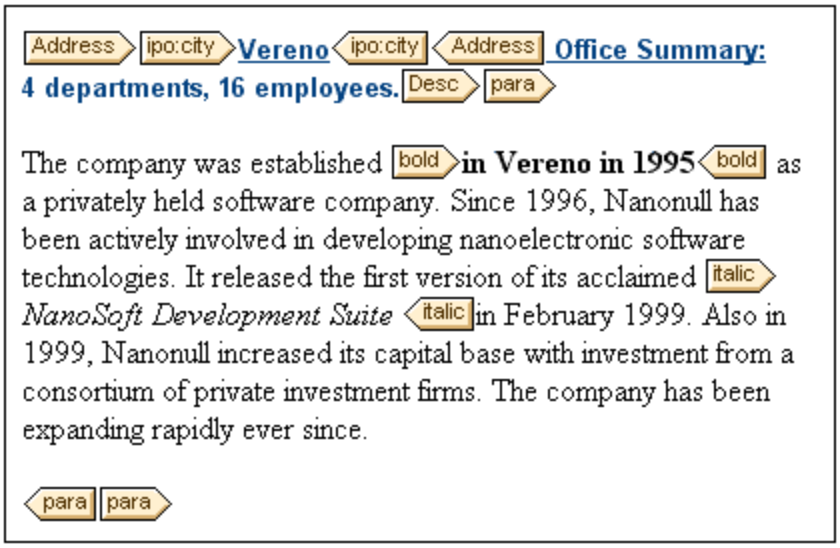
Sitio web de Altova: Edición de contenido XML, Creación de contenido XML

11.3.1 Edición básica

En la vista Authentic puede editar documentos XML. Sin embargo, la vista Authentic puede ocultar el marcado XML del documento y mostrar solamente el contenido del documento (*ver la primera imagen de este apartado*). De este modo, el usuario no entra en contacto con la tecnología XML y puede editar el documento como si se tratase de un documento de texto normal y corriente. No obstante, si lo prefiere, puede activar el marcado XML del documento en cualquier momento (*ver la segunda imagen de este apartado*).



Un documento editable en la vista Authentic sin marcado XML.



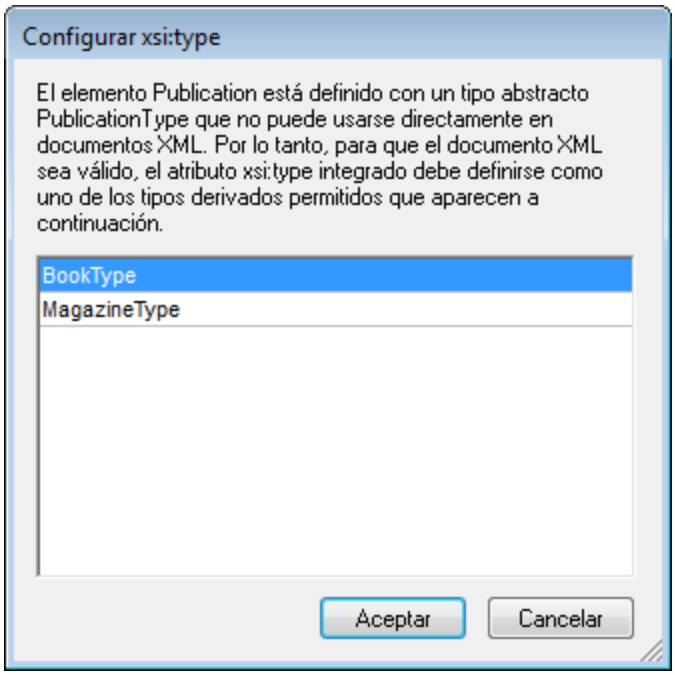
Un documento editable en la vista Authentic con las etiquetas de marcado XML.

Insertar nodos

A menudo es necesario agregar nodos nuevos al documento XML de Authentic. Por ejemplo, puede que necesite agregar un elemento Person nuevo en una libreta de direcciones. En estos casos el esquema XML permitiría añadir el elemento nuevo. Usted sólo tiene que hacer clic con el botón derecho en el nodo del documento antes o después del cual desea agregar el nodo nuevo. En el menú contextual que aparece seleccione **Insertar antes** o **Insertar después** según corresponda. Los nodos que se pueden insertar en la posición elegida aparecen en un submenú. Haga clic en el nodo que desea insertar y este se insertará en la posición elegida. Todos los nodos descendientes obligatorios se insertarán también. Si un nodo descendiente es opcional, entonces aparece el hipervínculo `Agregar nombreDelNodo` para que pueda insertar el nodo si lo desea.

Si el nodo que se añade es un elemento con un tipo abstracto, entonces aparece un cuadro de diálogo (*similar al de la imagen siguiente*) con una lista de los tipos derivados que están disponibles en el esquema XML.

© 2016-2022 Altova GmbH



El diálogo anterior aparece cuando se añade un elemento `Publication`. El elemento `Publication` es de tipo `PublicationType`, que es un tipo complejo abstracto. Los dos tipos complejos `BookType` y `MagazineType` se derivan del tipo complejo abstracto `PublicationType`. Por tanto, cuando se añade un elemento `Publication` al documento XML, es necesario especificar uno de estos dos tipos concretos derivados del tipo abstracto de `Publication`. El nuevo elemento `Publication` se añadirá con un atributo `xsi:type`:

```
<Publication xsi:type="BookType"> ... </Publication>
<Publication xsi:type="MagazineType"> ... </Publication>
<Publication xsi:type="MagazineType"> ... </Publication>
```

Al seleccionar uno de los tipos derivados disponibles y hacer clic en **Aceptar** ocurre esto:

- el tipo derivado seleccionado se establece como valor del atributo `xsi:type` del elemento nuevo y
- se inserta el elemento junto con sus nodos descendientes definidos en el modelo de contenido del tipo derivado seleccionado.

El tipo derivado seleccionado se puede cambiar más tarde. Para ello cambie el valor del atributo `xsi:type` del elemento en el ayudante de entrada Atributos. Cuando cambie el tipo del elemento de esta forma, se quitarán del documento todos los nodos del modelo de contenido del tipo anterior y se insertarán los nodos del modelo de contenido del tipo nuevo.

Edición de texto

Los documentos de la vista Authentic están compuestos por texto e imágenes. Para editar el texto del documento, sitúe el cursor en la posición donde desea insertar texto y empiece a teclear. Puede copiar, mover y eliminar texto de forma normal (por ejemplo con la tecla **Suprimir**) y mediante operaciones de arrastrar y colocar. La única excepción es la tecla **Entrar**. Puesto que el documento de la vista Authentic viene con formato previo, el usuario no puede agregar líneas ni espacios entre los componentes. Por tanto, en la vista Authentic la tecla **Entrar** sirve para anexar otra instancia del elemento que se está editando y sólo se puede usar para esto.

Copiar como XML o como texto

Puede copiar y pegar texto como XML o como texto.

- Si el texto se pega como XML, entonces se pega el marcado XML y el contenido de texto de los nodos. El marcado XML se pega incluso si solo copió parte del contenido del nodo. Para poder pegar el marcado es necesario que el esquema permite insertar el nodo en la posición elegida.
- Si el texto se pega como texto, entonces no se pega el marcado XML.

Para pegar el texto como XML o como texto, antes debe copiar el texto (**Ctrl+C**), hacer clic con el botón derecho en la posición donde desea pegarlo y seleccionar **Pegar como | XML** o **Pegar como | Texto** en el menú contextual. Si utiliza **Ctrl+V** para pegar el texto, este se pega en el modo predeterminado de la hoja de estilos SPS. El modo predeterminado lo especifica el diseñador de la hoja de estilos SPS. Para más información, consulte el apartado Menús contextuales 624.

También puede resaltar un texto y arrastrarlo hasta la posición donde desea pegarlo. Al soltar el texto en su nueva posición aparece un mensaje emergente preguntando si desea pegarlo como texto o como XML.

Formato de texto

Un principio fundamental de los documentos XML es que el contenido se puede separar de su presentación. El documento XML lleva el contenido, mientras que la hoja de estilos contiene la presentación (el formato). En la vista Authentic el documento XML se presenta por medio de la hoja de estilos. Esto significa que todo el formato que ve en la vista Authentic lo produce la hoja de estilos. Si ve texto en negrita, esta opción de formato viene dada por la hoja de estilos. Si ve una lista o una tabla, esa opción de formato también viene dada por la hoja de estilos. El documento XML, que usted edita en la vista Authentic, contiene únicamente el contenido, pero no formato. El formato está definido en la hoja de estilos. Esto significa que en la vista Authentic el usuario no puede aplicar formato al texto que está editando, porque está editando contenido. El formato que se aplica automáticamente al contenido que está editando está conectado al valor semántico o estructural de los datos que está editando. Por ejemplo, a una dirección de correo electrónico (que puede entenderse como una unidad semántica) se le aplica determinado formato porque se trata de una dirección de correo electrónico. Asimismo, un título debe aparecer en una posición determinada del documento (es una unidad semántica y estructural) y se le aplica determinado formato en base a lo indicado por el diseñador de la hoja de estilos. Por tanto, el usuario no puede cambiar el formato ni de las direcciones de correo electrónico ni de los títulos. Lo único que puede hacer es editar el contenido de las direcciones de correo electrónico y de los títulos.

En algunos casos es necesario presentar el contenido de manera especial. Por ejemplo, quizás sea necesario presentar cierta cadena de texto en negrita. En ese caso, la presentación debe estar vinculada a un elemento estructural del documento. Por ejemplo, el diseñador de la hoja de estilos separa estructuralmente la cadena de texto del resto de contenido. Si usted necesita usar dicha cadena de texto en la vista Authentic, debe introducir la cadena de texto dentro del marcado adecuado. Para más información, consulte el comando **Insertar elemento** del apartado Ayudante de entrada Elementos 620.

Usar RichEdit en la vista Authentic

En la vista Authentic los botones y controles de la barra de herramientas RichEdit (*imagen siguiente*) se habilitan cuando se coloca el cursor dentro de un elemento que se creó como componente RichEdit. De lo contrario, las opciones de esta barra de herramientas aparecen atenuadas.

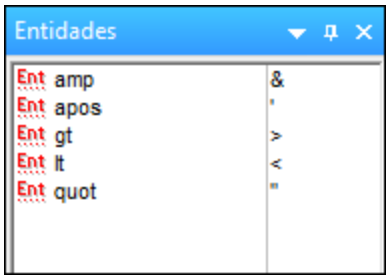
© 2016-2022 Altova GmbH



Seleccione el texto que desea editar y después utilice los botones y controles de la barra de herramientas RichEdit para seleccionar el estilo que desea aplicar al texto. RichEdit permite al usuario de la vista Authentic especificar la fuente, su peso, estilo, decoración, tamaño, color, color de fondo y alineación. El texto se incluirá entre las etiquetas que corresponden a los estilos elegidos.

Insertar entidades

En los documentos XML algunos caracteres no se pueden utilizar como texto normal porque son caracteres XML utilizados para el marcado. Se trata de los caracteres et (`&`), apóstrofe (`'`), menor que (`<`), mayor que (`>`) y comillas inglesas (`"`). Si desea usar estos caracteres en el texto del documento, debe insertarlos como referencias de entidad con ayuda del ayudante de entrada Entidades 623 (*imagen siguiente*).



En XML también puede crear sus propias entidades. Puede crear entidades para (i) caracteres especiales que no están disponibles en su teclado, (ii) cadenas de texto que desea utilizar con frecuencia en su documento, (iii) fragmentos de código XML o (iv) imágenes y otros recursos. Estas entidades se pueden definir 648 en la vista Authentic directamente. Después de definirlas, estas entidades aparecen en el ayudante de entrada Entidades 623 y se pueden insertar en el documento.

Insertar secciones CDATA

Las secciones CDATA son secciones de texto de un documento XML que el analizador XML no procesa como datos XML. Pueden utilizarse para escapar secciones de texto grande, como fragmentos de código de programa o XML que el usuario desea reproducir con sus etiquetas de marcado. Las secciones CDATA pueden aparecer dentro del contenido de un elemento y están delimitadas por `<![CDATA[` y `]]>`. Esto significa que la cadena de texto `]]>` no se debe utilizar dentro de una sección CDATA puesto que ello indicaría el fin de la sección. Si desea utilizar la cadena `]]>` dentro de la sección CDATA, entonces debe sustituir el carácter `>` por su referencia de entidad (`>`). Para insertar una sección CDATA dentro de un elemento, haga clic con el botón derecho en la posición elegida y seleccione **Insertar sección CDATA** en el menú contextual. Para ver las etiquetas de la sección CDATA en la vista Authentic, active la visualización del marcado 614. Otra manera de crear una sección CDATA es resaltar el texto y después seleccionar el comando **Insertar sección CDATA**.

Nota: las secciones CDATA no se pueden insertar en campos de entrada (es decir, en cuadros de texto y cuadros de texto multilínea). Las secciones CDATA sólo se pueden introducir en elementos que aparecen como componentes de texto en la vista Authentic.

Editar y usar hipervínculos

Un hipervínculo está formado por el texto del vínculo y el destino del vínculo. Para editar el texto del hipervínculo haga clic en el texto y edítelo. El destino del hipervínculo no se puede editar. (El destino del

hipervínculo lo define el diseñador de la hoja de estilos, escribiendo la dirección de destino estática o derivando la dirección de destino de los datos que contiene el documento XML). Para ir al destino del hipervínculo desde la vista Authentic haga clic en el texto del hipervínculo mientras pulsa la tecla **Ctrl**. (Recuerde que si solo hace clic en el hipervínculo entonces se habilita la edición del texto del hipervínculo.)

11.3.2 Tablas en la vista Authentic

Hay tres tipos de tablas, que a su vez se agrupan en dos categorías: tablas SPS (estáticas y dinámicas) y tablas CALS/HTML.

Hay dos tipos de **tablas SPS**: estáticas y dinámicas. Las tablas SPS las diseña el diseñador de la hoja de estilos SPS a la que está vinculado el documento XML. Usted no puede insertar una tabla SPS en el documento XML, pero puede introducir datos en los campos de las tablas SPS y agregar y eliminar filas en las tablas SPS dinámicas. El apartado tablas SPS 632 que aparece a continuación explica las características de este tipo de tablas.

Lo que sí puede insertar el usuario de la vista Authentic son **tablas CALS/HTML**. El propósito de estas tablas es que el usuario pueda insertar tablas en cualquier posición de la jerarquía del documento. Las características de edición de las tablas CALS/HTML 633 y de los iconos de edición para tablas CALS/HTML 637 se describen en los apartados siguientes.

11.3.2.1 Tablas SPS

En la vista Authentic se pueden usar dos tipos de tablas SPS: estáticas y dinámicas.

Las **tablas estáticas** tienen una estructura y un tipo de contenido fijos. El usuario de la vista Authentic puede introducir datos en las celdas de la tabla pero no puede cambiar su estructura (es decir, no puede agregar filas ni columnas, etc.) ni cambiar el tipo de contenido de las celdas. Puede introducir los datos a mano o seleccionando una de las opciones disponibles en forma de casilla de verificación, botón de opción o lista desplegable de cuadros combinados.

Nanonull, Inc.

Street:	119 Oakstreet, Suite 4876	**Phone:**	+1 (321) 555 5155
City:	Vereno	**Fax:**	+1 (321) 555 5155 - 9
State & Zip:	DC 29213	**E-mail:**	office@nanonull.com

Nota importante: los iconos y comandos para editar tablas dinámicas **no se deben usar** para editar tablas estáticas.

Las **tablas dinámicas** tienen filas que representan una estructura de datos repetitiva, es decir, cada fila tiene una estructura de datos idéntica (al contrario de las tablas estáticas). Por tanto, en este tipo de tablas puede

 © 2016-2022 Altova GmbH

anexar, insertar, subir, bajar y eliminar filas. Estos comandos están disponibles en el menú **Authentic**, así como en la barra de herramientas (*ver iconos de la imagen*).



Para realizar estas operaciones, ponga el cursor en cualquier posición dentro de la fila que desea editar y después seleccione el comando pertinente.

Administration

First	Last	Title	Ext	EMail	Shares	Leave		
						Total	Used	Left
Vernon	**Callaby**	Office Manager	581	v.callaby@nanonull.com	1500	25	4	21
Frank	Further	Accounts Receivable	471	f.further@nanonull.com	0	22	2	20
Loby	Matise	Accounting Manager	963	l.matise@nanonull.com	add Shares	25	7	18
Employees: 3 (20% of Office, 9% of Company)					**Shares: 1500 (13% of Office, 6% of Company)**			
Non-Shareholders: Frank Further, Loby Matise.								

Para desplazarse por las celdas de la tabla, use las teclas de dirección. Para avanzar de una celda a la siguiente, use la tecla **Tabulador**. Si el cursor está en la última celda de la última fila y pulsa la tecla **Tabulador**, se crea una fila nueva.

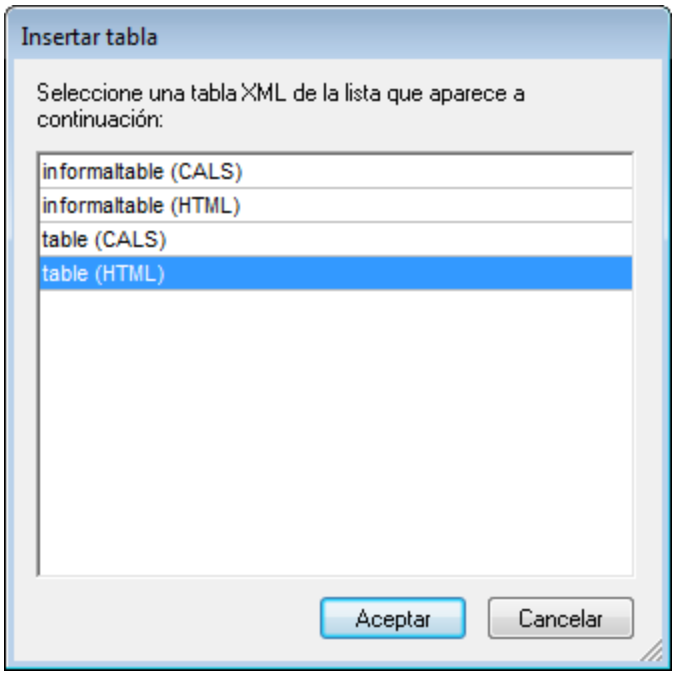
11.3.2.2 Tablas CALS/HTML

El usuario de la vista Authentic puede insertar tablas CALS/HTML para aquellas estructuras de datos XML que se diseñaran para ser presentadas en formato de tabla. El trabajo con tablas CALS/HTML implica tres pasos: insertar la tabla, darle formato e introducir los datos. Los comandos para trabajar con tablas CALS/HTML cuentan con iconos en la barra de herramientas (véase Iconos de edición para tablas CALS/HTML 637).

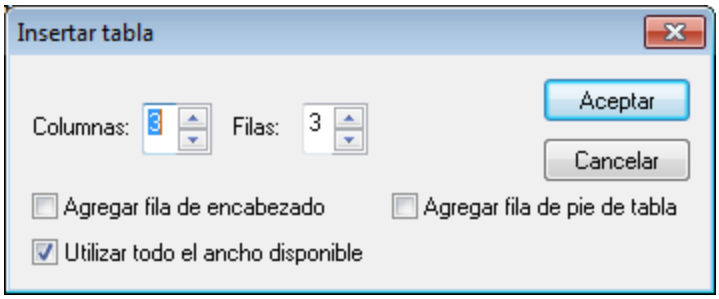
Insertar tablas

Para insertar una tabla CALS/HTML siga estos pasos:

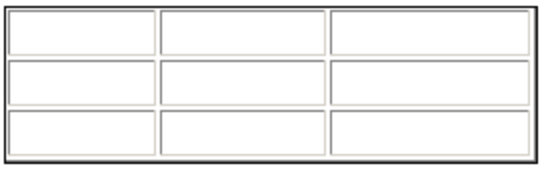
1. Ponga el cursor en la posición donde desea insertar la tabla y haga clic en el icono . (Recuerde que es el esquema el que determina dónde está permitido insertar una tabla). Aparece el cuadro de diálogo "Insertar tabla" (*imagen siguiente*). Este diálogo enumera todas las estructuras de datos de elementos XML para las que se definieron estructuras de tabla. En el ejemplo de la imagen que aparece a continuación puede ver que los elementos `table` y `informaltable` se definieron como tablas CALS y HTML.



2. Seleccione el elemento y el tipo de tabla que desea insertar y haga clic en **Aceptar**.
3. Aparece otro diálogo (*imagen siguiente*). Seleccione el número de columnas y filas para la tabla, indique si desea añadir un encabezado o pie de tabla y si la tabla debe ocupar todo el ancho de la página. Después haga clic en **Aceptar**.



Con la configuración de la imagen anterior se obtendría esta tabla:



Con los comandos del menú **Tabla** puede agregar y eliminar columnas, así como combinar y dividir filas y columnas. Pero para empezar debe crear la estructura general de la tabla.

Cómo dar formato a las tablas e introducir datos

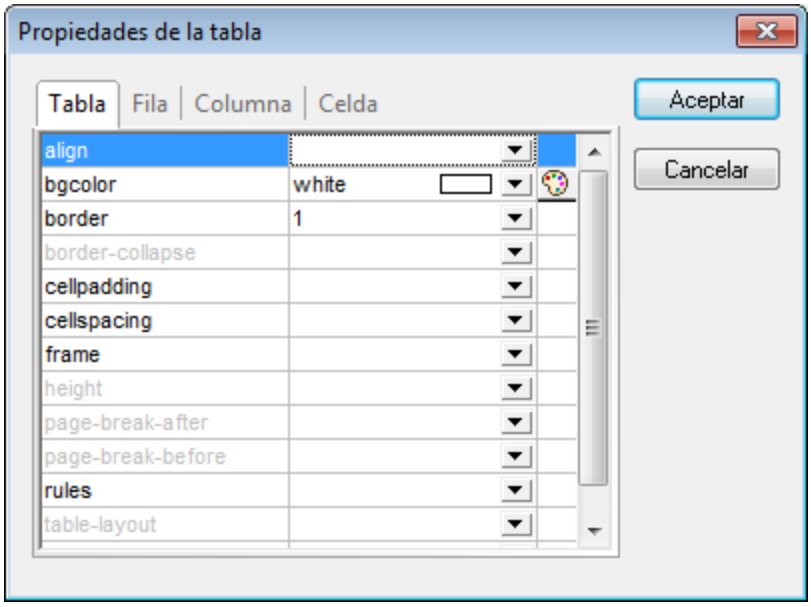
En el diseño del documento ya se asignó un formato a la tabla. No obstante, puede modificar el formato de la tabla en algunas circunstancias:

© 2016-2022 Altova GmbH

- Los elementos que se corresponden a los elementos de estructura de la tabla deben tener definidos como atributos (en el esquema XML subyacente) las propiedades de tabla CALS o HTML correspondientes. Para el formato de la tabla sólo se pueden usar los atributos que están definidos en el esquema XML. Si en el diseño se establecieron valores para estos atributos, entonces puede cambiar los valores de estos atributos en la vista Authentic.
- En el diseño el atributo `style` no puede contener estilos CSS. Si se definió un atributo `style` con estilos CSS para un elemento, entonces el atributo `style` tiene prioridad sobre el resto de los atributos de formato del elemento. Como resultado, se reemplazan las opciones de formato indicadas en la vista Authentic.

Siga estos pasos para dar formato a una tabla, fila, columna o celda:

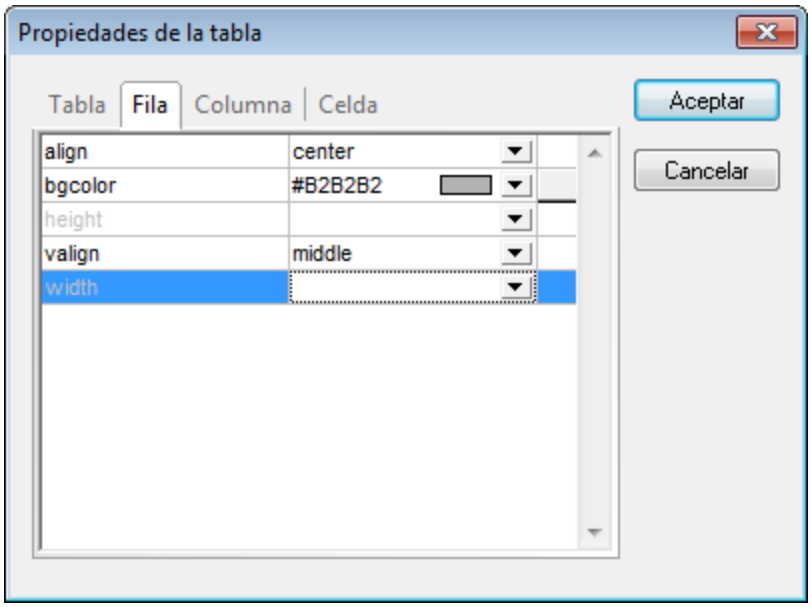
1. Ponga el cursor dentro de la tabla y haga clic en el icono  (Propiedades de la tabla). Aparece el cuadro de diálogo "Propiedades de la tabla" (*imagen siguiente*), donde puede indicar el formato de la tabla, de una fila, de una columna o de una celda.



2. Defina un valor "0" para las propiedades `cellpadding` y `cellspacing` y haga clic en **Aceptar**. La tabla tendrá este aspecto:



3. Ponga el cursor en la primera fila y haga clic en el icono  (Propiedades de la tabla). Aparece el cuadro de diálogo "Propiedades de la tabla". Haga clic en la pestaña **Fila**.



Vamos a usar la primera fila de la tabla como encabezado así que vamos a darle un color diferente para diferenciarla de las demás filas. Configure las propiedades de la primera fila tal y como muestra la imagen anterior y después escriba el texto de encabezado de cada columna. La tabla debería tener este aspecto:

Nombre	Teléfono	Correo electrónico

Observe que el texto está alineado en el centro, tal y como indica la propiedad `align`.

4. Ahora vamos a dividir la columna "Teléfono" en dos subcolumnas: "Oficina" y "Particular". Para ello debemos dividir el ancho horizontal de la columna "Teléfono" en dos columnas. Pero antes debemos dividir la celda de encabezado en dos filas. Ponga el cursor en la celda "Teléfono" y haga clic en el icono  (Dividir verticalmente). La tabla debería tener este aspecto:

Nombre	Teléfono	Correo electrónico

5. Ahora ponga el cursor en la celda situada debajo de la celda "Teléfono" y haga clic en el icono  (Dividir horizontalmente). Escriba los encabezados de columna "Oficina" y "Particular". La tabla debería tener este aspecto:

Nombre	Teléfono		Correo electrónico
	Oficina	Particular	

 © 2016-2022 Altova GmbH

El siguiente paso es dividir horizontalmente cada celda de la columna "Teléfono".

También puede agregar y eliminar columnas y filas, así como alinear verticalmente el contenido de las celdas, con los iconos de edición para tablas. Los iconos se describen en el apartado siguiente, titulado Iconos de edición para tablas CALS/HTML 637.

Desplazarse por las celdas de la tabla

Para desplazarse por las celdas de la tabla CALS/HTML utilice las flechas de dirección Arriba, Abajo, Derecha e Izquierda.

Introducir datos en una celda

Para introducir datos en una celda, ponga el cursor en la celda y teclee los datos.

Formato de texto

Al texto de las tablas CALS/HTML, al igual que a todo el texto del documento XML, se le da formato a través de elementos y atributos XML. Para añadir un elemento, resalte el texto y haga doble clic en el elemento pertinente en el ayudante de entrada Elementos. Para indicar un valor de atributo, ponga el cursor dentro del fragmento de texto y escriba el valor de atributo en el ayudante de entrada Atributos. Por ejemplo, puede poner en negrita el texto del encabezado de la tabla del ejemplo. Para ello resalte el texto y haga doble clic en el elemento `strong` en el ayudante de entrada Elementos. Para el elemento `strong` existe una plantilla global cuyo peso de fuente es `bold`.

Nota importante: para que el formato de texto aparezca en la vista Authentic, es necesario crear en StyleVision una plantilla global con formato de texto para el elemento en cuestión.

11.3.2.3 Iconos de edición para tablas CALS/HTML

En este apartado describimos los comandos necesarios para editar tablas CALS/HTML, que están disponibles en forma de iconos en la barra de herramientas. Estos comandos no están disponibles en ningún menú.

Para más información sobre cómo usar estos iconos, consulte el apartado Tablas CALS/HTML 633.

Insertar tabla



Este comando inserta una **tabla CALS/HTML** en la posición del cursor.

Eliminar tabla



Este comando elimina la tabla activa.

Anexar fila



Este comando anexa una fila al final de la tabla activa.

Anexar columna



Este comando anexa una columna al final de la tabla activa.

Insertar fila



Este comando inserta una fila encima de la posición del cursor en la tabla activa.

Insertar columna



Este comando inserta una columna a la izquierda de la posición del cursor en la tabla activa.

Combinar celda hacia la izquierda



Este comando une la celda actual (donde se encuentra el cursor) con la celda de la izquierda. La celda nueva incluye las etiquetas de ambas celdas, el contenido de las celdas se encadena y los encabezados siguen como estaban.

Combinar celda hacia la derecha



Este comando une la celda actual (donde se encuentra el cursor) con la celda de la derecha. El contenido de las celdas se encadena en la celda nueva.

Combinar celda hacia abajo



Este comando une la celda actual (donde se encuentra el cursor) con la celda situada justo debajo. El contenido de las celdas se encadena en la celda nueva.

Combinar celda hacia arriba



Este comando une la celda actual (donde se encuentra el cursor) con la celda situada justo encima. El contenido de las celdas se encadena en la celda nueva.

Dividir horizontalmente



Este comando crea una celda nueva a la derecha de la celda activa. El tamaño de las celdas resultantes es el mismo que el de la celda original.

Dividir verticalmente



Este comando crea una celda nueva debajo de la celda activa.

Alinear arriba



Este comando alinea el contenido en la parte superior de la celda.

Centrar verticalmente



Este comando alinea el contenido en el centro de la celda.

Alinear abajo



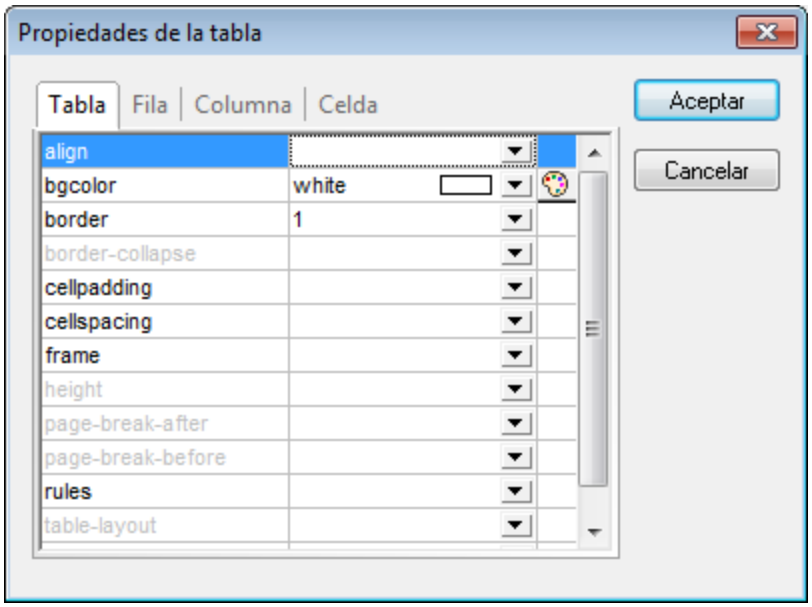
Este comando alinea el contenido en la parte inferior de la celda.

Propiedades de la tabla



Este comando abre el cuadro de diálogo "Propiedades de la tabla". Este icono solo está habilitado para las tablas HTML, no para las tablas CALS.

© 2016-2022 Altova GmbH



11.3.3 Editar una BD

En la vista Authentic puede editar tablas de base de datos (BD) y guardarlas en la BD. Este apartado ofrece una descripción de las funciones disponibles mientras se edita una tabla de BD. Antes de continuar, es necesario señalar estos aspectos generales:

- El diseñador de la hoja de estilos SPS puede limitar el número de registros de una tabla de BD que se presentan en la vista Authentic para que el diseño tenga un aspecto más compacto. Si así fuera, en la vista Authentic se carga únicamente ese número de registros. El usuario puede cargar y visualizar los demás registros de la tabla de BD con ayuda de los iconos de navegación (véase *Navegar por una tabla de BD* 639).
- Puede consultar la BD 640 para ver determinados registros.
- Puede agregar, modificar y eliminar registros de la BD y guardar los cambios en la BD. Consulte el apartado Modificar una tabla de BD 644.

Para abrir una hoja de estilos SPS basada en una BD en la vista Authentic:

- Haga clic en **Authentic** | **Editar datos de una base de datos** y busque el archivo SPS.

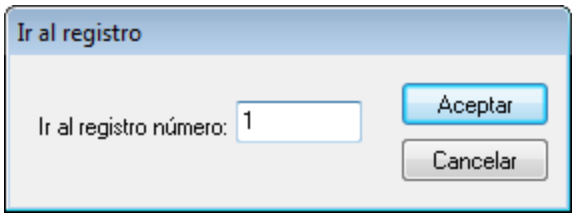
Nota: en la vista Authentic los datos procedentes de bases de datos SQLite no se pueden editar. Si intenta guardar datos SQLite desde la vista Authentic, aparece un cuadro de mensaje informando de esta restricción.

11.3.3.1 Navegar por una tabla de BD

Los comandos necesarios para navegar por las filas de la tabla de BD están disponibles en forma de botones en el documento de la vista Authentic. Por lo general, cada tabla de BD viene acompañada de un panel de navegación que incluye cuatro o cinco botones.



Los iconos, de izquierda a derecha, significan: Ir al primer registro de la tabla de BD; Ir al registro anterior; Abrir el cuadro de diálogo "Ir al registro" (*ver imagen siguiente*); Ir al registro siguiente; e Ir al último registro de la tabla.



Pulse el botón pertinente para navegar por la tabla de la BD.

Bases de datos XML

En caso de trabajar con bases de datos XML, como IBM DB2, en las que una celda (o fila) contenga un documento XML, en la vista Authentic se cargan una por una las filas de la BD. Para cargar un documento XML que esté en otra fila, use el comando de menú **Authentic | Seleccionar y editar una fila nueva con datos XML** 1414.

11.3.3.2 Consultas de BD

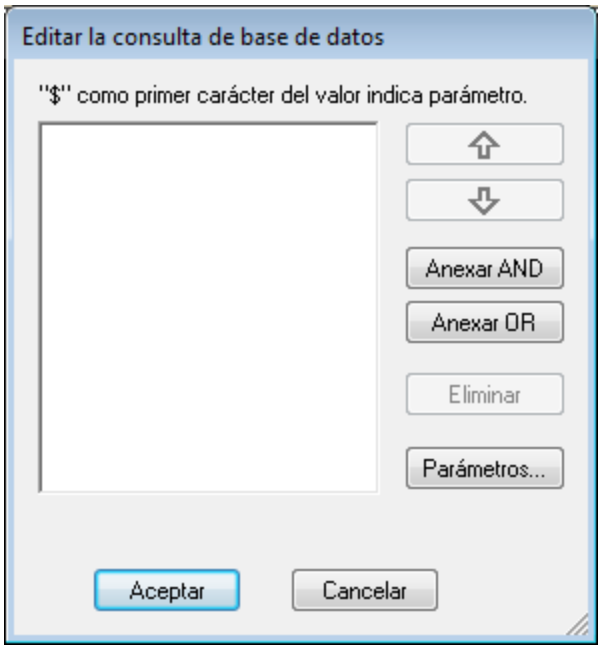
Las consultas de BD sirven para consultar los registros de una tabla de la vista Authentic. Por cada tabla se puede realizar una sola consulta y por cada consulta se puede consultar una sola tabla. Si al enviar la consulta hubiese cambios sin guardar en el documento de la vista Authentic, se le preguntará si desea guardar **todos** los cambios realizados en el documento o descartar **todos** los cambios. Tenga en cuenta que se guardarán/descartarán los cambios realizados en todas las tablas. Después de enviar la consulta la tabla se recarga utilizando las condiciones de la consulta.

Nota importante: si recibe un mensaje diciendo que hay demasiadas tablas abiertas, puede reducir el número de tablas que están abiertas con ayuda de una consulta.

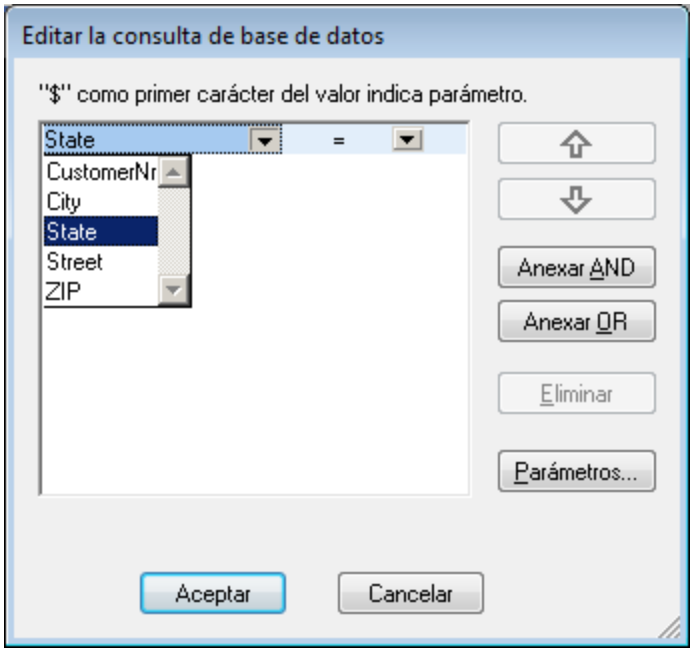
Para crear y enviar una consulta:

1. Haga clic en el botón de consulta  para abrir el cuadro de diálogo "Editar la consulta de la base de datos" (*ver imagen*). Este botón suele aparecer encima de cada tabla de BD o justo debajo. Si no hay un botón de consulta para su tabla, significa que el diseñador de la hoja de estilos SPS no habilitó la función de consulta de BD para esa tabla.

© 2016-2022 Altova GmbH



2. Haga clic en el botón **Anexar AND** o **Anexar OR**. Estos botones anexan un criterio vacío para la consulta (*imagen siguiente*).



4. Escriba la expresión para el criterio de la consulta. Las expresiones están compuestas por (i) un nombre de campo (puede elegir uno en la lista del cuadro combinado), (ii) un operador (puede elegir uno en la lista del cuadro combinado) y (iii) un valor (que debe introducir a mano). Para más información sobre cómo construir expresiones, lea el subapartado Uso de expresiones en criterios de consulta 642.
5. Si desea añadir otro criterio, haga clic en los botones **Anexar AND** o **Anexar OR** según el operador lógico que desee utilizar para unir los dos criterios (AND o OR). Después añada el criterio nuevo. Para

más información sobre los operadores lógicos, lea el subapartado Reordenar criterios en las consultas de BD 643.

Uso de expresiones en criterios de consulta

Las expresiones de los criterios de consulta de BD están formados por un nombre de campo, un operador y un valor. Los **nombres de campo disponibles** son los elementos secundarios de la tabla de datos de nivel superior seleccionada. Los nombres de estos campos aparecen en la lista del cuadro combinado (imagen anterior). Los **operadores** que puede usar en las expresiones son:

=	Igual a
<>	No igual
<	Menos que
<=	Menos o igual que
>	Más que
>=	Más o igual que
LIKE	Fonéticamente igual
NOT LIKE	Fonéticamente no igual
IS NULL	Está vacío
NOT NULL	No está vacío

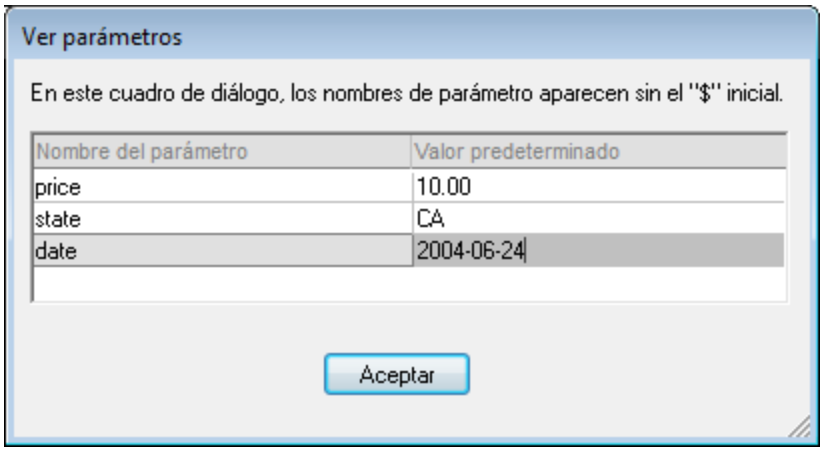
Si selecciona la opción IS NULL o NOT NULL, el campo Valor se deshabilita. Los **valores** deben introducirse sin comillas (ni ningún otro delimitador). También es necesario que los valores tengan el mismo formato que su campo de BD correspondiente. De lo contrario, la expresión dará `FALSE` como resultado. Por ejemplo, si un criterio de consulta para un campo del tipo de datos `date` de una BD de MS Access tiene una expresión `StartDate=25/05/2004`, la expresión dará `FALSE` como resultado porque el tipo de datos `date` de las BD de MS Access tiene el formato `YYYY-MM-DD`.

Usar parámetros en consultas de BD

En su consulta puede introducir el nombre de un **parámetro** como valor de una expresión. Los parámetros son variables que se pueden usar en las consultas en lugar de valores literales. Cuando introduzca un parámetro en una expresión, en dicha expresión se usará el valor del parámetro. Los parámetros disponibles son los que definió el diseñador de la hoja de estilos en el archivo SPS. Los parámetros disponibles se enumeran en el cuadro de diálogo "Ver parámetros" (*imagen siguiente*). A los parámetros se les asigna un valor predeterminado en el archivo SPS. Este valor se puede reemplazar pasando otro valor al parámetro por medio de la línea de comandos (siempre y cuando el documento de salida se compile por medio de la línea de comandos).

Para ver qué parámetros se definieron en el archivo SPS pulse el botón **Parámetros** del cuadro de diálogo "Editar la consulta de base de datos". Como resultado aparece el cuadro de diálogo "Ver parámetros" (*imagen siguiente*).

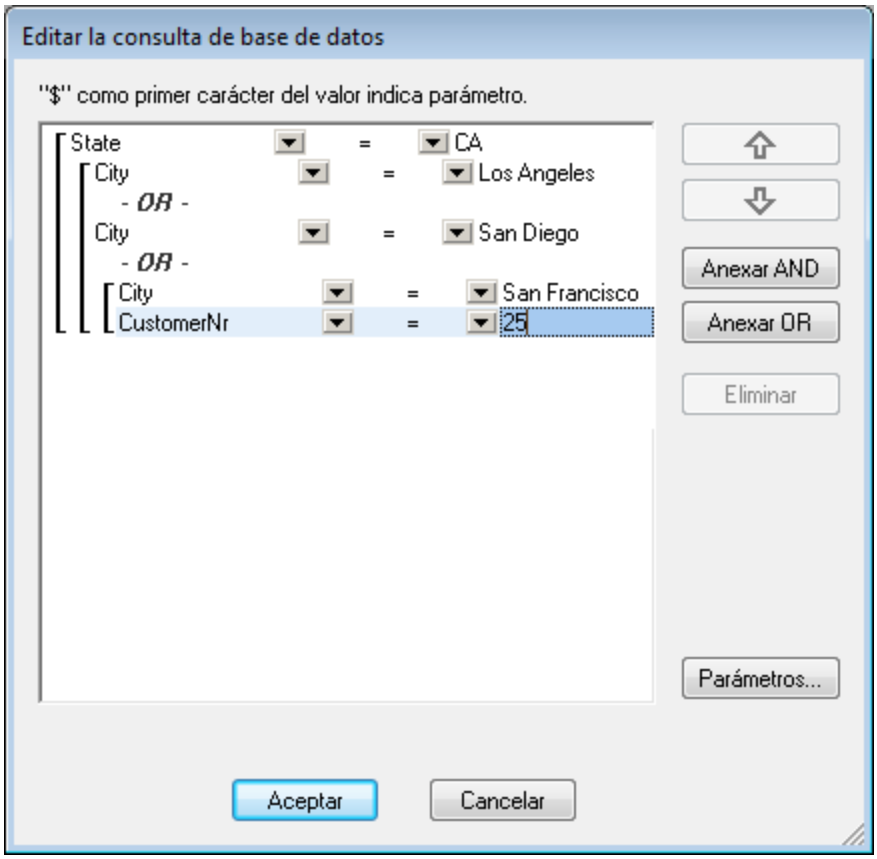
© 2016-2022 Altova GmbH



El cuadro de diálogo "Ver parámetros" incluye todos los parámetros que se definieron para la hoja de estilos en el archivo SPS. Los parámetros se pueden editar en el diseño de la hoja de estilos.

Reordenar criterios en las consultas de BD

La estructura lógica de la consulta de BD y la relación existente entre los criterios se indica de forma gráfica. Cada nivel de la estructura lógica se indica con un corchete. Los criterios o conjuntos de criterios adyacentes van unidos por el operador AND, mientras que si dos criterios están separados con la palabra `OR`, entonces se usa el operador OR. A los criterios también se aplica sangría para ofrecer una visión de conjunto de la estructura lógica de la consulta de la BD.



La consulta de BD de la imagen anterior se leería así:

```
State=CA AND (City=Los Angeles OR City=San Diego OR (City=San Francisco AND
CustomerNr=25))
```

Puede reordenar la consulta subiendo o bajando un criterio o conjunto de criterios. Para mover un criterio o conjunto de criterios, siga estos pasos:

1. Para seleccionar un criterio, haga clic en él. Para seleccionar un nivel entero, haga clic en el corchete que abarca dicho nivel.
2. Haga clic en uno de los dos botones en forma de flecha según corresponda.

Antes de subir o bajar un criterio debe tener en cuenta varias cuestiones:

- Si el criterio situado justo encima o debajo está al mismo nivel, el criterio elegido toma la posición de otro y viceversa.
- Si sube o baja un conjunto de criterios (es decir, criterios que están dentro de un corchete), el conjunto cambia de posición dentro del mismo nivel, no cambia de nivel.
- Si el criterio situado justo encima o debajo está en otro nivel (superior o inferior), el criterio seleccionado pasará al siguiente nivel (superior o inferior), **nivel por nivel**.

Para eliminar un criterio de una consulta de BD, selecciónelo y haga clic en **Eliminar**.

Modificar una consulta de BD

Para modificar una consulta de BD:

1. Haga clic en el botón de consulta . Se abre el cuadro de diálogo "Editar la consulta de base de datos". Ahora puede editar las expresiones de los criterios, agregar criterios nuevos, reordenarlos o eliminarlos.
2. Haga clic en **Aceptar**. Lo datos de la BD se recargan automáticamente en la vista Authentic, para reflejar los cambios realizados en la consulta de BD.

11.3.3.3 Modificar una tabla de BD

Agregar un registro

Para agregar un registro a una tabla de BD:

1. Ponga el cursor en la fila de la tabla de BD y haga clic en el icono  (para anexar una fila) o en  (para insertar una fila). Como resultado se crea un registro nuevo en el archivo XML temporal.
2. Haga clic en el comando **Archivo | Guardar** para agregar el registro nuevo en la BD. En la vista Authentic se anexa una fila nueva para el registro nuevo en la tabla de BD. El valor del campo `AltovaRowStatus` de este registro pasa a ser `A` (de "Added", es decir, agregado).

Los datos que introduzca para el registro nuevo aparecen en negrita y subrayados. Así puede distinguir los registros agregados de los registros ya existentes (siempre que no se aplicase ese mismo formato a ningún registro ya existente). Los errores de tipos de datos aparecen en color rojo para que pueda detectarlos inmediatamente.

© 2016-2022 Altova GmbH

El nuevo registro se agrega a la BD nada más hacer clic en **Archivo | Guardar**. Una vez agregado el nuevo registro a la BD, su campo `AltovaRowStatus` se inicializa (se indica con `---`) y el registro aparece en la vista Authentic como cualquier otro registro.

Modificar un registro

Para modificar un registro, ponga el cursor en el registro de la tabla de BD y edítelo. Si se limitó el número de registros que se pueden mostrar, entonces quizás tenga que navegar hasta el registro (véase Navegar por una tabla de BD 639).

Cuando se modifica un registro, las entradas de todos los campos del registro se subrayan y el valor del campo `AltovaRowStatus` de las instancias principales de este registro pasa a ser `U` (de "Updated", es decir, actualizado). En las instancias secundarias de este registro el valor del campo `AltovaRowStatus` pasa a ser `u` (en minúsculas). Las instancias principales y secundarias de un registro están definidas por la estructura de la BD (y del esquema XML generado a partir de ella). Por ejemplo, si se incluye una tabla Dirección en una tabla Cliente, entonces la tabla Dirección puede aparecer en el documento de diseño en dos tipos de instancias: como tabla Dirección propiamente dicha y dentro de las instancias de la tabla Cliente. El tipo de instancia que se modifique es el tipo principal. Los demás tipos son los tipos secundarios. Los errores de tipo de datos aparecen en color rojo para que pueda detectarlos inmediatamente.

Los cambios se guardan en la BD al hacer clic en **Archivo | Guardar**. Después de guardar en la BD un registro en el que se realizaron cambios, su campo `AltovaRowStatus` se inicializa (se indica con `---`) y el registro aparece en la vista Authentic como cualquier otro registro.

Nota importante:

- Si se modifica un solo campo del registro en la vista Authentic, al guardar los datos en la BD se actualiza el registro entero.
- El valor de fecha `0001-01-01` se define como valor `NULL` en algunas BD, lo cual podría dar lugar a errores.

Eliminar un registro

Para eliminar un registro:

1. Ponga el cursor en la fila del registro que desea eliminar y haga clic en el icono . El registro que se va a eliminar aparece tachado. El campo `AltovaRowStatus` toma el valor `D` en las instancias principales del registro, `d` en las instancias secundarias y `X` en los registros eliminados de forma indirecta. Los registros eliminados de forma indirecta son aquellos campos del registro eliminado que se almacenan en otra tabla diferente. Por ejemplo, la tabla Dirección puede estar incluida en una tabla Cliente. Si se elimina un registro de Cliente, su registro Dirección correspondiente se elimina de forma indirecta. Si se elimina un registro Dirección de la tabla Cliente, entonces el registro Dirección de la tabla Cliente se elimina de forma principal, pero el mismo registro se elimina de forma secundaria en la tabla independiente Dirección si se creó una instancia de esta tabla.
2. Haga clic en **Archivo | Guardar** para guardar los cambios en la BD.

Nota importante: después de guardar los cambios en la BD el comando **Deshacer** se restablece. Es decir, ya no podrá deshacer las acciones llevadas a cabo antes de guardar los cambios.

11.3.4 Trabajar con fechas

En la vista Authentic las fechas se pueden editar de dos maneras diferentes:

- Con ayuda del selector de fecha[646].
- O escribiendo el valor[647] de la fecha a mano.

El método al que tiene acceso el usuario de la vista Authentic lo define el diseñador de la hoja de estilos en el archivo SPS. En los siguientes apartados describimos ambos métodos más detalladamente.

Nota sobre los formatos de fecha

En el documento XML las fechas se almacenan en un tipo de datos de fecha. Para que el documento XML sea válido, la fecha debe almacenarse en un formato léxico concreto, dependiendo del tipo de datos. Por ejemplo, para el tipo de datos `xs:date` la fecha debe tener el formato léxico `YYYY-MM-DD`. Si en un nodo con el tipo de datos `xs:date` introduce una fecha que no se ajusta a este formato, el documento XML no será válido.

Para garantizar que las fechas se introduzcan en el formato correcto, el diseñador del SPS puede incluir en el diseño un selector de fecha gráfico. Así se asegura de que la fecha seleccionada en el selector de fecha se introduce en el formato léxico correcto. Si no hay un selector de fecha, el usuario de la vista Authentic debe validar el documento XML para comprobar si todas las fechas introducidas tienen el formato léxico correcto.

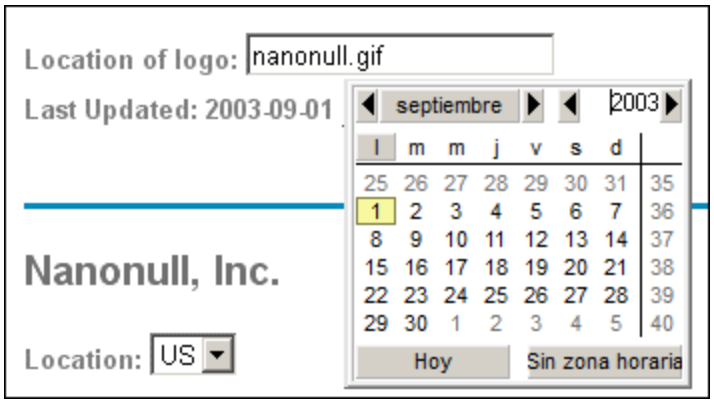
11.3.4.1 Selector de fecha

El selector de fecha es un calendario gráfico que sirve para introducir fechas en un formato estándar en el documento XML. Es importante tener un formato estándar para procesar los datos del documento. El icono del selector de fecha aparece junto al campo de fecha al que modifica (*ver imagen*).



Para abrir el selector, haga clic en el icono del selector de fecha (*ver imagen*).

© 2016-2022 Altova GmbH

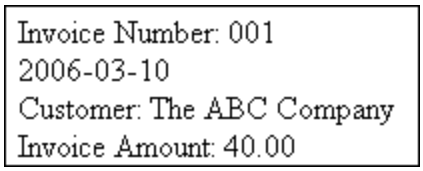


Para seleccionar una fecha, haga clic en la fecha, el mes y el año. Los datos introducidos en el documento XML y los datos de la vista Authentic se actualizan en función de la fecha seleccionada en el selector. Si lo desea también puede introducir la zona horaria.

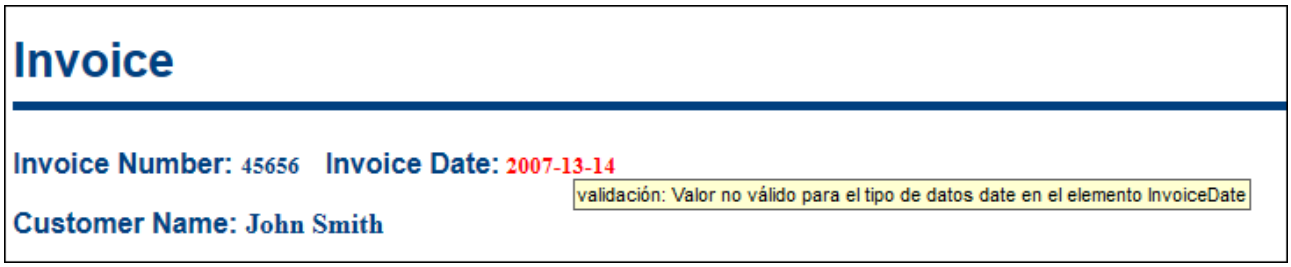
11.3.4.2 Entrada de texto

En los campos de fecha que no tengan un selector de fecha (*ver imagen*), puede editar la fecha directamente con solo teclear la fecha nueva.

Nota: no cambie el formato al cambiar la fecha.



Si edita una fecha y la fecha nueva no está dentro del rango válido, la fecha se vuelve de color rojo para informarle del error. Si pasa el puntero del ratón sobre la fecha no válida, aparece un mensaje de error (*ver imagen*).



Si intenta cambiar el formato de la fecha, la fecha se vuelve de color rojo para informarle del error (*ver imagen*).

Invoice Number: 001
2006/03/10
Customer: The ABC Company
Invoice Amount: 40.00

11.3.5 Definir entidades

Acerca de las entidades

Puede definir las entidades que se deben usar en la vista Authentic, tanto si el documento está basado en una DTD como en un esquema XML. Una vez definidas, las entidades aparecen en el ayudante de entrada Entidades y en el submenú del menú contextual. Al hacer doble clic en una entidad del ayudante de entrada Entidades, la entidad se inserta en el punto de inserción del cursor.

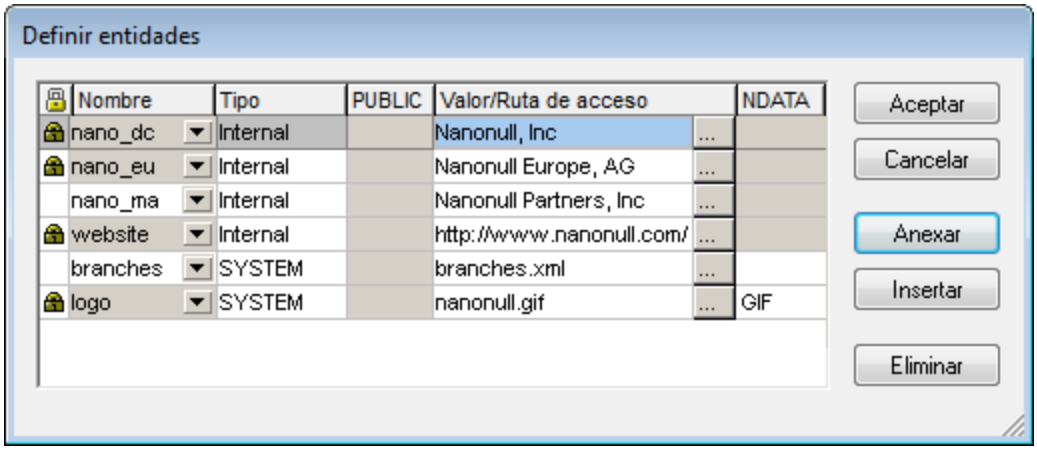
Puede usar entidades para las cadenas de texto, fragmentos XML y recursos externos que vaya a utilizar repetidas veces en el documento. Las entidades (que no son más que una abreviatura de los datos) se definen en el cuadro de diálogo "Definir entidades". Una vez definidas, puede usarlas tantas veces como quiera en el documento, lo cual permite ahorrar tiempo y mantener mejor el documento.

Tipos de entidades
Hay dos tipos principales de entidades: las **entidades analizadas**, que son datos XML (ya sea una cadena de texto o un fragmento de un documento XML) y las **entidades sin analizar**, que son datos no XML, como por ejemplo archivos binarios (normalmente objetos gráficos, sonoros o multimedia). Cada entidad tiene un nombre y un valor. En el caso de las entidades analizadas, la entidad es un marcador de posición para los datos XML. El valor de la entidad son los datos XML propiamente dichos o un URI que apunta a un archivo XML que contiene los datos XML. En el caso de las entidades sin analizar, el valor de la entidad es un URI que apunta al archivo no XML.

Definir entidades

Para definir una entidad:

1. Haga clic en **Authentic | Definir entidades XML...**. Se abre el cuadro de diálogo "Definir entidades" (*ver imagen*).



 © 2016-2022 Altova GmbH

2. Escriba el nombre de la entidad en el campo Nombre. Es el nombre que aparecerá en el ayudante de entrada Entidades.
3. Seleccione el tipo de entidad en la lista desplegable del campo Tipo. Los tipos son: **Internal** para entidades cuyo texto está almacenado en el documento XML propiamente dicho. **PUBLIC** y **SYSTEM** indican que el recurso se encuentra fuera del archivo XML y que se localizará por medio de un identificador público o de sistema respectivamente. Un identificador de sistema es un URI que da la ubicación del recurso. Un identificador público es un identificador independiente de la ubicación, que permite a los procesadores identificar el recurso. Si indica tanto un identificador público como de sistema, el identificador público se convierte en identificador de sistema y se usa el identificador de sistema.
4. Si selecciona el tipo PUBLIC, escriba el identificador público del recurso en el campo PUBLIC. Si selecciona el tipo Internal o SYSTEM, entonces se deshabilita el campo PUBLIC.
5. En el campo Valor/Ruta de acceso puede introducir una de estas tres opciones:

   - Si el tipo de entidad es Internal, escriba la cadena de texto que desea usar como valor de la entidad. No escriba comillas para delimitar el texto. Si introduce comillas, estas se tratarán como si fueran parte de la cadena de texto.
   - Si el tipo de entidad es SYSTEM, escriba el URI del recurso o seleccione un recurso de su red local (con el botón Examinar ). Si el recurso contiene datos analizados, entonces debe ser un archivo XML (con la extensión de archivo `.xml`). También puede ser un archivo binario, como un archivo GIF.
   - Si el tipo de entidad es PUBLIC, entonces debe introducir un identificador de sistema en el campo PUBLIC.

6. Los datos introducidos en el campo NDATA indican al procesador que esta entidad no se debe analizar, sino que debe enviarse al procesador adecuado. El campo NDATA debe contener un valor que indique que la entidad es una entidad sin analizar.

Características del cuadro de diálogo

En el cuadro de diálogo "Definir entidades" el usuario puede:

- Anexar entidades.
- Insertar entidades.
- Eliminar entidades.
- Ordenar entidades por orden alfabético en base al contenido de cualquiera de las columnas. Para ello haga clic en el encabezado de la columna pertinente. Un clic ordena el contenido de la columna en orden ascendente, dos clics ordenan el contenido en orden descendente.
- Cambiar el tamaño del cuadro de diálogo y el ancho de las columnas.
- Bloqueo. Cuando se usa una entidad en el documento XML, esta se bloquea y ya no se puede editar en el cuadro de diálogo "Definir entidades". Las entidades bloqueadas se identifican con un símbolo en forma de candado que aparece en la primera columna del cuadro de diálogo. Este bloqueo garantiza la validez del documento XML, en lo que a las entidades se refiere. (El documento no sería válido si se hace referencia a una entidad pero esta no está definida.)
- Las entidades repetidas aparecen marcadas.

Limitaciones de las entidades

- Las entidades que están dentro de otra entidad no se resuelven, ya sea en el diálogo, en la vista Authentic o en el resultado XSLT, y el carácter et de dicha entidad aparece como caracteres de escape, es decir `&`.

- Las entidades sin analizar externas que no sean archivos de imagen no se resuelven en la vista Authentic. Si se define una imagen del diseño para que lea una entidad externa sin analizar y el URI de la imagen se define como un nombre de entidad ("`logo`", por ejemplo), entonces este nombre de entidad se puede definir en el cuadro de diálogo "Definir entidades" (*imagen anterior*) como entidad externa sin analizar con un valor que se convierte en el URI del archivo de imagen (como en el caso de la entidad `logo` de la imagen).

11.3.6 Firmas XML

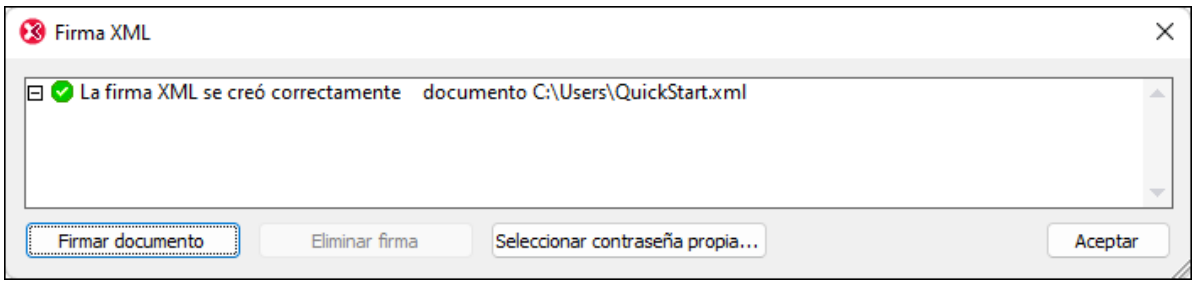
Los archivos SPS se pueden diseñar con una firma XML configurada para la vista Authentic. Cuando se habilita una firma XML en el SPS, el usuario de la vista Authentic puede firmar el archivo XML de Authentic digitalmente con la firma habilitada. Si se realizan cambios en el documento después de firmarlo, la verificación de la firma dará error. Cuando un documento XML de Authentic firmado se abre en la vista Authentic, el proceso de verificación se ejecuta automáticamente en el documento y el resultado de la verificación aparece en una ventana.

Nota: las firmas XML se pueden usar y verificar en la vista Authentic de las ediciones Enterprise y Professional de estos productos de Altova: Authentic Desktop, Authentic Browser, XMLSpy y StyleVision.

Operaciones con firmas XML

Estas son las operaciones con firmas XML que el usuario de la vista Authentic puede realizar:

- *Elegir el certificado o la contraseña:* las firmas se autentifican con un certificado o con una contraseña. Es necesario indicar el objeto de autenticación (el certificado o la contraseña) tanto al crear la firma como al verificarla. Si el documento XML de Authentic tiene asignado un SPS en el que se habilitó la firma XML, el SPS puede especificar un certificado o una contraseña predeterminados para la firma. Se especifique o no el certificado o la contraseña predeterminados, la firma se puede configurar para que el usuario de la vista Authentic pueda seleccionar su propio certificado o contraseña. El usuario de la vista Authentic puede seleccionar su certificado o contraseña en el cuadro de diálogo "Firma XML" (*imagen siguiente*). Al seleccionar un certificado o contraseña, se anula el certificado o contraseña predeterminados. El certificado o la contraseña del usuario se almacena en memoria y se usa en la sesión actual. Si, después de seleccionar el certificado o la contraseña, el usuario de la vista Authentic cierra el archivo o la aplicación, el SPS devuelve el certificado o la contraseña a su configuración predeterminada.



- *Firmar el documento:* el documento XML de Authentic se puede firmar automática o manualmente. La opción de firma automática se especifica en la configuración de la firma en el archivo SPS y hace que el documento XML de Authentic se firme automáticamente cada vez que se guarda. Si la opción de firma automática no está activada, entonces puede firmar el documento manualmente. Para ello, haga

 © 2016-2022 Altova GmbH

clic en el icono de firma XML  de la barra de herramientas o en el comando **Authentic | Firma XML**. Aparece el cuadro de diálogo "Firma XML" (*imagen anterior*). Haga clic en el botón **Firmar documento** de este cuadro de diálogo. Tenga en cuenta que si firma el documento con una firma incrustada, es necesario que el esquema XML autorice el uso del elemento `Signature` como último elemento secundario del elemento raíz. De lo contrario, el documento no será válido. Al firmar el documento, el objeto de autenticación y la ubicación de la firma vienen determinadas por la configuración de la firma. Asegúrese de que tiene la información necesaria sobre la firma y hable con el diseñador de su SPS.

- *Verificar el documento XML de Authentic:* si un SPS tiene firmas XML habilitadas, el proceso de verificación se ejecuta cada vez que se carga el documento XML en la vista Authentic. Si la contraseña o la información de clave del certificado no está guardada con el SPS y con la firma respectivamente, el usuario de la vista Authentic debe introducir la contraseña o seleccionar un certificado para poder verificar el documento. Recuerde que si se genera una firma incrustada, la firma se guarda en el archivo XML cuando este se guarda. La firma XML generada debe quitarse (utilice el botón **Eliminar firma** del cuadro de diálogo "Firma XML", *ver imagen anterior*) si no quiere guardarla con el archivo XML. Asimismo, si se genera una firma separada y no desea guardarla con el archivo XML también debe quitarla.

11.3.7 Imágenes en la vista Authentic

En la vista Authentic puede indicar qué imágenes se deben usar en el documento final de salida (en HTML, RTF, PDF o Word 2007). Tenga en cuenta que es posible que algunos formatos de imagen no sean compatibles con todos los formatos de salida ni con todas las aplicaciones. Por ejemplo, el formato SVG es compatible con PDF, pero no con RTF. Además, para ver este formato en HTML es necesario instalar un complemento para el explorador. Por tanto, cuando seleccione el formato de la imagen, seleccione un formato que sea compatible con el formato de salida de su documento. La mayoría de formatos de imagen son compatibles con todos los formatos de salida (*consulte la lista que aparece más abajo*).

La vista Authentic se basa en Internet Explorer y puede mostrar la mayoría de los formatos de imagen que Internet Explorer puede mostrar. Estos son los formatos de imagen compatibles con la vista Authentic:

- GIF
- JPG
- PNG
- BMP
- WMF (Microsoft Windows Metafile)
- EMF (Enhanced Metafile)
- SVG (sólo para documentos de salida en PDF)

Rutas de acceso relativas

Las rutas de acceso relativas se resuelven en relación al archivo SPS.

11.3.8 Teclas de acceso rápido en la vista Authentic

Tecla Entrar

En la vista Authentic la tecla **Entrar** se usa para anexar elementos adicionales cuando el cursor está situado en ciertas posiciones. Por ejemplo, si el capítulo de un libro puede tener varios párrafos (porque el esquema lo permite), al pulsar la tecla **Entrar** dentro del texto de un párrafo se crea un párrafo nuevo inmediatamente después del párrafo actual. Si el capítulo puede tener un título y varios párrafos, al pulsar la tecla **Entrar** dentro del capítulo pero no dentro de un párrafo (incluido el elemento Título), se crea un capítulo nuevo justo después del capítulo actual (siempre y cuando el esquema autorice varios capítulos).

Nota: la tecla **Entrar no inserta** una línea nueva. Incluso cuando el cursor está dentro de un nodo de texto, como un párrafo por ejemplo.

Uso del teclado

Puede usar el teclado igual que siempre, tanto para escribir como para navegar. No obstante, debe tener en cuenta que:

- La tecla **Tabulador** mueve el cursor hacia adelante, deteniéndose antes y después de los nodos y resaltando su contenido. Sin embargo, el cursor no se detiene en contenido estático.
- Los hipervínculos `agregar...` y `agregar Nodo` se consideran nodos de contenido y se resaltan al llegar a ellos con la tecla **Tabulador**. Para activar estos nodos, pulse la barra espaciadora o la tecla **Entrar**.

© 2016-2022 Altova GmbH

11.4 Scripting de Authentic

La característica **Scripting de Authentic** ofrece una mayor flexibilidad e interacción en los diseños SPS. Estos diseños se pueden crear o editar en las ediciones StyleVision Enterprise y Professional y se pueden ver en la vista Authentic de las ediciones Enterprise y Professional de los productos de Altova.

A continuación aparecen los productos de Altova que ofrecen una vista Authentic y que son compatibles con los scripts de Authentic. No obstante, tenga en cuenta que en la versión de confianza del complemento Authentic Browser para exploradores la característica de scripting interna está desactivada por motivos de seguridad.

Producto	¿Creación de scripts de Authentic?	¿Scripts de Authentic habilitados?
StyleVision Enterprise	Sí	Sí
StyleVision Professional	Sí	Sí
StyleVision Standard *	No	No
XMLSpy Enterprise	No	Sí
XMLSpy Professional	No	Sí
XMLSpy Standard	No	No
AuthenticDesktop Enterprise	No	Sí
Authentic Browser Plug-in Enterprise Trusted **	No	Sí
Authentic Browser Plug-in Enterprise Untrusted	No	Sí

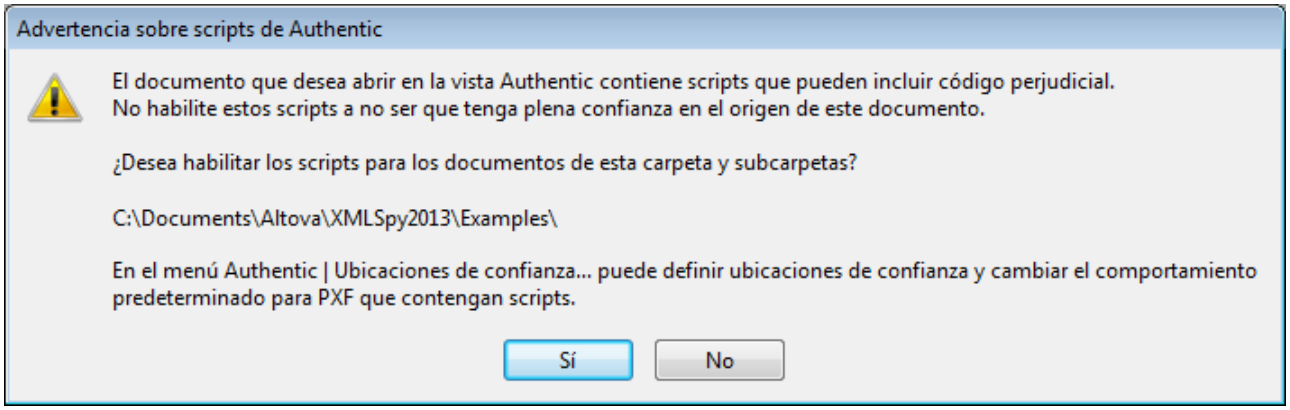
* *No incluye la vista Authentic.*

** *La vista puede mostrar diseños con scripts. No ofrece ejecución interna de macros ni control de eventos. Los eventos externos se disparan.*

Los scripts de Authentic se comportan del mismo modo en todos los productos de Altova, por lo que no necesita configurar cada producto por separado ni añadir código adicional.

Cuadro de diálogo de advertencia sobre scripts

Si un archivo PXF (o un archivo XML vinculado a una SPS) contiene un script y el archivo se abre o se visualiza en la vista Authentic, aparece este cuadro de diálogo:



Tiene dos opciones:

- Haga clic en **Sí**para añadir la carpeta que contiene el archivo a la lista de ubicaciones de confianza para scripts de Authentic. A partir de ahora los archivos de esa carpeta no emitirán advertencias cuando se abran en la vista Authentic. La lista de ubicaciones de confianza se puede abrir y modificar con el comando de menú **Authentic | Ubicaciones de confianza**[1421].
- Haga clic en No si prefiere no añadir la carpeta que contiene el archivo a la lista de ubicaciones de confianza. El archivo se abrirá en la vista Authentic pero tendrá los scripts deshabilitados. El cuadro de diálogo "Advertencia sobre scripts de Authentic" aparecerá cada vez que intente abrir el archivo en la vista Authentic. Si más adelante quiere añadir la carpeta del archivo a la lista de ubicaciones de confianza, haga clic en el comando de menú Authentic | Ubicaciones de confianza y añada la carpeta a la lista del cuadro de diálogo "Ubicaciones de confianza".

Para más información sobre el cuadro de diálogo "Ubicaciones de confianza" consulte la descripción del comando **Authentic | Ubicaciones de confianza**[1421] en la Referencia del usuario.

Nota: cuando se accede a StyleVision desde su interfaz COM (consulte la Referencia del programador[1639] para aprender a hacerlo), esta comprobación de seguridad no se hace y **la advertencia sobre scripts de Authentic no aparece**.

Esto puede hacerse de dos formas diferentes:

El usuario que diseña el diseño SPS puede usar la característica Scripting de Authentic de dos formas distintas para conseguir documentos de Authentic más interactivos:

- Asignando scripts para acciones definidas por el usuario (macros) a los elementos de diseño, botones de la barra de herramientas y comandos de los menús contextuales.
- Añadiendo agentes de control de eventos al diseño que reaccionen a los eventos de la vista Authentic.

Todos los scripts necesarios para convertir los documentos de Authentic en documentos interactivos se crean en StyleVision. Los formularios, las macros y los controladores de eventos se crean dentro del editor de scripts de StyleVision y los scripts se guardan con el archivo SPS. Más adelante, en la vista Diseño de StyleVision, el usuario asignará los scripts guardados a elementos de diseño, a botones de barras de herramientas y a menús contextuales. Cuando se abre un documento XML basado en el SPS en un producto de Altova compatible con scripts de Authentic (*ver tabla más arriba*), el documento ofrecerá las funciones interactivas que se crearon para él.

© 2016-2022 Altova GmbH

Documentación de Scripting de Authentic

La documentación de la función scripting de Authentic forma parte de la documentación de StyleVision. Puede consultar esta documentación en el sitio web de Altova, en la página de documentación y archivos de ayuda para productos de Altova.

12 HTML y CSS

XMLSpy ofrece funciones de edición inteligente para documentos HTML[657] y CSS[659]. Este tipo de documentos se pueden editar en la vista Texto[145] y se puede obtener una vista previa del documento HTML activo en la vista Explorador.

Los apartados de esta sección describen las funciones de edición inteligente para cada tipo de documento: HTML[657] y CSS[659].

© 2016-2022 Altova GmbH

12.1 HTML

Los documentos HTML se pueden editar en la vista Texto y después de editarse pueden verse inmediatamente en la vista Explorador. La vista Texto incluye varias funciones de edición HTML muy prácticas. Estas funciones se describen detalladamente en la sección Vista Texto[145], pero a continuación describimos las funciones de edición HTML más importantes y otras opciones relacionadas con HTML.

Nivel de compatibilidad

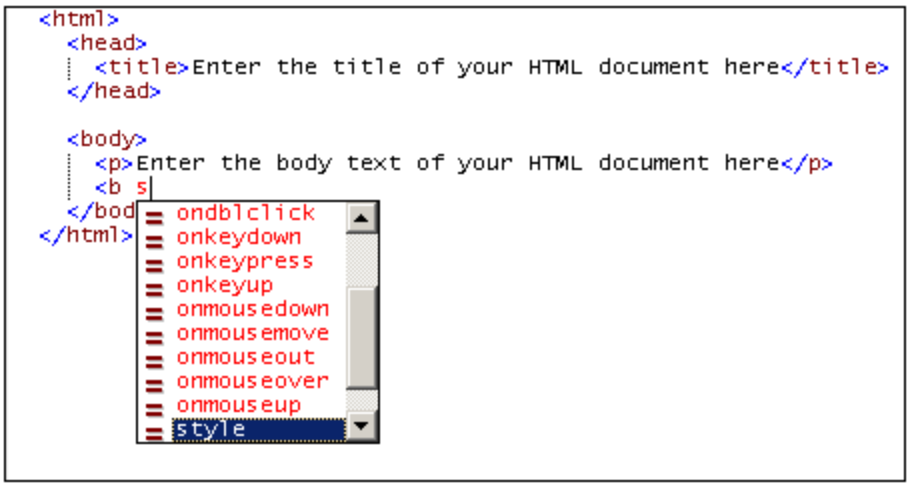
XMLSpy es compatible con HTML 4.0 y HTML 5.0. Las funciones de edición inteligente y los ayudantes de entrada son compatibles con estas versiones.

Ayudantes de entrada

XMLSpy ofrece tres ayudantes de entrada cuando el documento activo es un documento HTML: los ayudantes de entrada Elementos, Atributos y Entidades. Estos ayudantes de entrada son contextuales, es decir, las opciones que aparecen en ellos dependen de la posición del cursor. Para más información sobre cómo se utilizan los ayudantes de entrada HTML, consulte la sección Vista Texto[165].

Finalización automática

Mientras teclea el código HTML en su documento, XMLSpy ofrece opciones de finalización automática en una lista desplegable y emergente. Esta lista contiene todos los nodos que se pueden insertar en la posición actual del cursor. La lista aparece con una opción resaltada, que es la primera opción de la lista que más se ajusta al texto tecleado (*ver imagen siguiente*). Haga clic en una opción para insertarla en el punto de inserción del cursor.



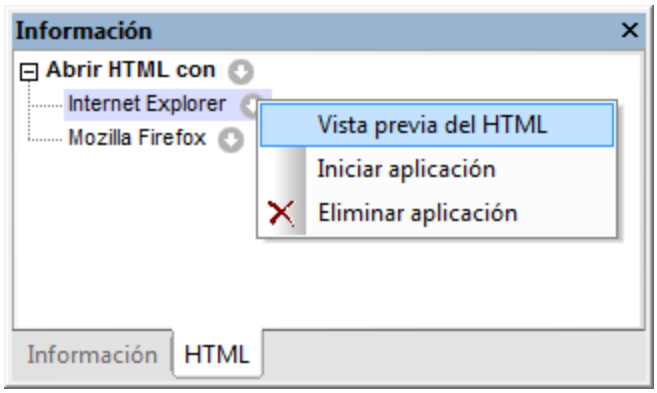
En el caso de los elementos HTML, la lista de finalización automática aparece al introducir el corchete inicial de la etiqueta. Al introducir la etiqueta inicial de un nodo elemento, la etiqueta final se inserta automáticamente. Esta práctica función le ayudará a crear documentos HTML con un formato correcto.

En el caso de los atributos HTML, la lista de finalización automática aparece al introducir un espacio después del nombre de elemento en una etiqueta inicial. Al hacer clic en un nombre de atributo de la lista de finalización automática, el atributo elegido se inserta entre comillas y el cursor se coloca entre las comillas.

El ayudante de entrada Entidades incluye entidades de caracteres de los conjuntos de entidades, Latin-1, caracteres especiales y símbolos de HTML 4.0 y HTML 5.0.

Ventana de información HTML

La ventana Información HTML (*ver imagen*) enumera las aplicaciones que se pueden usar para acceder al archivo HTML activo. Por ejemplo, si el documento activo en XMLSpy es un archivo HTML y hace doble clic en el elemento Mozilla Firefox de la ventana Información, entonces se inicia Mozilla Firefox y se carga el documento HTML en Firefox.



Otros aspectos de esta ventana a tener en cuenta son:

- El icono situado a la derecha del elemento *Abrir HTML con* sirve para agregar aplicaciones a la lista de aplicaciones. A esta lista puede agregar todos los exploradores instalados en el sistema o cualquier otra aplicación que desee utilizar (p. ej. un editor de texto). Para ello, haga clic en el icono situado a la derecha de *Abrir HTML con* y después seleccione el comando **Agregar aplicación nueva** o **Agregar exploradores instalados**.
- Después de agregar una aplicación a la lista (excepto si esta se agrega con el comando **Agregar exploradores instalados**), puede cambiar el nombre con el que aparece en la lista. Para ello, seleccione la aplicación, pulse **F2** y edite el nombre.
- Al hacer clic en los iconos situados a la derecha de cada aplicación de la lista se abre un menú que contiene comandos para (i) abrir la aplicación, (ii) abrir la aplicación y cargar el archivo HTML en ella y (iii) quitar la aplicación de la lista. Al hacer doble clic en el nombre de una aplicación el archivo HTML se abre en la aplicación.
- Las aplicaciones que se añaden o eliminan en esta lista también se añaden o eliminan en la ventana Información CSS.

Asignar una DTD

Si está trabajando con un documento XHTML, puede asignarle una DTD o un esquema XML desde el menú **DTD/Esquema**. Los documentos XHTML se editan exactamente igual que los documentos XML 290.

Comandos de la vista Explorador

Los comandos de la vista Explorador están disponibles en el menú **Explorador**.

© 2016-2022 Altova GmbH

12.2 CSS

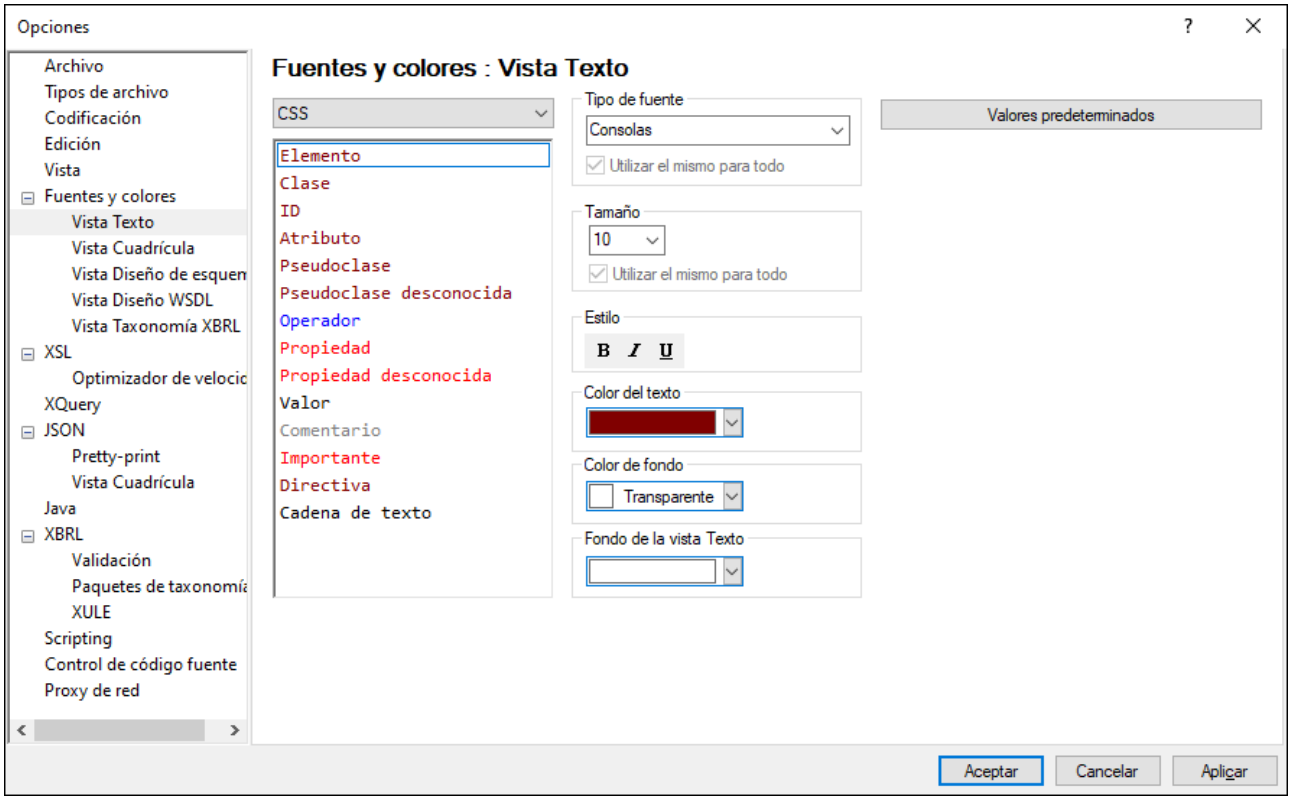
La Vista Texto ofrece varias funciones de edición para documentos CSS que describimos a continuación.

Color de sintaxis

Una regla CSS está formada por un selector, una o varias propiedades y los valores de estas propiedades. Estos tres componentes pueden ser de diferentes tipos: por ejemplo, un selector puede ser una clase, una pseudoclase, un ID, un elemento o un atributo. Además de reglas, los documentos CSS pueden incluir otros componentes, como comentarios por ejemplo. En la vista Texto cada una de estas categorías de elementos puede presentarse con un color diferente (*imagen siguiente*), dependiendo de la configuración elegida en el cuadro de diálogo Opciones (*imagen siguiente*).

```
.header
{
    font-family: "Arial", sans-serif;
    font-weight: bold;
    color: red;
}
```

Puede configurar los colores de los diferentes componentes CSS en la sección Fuentes y colores: Vista Texto del cuadro de diálogo "Opciones" (*imagen siguiente*). En esta pestaña, seleccione primero la opción CSS en el cuadro combinado situado en la parte superior izquierda.



Plegamiento de código

El plegamiento de código es una característica que permite expandir y contraer cada regla CSS. Las reglas CSS que se pueden expandir/contraer se marcan en el margen de plegamiento de código con el signo +/-. El margen puede activarse o desactivarse en el cuadro de diálogo "Configurar la vista Texto" 1489. Cuando una regla CSS está contraída, se marca con puntos suspensivos en el margen de la vista. Si pasamos el puntero por encima de los puntos suspensivos, aparece en pantalla un cuadro emergente con el contenido de la regla CSS. Si el contenido es demasiado largo para el cuadro emergente, esto se indica con puntos suspensivos también.

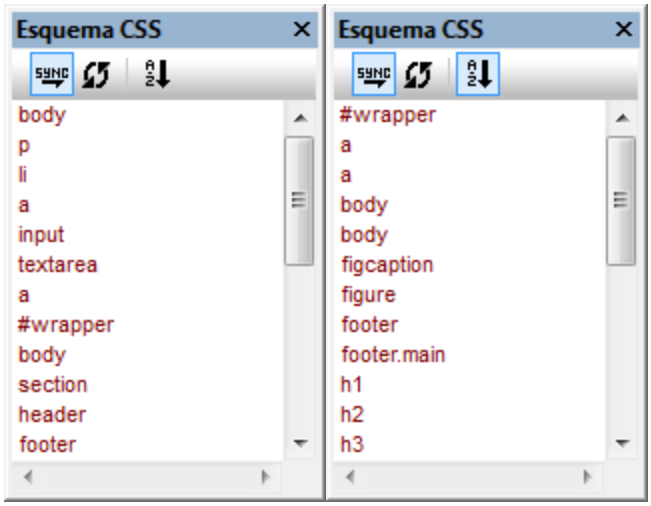
El icono **Expandir o contraer todos los plegamientos** de la barra de herramientas Texto expande **todas** las reglas CSS o las contrae hasta el elemento de nivel superior.

Nota: observe también que las llaves que delimitan la regla aparecen en negrita al colocar el cursor justo antes o después de una de las llaves. Esta práctica función le ayudará a encontrar el principio y el final de la definición de la regla. Esto sirve para señalar con claridad donde empieza y termina la definición del elemento.

Esquema CSS

El ayudante de entrada Esquema CSS (*imagen siguiente*) ofrece un resumen o esquema del documento, enumerando sus selectores. Al hacer clic en un selector del Esquema CSS el selector se resalta en el documento. De las dos imágenes que aparecen a continuación, la de la izquierda enumera todos los selectores en el mismo orden en que aparecen en el documento. En la de la derecha se activó la función Orden alfabético (con el botón de la barra de herramientas) y los selectores se ordenaron alfabéticamente.

Tenga en cuenta estos aspectos: Es importante tener en cuenta que (i) para ordenar los selectores alfabéticamente se utilizan todas las partes del selector, incluidos los puntos, las almohadillas y los dos puntos y que (ii) si el documento CSS contiene varios selectores que estén agrupados para definir una sola regla (p. ej. `h4, h5, h6 {...}`), entonces cada selector del grupo se enumera por separado.



Al hacer clic en un selector del Esquema CSS el selector se resalta en el documento.

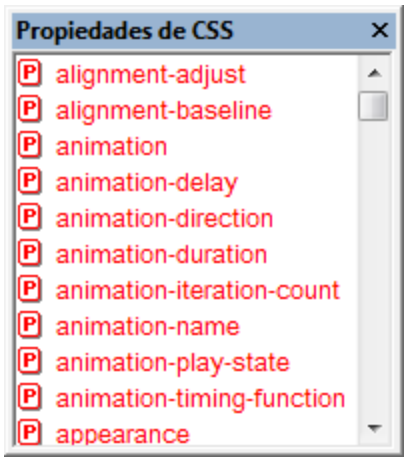
Sync	Activa o desactiva la sincronización automática (con el documento). Cuando se activa la sincronización automática, el ayudante de entrada se actualiza al mismo tiempo que el documento.

© 2016-2022 Altova GmbH

	Sincroniza el ayudante de entrada con el estado actual del documento.
	Activa o desactiva el orden alfabético. Cuando el orden alfabético está desactivado, los selectores se enumeran en el orden en el que aparecen en el documento. Cuando se activa el orden alfabético, los selectores ID aparecen primero porque van precedidos de una almohadilla (p.ej. `#intro`).

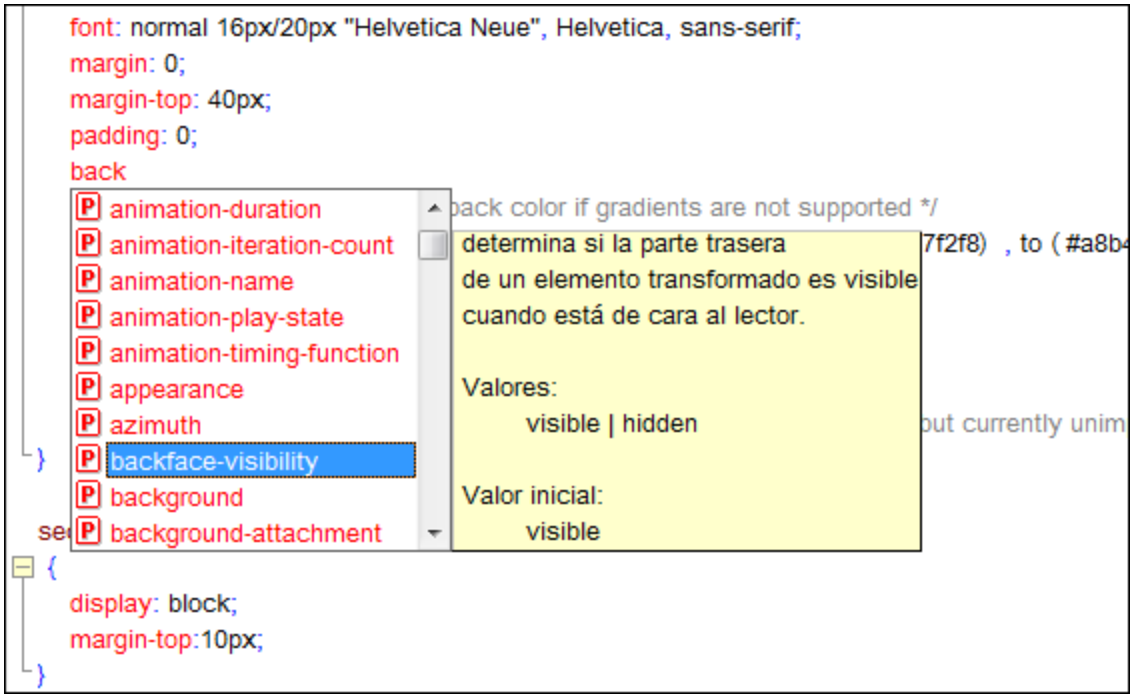
Ayudante de entrada Propiedades

El ayudante de entrada Propiedades (*imagen siguiente*) enumera todas las propiedades CSS por orden alfabético. Para insertar una propiedad en el punto de inserción del cursor, haga doble clic en la propiedad en este ayudante de entrada.



Finalización automática de propiedades e información rápida

Mientras teclea el nombre de una propiedad, XMLSpy muestra una lista emergente que enumera las propiedades que empiezan por las letras tecleadas (*ver imagen siguiente*). También puede abrir esta lista emergente con solo poner el cursor dentro de un nombre de propiedad y pulsar **Ctrl+Barra espaciadora**.



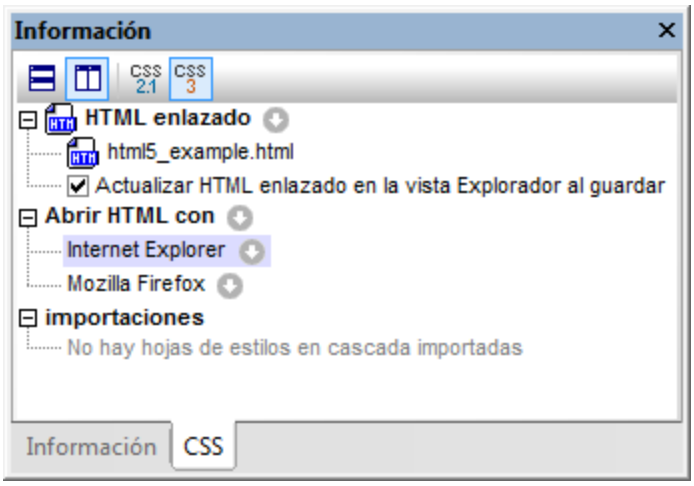
Además, XMLSpy muestra información rápida emergente para la propiedad que esté resaltada en la lista. La información rápida emergente incluye la definición de la propiedad CSS y los valores que puede tomar. Para insertar una propiedad de la lista, pulse **Entrar** o haga clic en ella.

Ventana Información CSS

La ventana Información CSS se habilita cuando el documento activo es un archivo CSS (*ver imagen siguiente*). La ventana Información CSS ofrece estas funciones:

- Botones para cambiar de CSS 2.1 a CSS 3.0. Los ayudantes de entrada y las funciones de edición inteligente cambian dependiendo de la versión de CSS seleccionada en la ventana Información.
- Opción para enlazar el archivo CSS a un archivo HTML. Esta función permite modificar el documento CSS y ver los resultados inmediatamente. Asimismo, el archivo HTML enlazado se puede abrir en varios exploradores web desde la ventana Información CSS. Así podrá ver el efecto de los cambios realizados en varios exploradores.
- La ventana Información CSS enumera las hojas de estilos CSS importadas, así podrá ver de un vistazo la estructura de la hoja de estilos CSS activa.

© 2016-2022 Altova GmbH



Otros aspectos de esta ventana que se deben tener en cuenta son:

- La barra de herramientas de la ventana Información contiene botones para CSS 2.1 y CSS 3.0. Los ayudantes de entrada y las funciones de edición inteligente cambian dependiendo de la versión de CSS seleccionada.
- Por cada documento CSS puede enlazar un solo archivo HTML. Para ello, haga clic en el icono situado junto al elemento *HTML enlazado*, seleccione el comando **Establecer enlace a HTML** y busque el archivo HTML que desea enlazar. El archivo HTML enlazado aparecerá en la lista *HTML enlazado* de la ventana Información (*imagen anterior*). Ni el documento CSS ni el archivo HTML cambian cuando se establece este enlace. El enlace sólo sirve para crear un archivo HTML temporal al que se aplica el documento CSS con el propósito de realizar pruebas.
- Al hacer doble clic en el archivo HTML enlazado este se abre en XMLSpy.
- La barra de herramientas también incluye botones para colocar el documento CSS y el archivo HTML en mosaico horizontal o vertical.
- Cuando se guardan cambios en el documento CSS, el archivo HTML que está abierto en XMLSpy se puede actualizar automáticamente Para habilitar esta actualización automática, marque la casilla *Actualizar HTML enlazado* en la vista Explorador al guardar. La actualización sólo se produce si el archivo HTML contiene una referencia al documento CSS que se está editando.
- Para cambiar el archivo HTML enlazado, seleccione el comando **Establecer enlace a HTML** y elija otro archivo.
- Para eliminar el enlace al archivo HTML, haga clic en el icono situado a la derecha de *HTML enlazado* y seleccione el comando **Eliminar enlace**.
- El icono situado a la derecha del elemento *Abrir HTML con* sirve para agregar aplicaciones a la lista de aplicaciones. A esta lista puede agregar todos los exploradores instalados en el sistema o cualquier otra aplicación que desee utilizar (p. ej. un editor de texto). Para ello, haga clic en este icono y después seleccione el comando **Agregar aplicación nueva** o **Agregar exploradores instalados**.
- Después de agregar una aplicación a la lista (excepto si esta se agrega con el comando **Agregar exploradores instalados**), puede cambiar el nombre con el que aparece en la lista. Para ello, seleccione la aplicación, pulse **F2** y edite el nombre.
- Al hacer clic en los iconos situados a la derecha de cada aplicación de la lista se abre un menú que contiene comandos para (i) abrir la aplicación, (ii) abrir la aplicación y cargar el archivo HTML en ella y (iii) quitar la aplicación de la lista. Al hacer doble clic en el nombre de una aplicación el archivo HTML se abre en la aplicación.
- Las aplicaciones que se añaden o eliminan en esta lista también se añaden o eliminan en la ventana Información HTML.
- La lista Importaciones enumera los archivos CSS importados en el documento CSS activo.

© 2016-2022 Altova GmbH

13 JSON y esquemas JSON

JSON (*notación de objetos JavaScript*) es un formato ligero de almacenamiento e intercambio de datos que utiliza la sintaxis JavaScript y, al igual que XML, presenta un formato de texto totalmente legible. Los textos JSON se pueden leer y utilizar en cualquier lenguaje de programación y, por tanto, se ha convertido en un formato de intercambio de datos muy utilizado, especialmente en Internet.

XMLSpy ofrece funciones de edición y validación de documentos de datos JSON (también llamados documentos de instancia)[668], así como funciones para crear documentos de esquema JSON[671] correctos desde un punto de vista sintáctico y gramatical.

XMLSpy también es compatible con Avro y Avro Schema[767].

JSON5

JSON5 es una extensión de JSON que incluye algunas extensiones ECMAScript5 (consulte json5.org para obtener más información). JSON5 es un subconjunto estricto de JavaScript, añade tipos de datos nuevos y funciona con todo el contenido JSON actual.

Todas las funciones de XMLSpy que son compatibles con documentos de instancias JSON también lo son con documentos de instancia JSON5. Sin embargo, debe tener en cuenta que existen algunas diferencias en el modo en que XMLSpy se ocupa de estos dos formatos:

- Los documentos de instancia JSON5 se pueden validar con esquemas JSON. JSON5 no es un sucesor oficial de JSON y, por tanto, utiliza su propia extensión de archivo (`json5`).
- En XMLSpy los archivos que tienen la extensión `.json` se consideran archivos de instancia JSON y los que tienen la extensión `.json5` se consideran archivos de instancia JSON5.
- Los documentos de instancia JSON5 se pueden validar con esquemas JSON. Los documentos de instancia JSON, que pueden ser representaciones de instancias Avro, pueden validarse con esquemas JSON y con esquemas Avro Los documentos de instancia JSON, que pueden ser representaciones de instancias Avro, pueden validarse con esquemas JSON y con esquemas Avro (véase *Validar documentos JSON*[754]).

En esta documentación el término *instancias JSON*hace referencia tanto a documentos de instancia JSON como JSON5 a no ser que se indique lo contrario. Para más información consulte el apartado Diferencias entre JSON5 y JSON[670].

JSON y JSON Schema XMLSpy

Estos dos tipos de documentos (instancias JSON y esquemas JSON) están escritos en formato JSON y deben cumplir las reglas JSON de formato correcto y validez. Ambos tipos de documentos suelen tener la extensión de archivo `.json`. Las instancias JSON se pueden editar en las vistas Texto[674] y Cuadrícula[679], mientras que los documentos de esquema JSON[714] se pueden editar en estas dos vistas y en la vista Esquema JSON, que es un editor gráfico de esquemas.

Estas son las características que ofrece XMLSpy para trabajar con instancias y esquemas JSON:

- La vista Texto[674] ofrece color de sintaxis, revisión sintáctica, finalización automática para esquemas JSON y para documentos de instancia si tienen asociaciones de esquema, función de plegamiento de código y marcado estructural. Todas estas características facilitan y aceleran las tareas de edición de

instancias JSON y esquemas JSON válidos. La vista Texto[674] también permite validar documentos de instancia y de esquema.
- La vista Cuadrícula[679] consiste en una estructura tabular que permite visualizar mejor la estructura de los documentos. Puede editar los documentos en esta vista directamente y también cambiar de la vista Texto[674] a la vista Cuadrícula[679] durante el proceso de edición. Además, la vista Cuadrícula[679] incluye una función para validar documentos de instancia y de esquema.
- La función de validación de documentos de instancia JSON está disponible en las vistas Texto[674] y Cuadrícula[679]. Para la validación se utiliza el esquema JSON que se asignó a la instancia en la ventana Información[754].
- La vista Esquema JSON[714] presenta los esquemas JSON de forma gráfica e incluye funciones gráficas y de edición que permiten crear esquemas JSON válidos con total facilidad La vista también incluye unos prácticos ayudantes de entrada. Además, la vista comprueba continuamente la validez del esquema y avisa de errores.

Instancias JSON: abrir documentos de instancias y crear documentos de instancia nuevos

- En la sección **Tipos de archivo**[1587] del cuadro de diálogo "Opciones" (Herramientas | Opciones) puede elegir la vista predeterminada para documentos de instancia JSON/JSON5 (Texto[674] o Cuadrícula[679]). Los documentos JSON/JSON5 se abrirán en la vista inicial predeterminada que seleccione aquí. Recuerde que puede cambiar de vista en todo momento.
- Para crear un documento de instancia JSON o JSON5 nuevo haga clic en **Archivo | Nuevo** y seleccione json: `notación de objetos JavaScript` o `json5: JSON with ECMAScript 5 extensions` respectivamente. La aplicación le ofrece la opción de elegir un archivo de esquema JSON o Avro[767] (para JSON, no para JSON5) para el archivo de instancia nuevo. Si selecciona un esquema, la asignación se introducirá en la ventana Información[754]. El documento de instancia nuevo se abrirá en la vista Texto[674] o Cuadrícula[679], dependiendo de la configuración elegida en las secciones **Tipos de archivo**[1587] y **Archivo**[1587] del cuadro de diálogo "Opciones".

Esquemas JSON: abrir esquemas y crear esquemas nuevos

- Los documentos de esquema JSON se abren siempre en la vista Esquema JSON[714]. Recuerde que puede cambiar a la vista Texto[674] o Cuadrícula[679] en todo momento.
- Para crear un documento de instancia JSON o JSON5 nuevo haga clic en **Archivo | Nuevo** y seleccione `json: Esquema JSON` El documento de esquema JSON nuevo se abrirá en la vista Esquema JSON[714] y comenzará con la palabra clave `$schema`. Recuerde que puede cambiar a la vista Texto[674] o Cuadrícula[679] en todo momento.

Los apartados de esta sección describen en detalle las funciones de estas vistas.

Contenido de esta sección

Estos son los apartados de esta sección:

- Datos JSON[668]: información básica sobre los documentos JSON
- Esquemas JSON[671]: descripción de los esquemas JSON y su funcionamiento
- Líneas JSON y Comentarios JSON[673] ofrece información sobre dos especificaciones JSON más compatibles con XMLSpy
- Documentos JSON en la vista Texto:[674] explica cómo se trabaja con las funciones JSON de la vista Texto
- Documentos JSON en la vista Cuadrícula[679]: describe cómo se editan documentos JSON en la vista Cuadrícula

© 2016-2022 Altova GmbH

- Vista Esquema JSON 714: explica las funciones de edición de esquemas JSON de esta vista y describe cómo trabajar con proyectos JSON
- Validar documentos y datos JSON 754: explica cómo se asigna un esquema JSON a un documento JSON y cómo se validan documentos JSON
- Inserting JSON Fragments 756 explica cómo insertar rápidamente fragmentos JSON en un documento JSON desde fuentes externas
- Transformaciones JSON con XSLT/XQuery 758: explica cómo realizar consultas con XPath/XQuery 3.1 en documentos JSON
- Expresiones XQuery para JSON 760 ofrece una introducción sobre el uso de XQuery en documentos JSON
- Generar esquemas JSON a partir de instancias JSON 762: describe la función de generación de esquemas JSON
- Generar instancias JSON a partir de esquemas JSON 765: describe cómo generar una instancia a partir de un esquema
- Convertir datos JSON en XML y viceversa 766: describe cómo convertir datos JSON en XML y viceversa en XMLSpy

13.1 Datos JSON

En este apartado describimos brevemente cómo se estructuran lo datos JSON. Este tipo de datos suelen almacenarse en documentos de instancia JSON, pero también se pueden almacenar en un fragmento de datos JSON incluido en un documento de otro tipo. Un fragmento o un documento de datos JSON es una estructura de datos JSON como la que se describe a continuación.

XMLSpy también es compatible con **JSON5**, una extensión de JSON que añade algunas extensiones ECMScript 5. Consulte json5.org para obtener más información.

Objetos y matrices JSON

Un documento JSON (cuya extensión de archivo suele ser **.json**) está compuesto por estas estructuras de datos centrales:

Objeto

Un objeto se delimita por medio de llaves y es una colección desordenada de cero o más pares **clave:valor** o más. Estos pares **clave:valor** conforman las **propiedades del objeto**. La clave debe ser siempre una cadena y, por tanto, siempre debe ir entre comillas. La clave (que también se conoce como *nombre de la propiedad*) se separa de su valor por medio de dos puntos. El valor de una propiedad puede ser de cualquier tipo de datos JSON (*ver lista más abajo* 669). Las propiedades se separan por medio de comas. A continuación puede ver un ejemplo de un objeto con tres propiedades (todas con valores de tipo atómico):

```
{
  "emailtype": "home",
  "emailaddress": "contact01.home@altova.com",
  "citycode": 22
}
```

Matriz

Una **matriz** se delimita por medio de corchetes y es una lista ordenada y separada por comas de cero o más **ítems**. Estos ítems pueden ser de cualquier tipo de datos JSON (*ver lista más abajo* 669).

Ejemplo de una matriz compuesta por dos objetos

La matriz que aparece a continuación contiene dos objetos (cada uno de ellos aparece entre llaves). La matriz propiamente dicha aparece entre corchetes.

```
[
      {
      "emailtype": "home",
      "emailaddress": "contact01.office@altova.com",
      "citycode": 22
      },
      {
      "emailtype": "office",
      "emailaddress": "contact01.office@altova.com",
      "citycode": 22
      }
  ]
```


© 2016-2022 Altova GmbH

- *Ejemplos de matrices que son valores de las propiedades de un objeto*

  El fragmento que aparece a continuación pertenece a un objeto formado por tres pares `clave:valor`. Cada valor es una matriz que contiene una tupla (secuencia). Las tuplas pueden entenderse como matrices de una sola dimensión. Los tres ítems de la tupla son de tipo atómico.

```
{
      "x": [ 1, 2, "abc" ],
      "y": [ 3, 4, "def" ],
      "z": [ 5, 6, "ghi" ]
}
```

Tipos de datos JSON

Los ítems de las matrices y los valores de las propiedades de los objetos pueden tener estos tipos:

- `string` (cadena, que debe ir entre comillas). Además las cadenas pueden tener un formato concreto, como el formato `date-time` o `email`
- `number`: un número con una parte fraccionaria. Incluye números enteros
- `integer`: un número sin parte fraccionaria. Se trata de un subconjunto del tipo `number`
- `boolean` (`true/false`, no debe ir entre comillas)
- `object`: si se utiliza dentro de otro objeto, este tipo permite anidar datos
- `array`: permite generar estructuras más complejas de las que permiten los objetos
- `null` (`null`, no debe ir entre comillas)

Ejemplo de datos JSON

A continuación puede ver un ejemplo de un fragmento de datos JSON. Observe que el documento está compuesto por objetos y matrices. Fíjese también en el tipo de datos de los valores de las claves: observe que los valores de las cadenas aparecen entre comillas y los demás tipos aparecen en verde.

```
{
  "first": "Jason",
  "last": "Jones"
  "isManager": true,
  "age": 35,
  "address": {
    "street": "Jason Avenue",
    "city": "Jasonville",
    "state": "JS",
    "postcode": "JS12 ON34"
  },
  "phone": [
    {
      "type": "home",
      "number": "12 3456-7890"
    },
    {
      "type": "office",
      "number": "789 012-34567"
```

```
    }
  ],
  "children": [],
  "partner": null
}
```

Diferencias entre JSON5 y JSON

JSON5 es un subconjunto estricto de JavaScript, añade tipos de datos JSON nuevos y funciona con todo el contenido JSON actual. Sin embargo, existen algunas diferencias entre estos dos formatos:

- JSON admite comentarios, que se delimitan así: `// comentario //` o `/* comentario */`.
- En JSON5 las claves de los pares `clave:valor` no tienen que estar entre comillas.
- En JSON5 se pueden escribir cadenas que ocupen varias líneas.
- Los documentos JSON5 se pueden validar con esquemas JSON pero no con esquemas Avro.

© 2016-2022 Altova GmbH

13.2 Esquemas JSON

Al igual que los esquemas XML especifican la estructura y el contenido de un documento XML, los esquemas JSON especifican cómo se organizan los datos JSON en un documento JSON. Es decir, especifica qué campos de datos deben aparecer y cómo se representan los valores. Aquí puede consultar la especificación JSON Schema y obtener más información sobre la misma.

Un esquema JSON es un objeto JSON en sí mismo. El esquema entero está entre llaves, que son lo delimitadores de los objetos JSON. Los esquemas están escritos en sintaxis JSON y se suelen guardar con la extensión de archivo `.json`. Además, se sabe que es un esquema JSON porque la primera palabra clave del objeto de nivel superior del documento es **`$schema`**. Esta palabra clave debe tener uno de estos valores:

- En las versiones hasta el borrador draft-07: **`"http://json-schema.org/draft-N/(hyper-)schema#"`**, donde **`N`** es el número de la versión (**`04`**, **`06`** o **`07`**).
- Versions from draft-2019-09 onwards: **`"https://json-schema.org/draft/YYYY-MM/(hyper-)schema"`**, donde **`YYYY`** y **`MM`** son, respectivamente, el año y el mes del borrador, por ejemplo, **`2019-09`**.

Este es un ejemplo de cómo se usa la palabra clave `$schema`.

```
{
   "$schema": "https://json-schema.org/draft/2020-12/schema",
   ...
}
```

Nota: aunque la palabra clave `$schema` puede tener el valor **`"http://json-schema.org/schema#"`**, que especifica la última versión del esquema, es mejor utilizar una URL que identifique la versión concreta. Para más información consulte también el apartado Versión del esquema JSON 715.

En XMLSpy puede crear esquemas JSON de forma gráfica en la vista Esquema JSON, tal y como explicamos en la sección Vista Esquema JSON 714 de este manual Además de múltiples funciones de edición, la vista Esquema JSON ofrece estas funciones para trabajar con esquemas:

- Validación con el validador JSON de XMLSpy: asigne un esquema JSON al documento de instancia y compruebe su validez desde XMLSpy. Consulte el apartado Validar documentos JSON 754 para obtener más información.
- Configurar las opciones de validación JSON 1595.
- Generar esquemas JSON a partir de instancias JSON 762: si ya tiene un documento de instancia JSON, puede usarlo para generar un esquema JSON.
- Conversión de datos JSON en XML y viceversa 766: puede convertir documentos de un formato al otro.

Terminología

A continuación puede consultar la definición de los términos de esquemas JSON más utilizados en la interfaz gráfica y en este manual.

Término	Definición
Esquema	Objeto de esquema de nivel superior del documento de esquema JSON.
Objeto	Un tipo JSON que contiene cero o más propiedades.

Propiedad	Un par `clave:valor` 668 de un objeto. Su valor puede ser cualquier tipo de datos JSON.
Palabra clave	La parte `clave` del par `clave:valor` 668 de un objeto. Siempre es una cadena de texto.
Subesquema	Un objeto secundario de un operador o una dependencia.
Definición	Descripción completa de cualquier tipo JSON. Las definiciones pueden ser globales o locales 721.
Matriz	Una lista ordenada y separada por comas de cero o más `ítems` de cualquier tipo de datos JSON.
Tipos atómicos	Los tipos de datos JSON `string`, `number`, `integer`, `boolean` y `null`
Selectores de tipo	Los tipos `any` y `multiple`, que seleccionan cualquier tipo o varios tipos 740 respectivamente.
Operadores	Selectores de repetición que se pueden añadir como secundarios de las definiciones. Consulte el apartado *Operadores* 746 para obtener más información.

Tipos de datos JSON

Los ítems de las matrices y los valores de las propiedades de los objetos pueden tener estos tipos:

- `string` (cadena, que debe ir entre comillas). Además las cadenas pueden tener un formato concreto, como el formato `date-time` o `email`
- `number`: un número con una parte fraccionaria. Incluye números enteros
- `integer`:: un número sin parte fraccionaria. Se trata de un subconjunto del tipo `number`.
- `boolean` (`true/false`, no debe ir entre comillas)
- `object`: si se utiliza dentro de otro objeto, este tipo permite anidar datos
- `array`: array: permite generar estructuras más complejas de las que permiten los objetos
- `null` (`null`, no debe ir entre comillas)

© 2016-2022 Altova GmbH

13.3 Líneas JSON y JSON con comentarios

XMLSpy admite los formatos Líneas JSON (JSONL) y JSON con comentarios (JSONC), lo que significa que tanto la validación como la edición inteligente se pueden aplicar a este tipo de documentos en la misma medida que se puede aplicar a otro tipo de documentos JSON. Esta sección contiene las características clave de este tipo de documentos.

Líneas JSON

El formato Líneas JSON (JSONL) permite almacenar datos estructurados en los que cada registro está separado del siguiente por una nueva línea; es decir, cada registro ocupa su propia línea. Como resultado, cada uno de los registros se puede procesar por separado, lo que hace que sea un formato muy útil a la hora de procesar datos como archivos de registro.

Ejemplo de documento de Líneas JSON

```
["Team", "Played", "Won", "Drew", "Lost", "Points"]
["USA", 2, 1, 1, 0, 4]
["France", 3, 1, 1, 1, 4]
["Germany", 1, 0, 1, 0, 1]
["USA", 1, 0, 0, 1, 0]
```

Para ver más ejemplos consulte http://jsonlines.org/examples/.

XMLSpy reconoce a los archivos en formato Líneas JSON como tales si estos tienen la extensión de archivo **`.jsonl`**.

Comentarios JSON

Los documentos que no sean JSON5 no admiten comentarios. El formato JSON con comentarios (JSONC) se introdujo para permitir escribir comentarios en documentos JSON. XMLSpy admite los siguientes comentarios en JSONC:

- Comentarios de una única línea: con prefijo `//`. Por ejemplo: **`// My comment`**
- Comentarios de varias líneas: delimitados por `/*` y `*/`. Por ejemplo: **`/* My comment */`**

XMLSpy reconoce a los archivos JSONC como tales si estos tienen la extensión de archivo **`.jsonc`**.

Nota: los comentarios también se permiten en los documentos JSON5. Los comentarios en otros tipos de archivos JSON (aparte de JSONC y JSON5) podrían provocar errores durante el procesamiento.

13.4 Documentos JSON en la vista Texto

Sitio web de Altova: Editor JSON

Los esquemas JSON, esquemas Avro[769] y documentos de instancia JSON/JSON5 (incluidas las instancias de datos Avro en formato JSON) se pueden editar con las funciones de edición inteligentes de la vista Texto. Entre estas funciones se encuentran los márgenes de plegamiento de código[674], el marcado estructural[674], el color de sintaxis[675], la revisión sintáctica[676] y la finalización automática[676]. XMLSpy también incluye una función de conversión de datos JSON/JSON5 en XML y viceversa[766] y una función para generar esquemas JSON a partir de instancias JSON/JSON5[762].

Márgenes de plegamiento de código

El plegamiento de código se habilita en palabras clave y definiciones JSON y sirve para expandir y contraer estos nodos, que se marcan con el signo +/- (*imagen siguiente*). El margen de plegamiento de código puede activarse/desactivarse en el cuadro de diálogo Configurar la vista Texto[1489]. Cuando un nodo está contraído, aparecen puntos suspensivos junto al nodo. Si pasamos el puntero por encima de los puntos suspensivos, aparece un recuadro con el contenido del nodo contraído. Si el contenido no cabe en el recuadro, aparecen puntos suspensivos al final del texto.

```
10  "OrgChart": {
11    "xmlns": "http://www.xmlspy.com/schemas/orgchart",
12    "xmlns:ipo": "http://www.altova.com/IPO",
13    "xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
14    "xmlns:ts": "http://www.xmlspy.com/schemas/textstate",
15    "xsi:schemaLocation": "http://www.xmlspy.com/schemas/orgchart OrgChart.xsd",
16    "CompanyLogo": {
17      "href": "nanonull.gif"
18    },
```

El icono **Expandir o contraer todos los plegamientos** de la barra de herramientas Texto sirve para expandir o contraer todos los nodos del documento.

Estas son las opciones que ofrece el icono +/- que aparece en el margen de plegamiento:

Clic [-]	Contrae el nodo.
Clic [+]	Expande el nodo de modo que sus descendientes aparezcan expandidos o contraídos, dependiendo de cual fuera su estado antes de que se contrajera el nodo.
Mayús+Clic [-]	Contrae todos los nodos descendientes, pero deja expandido el nodo en el que se hizo clic.
Ctrl+Clic [+]	Expande el nodo en el que se hizo clic y todos sus nodos descendientes.

Marcado estructural

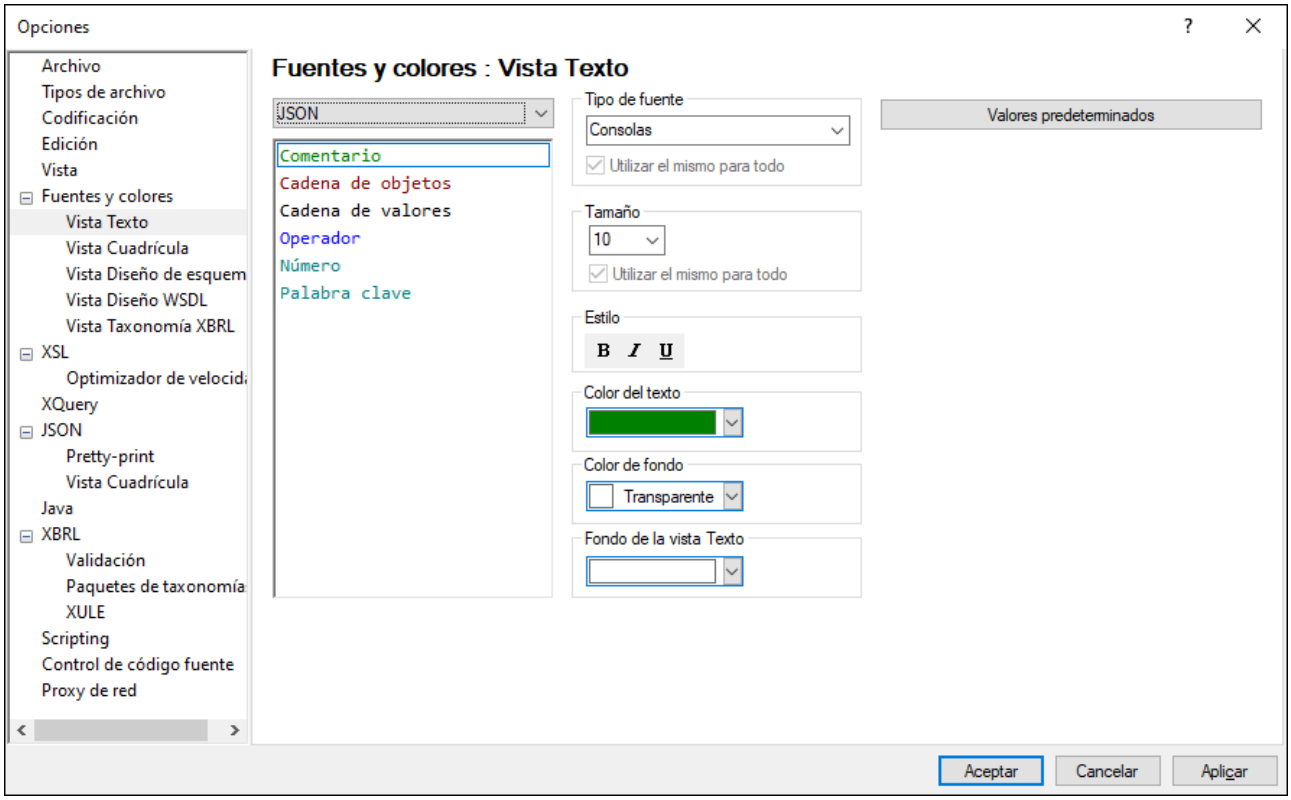
El par de llaves o corchetes que delimitan los objetos y las matrices JSON se marca en negrita (*imagen siguiente*) cuando colocamos el cursor antes o después de una llave o de un corchete. Esto sirve para señalar con claridad donde empieza y termina la definición del elemento.

© 2016-2022 Altova GmbH

```
"Department": [ { "Name": "Administration",
    "Person": [ { "First": "Vernon",
        "Last": "Callaby",
        "Title": "Office Manager",
        "PhoneExt": 582,
        "EMail": "v.callaby@nanonull.com",
        "Shares": 1500,
        "LeaveTotal": 25,
        "LeaveUsed": 4,
        "LeaveLeft": 21},
```

Color de sintaxis

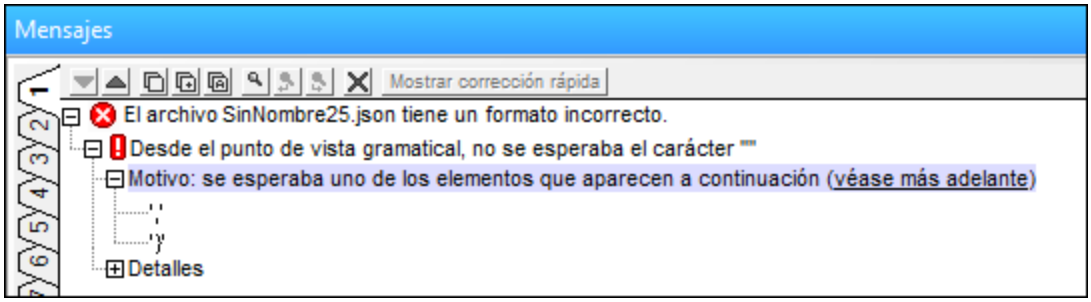
Los documentos JSON (esquemas o instancias JSON o Avro[767]), así como los documentos JSON5, están compuestos por cadenas de objetos, cadenas de valor, operadores, números y palabras clave. En la vista Texto cada una de estas categorías de elementos puede presentarse con un color diferente (*imagen siguiente*), dependiendo de la configuración elegida en el cuadro de diálogo Opciones[1583] (*imagen siguiente*). La configuración del color de sintaxis se puede modificar en la sección *Fuentes y colores: Vista Texto* de este cuadro de diálogo. Basta con seleccionar la opción **JSON** en el cuadro combinado situado en la esquina superior izquierda y seleccionar el color correspondiente para cada elemento JSON en el panel *Estilo*.



Nota: la sintaxis JSON5 (al contrario de la sintaxis JSON) admite el uso de comentarios. En JSON5 los comentarios se delimitan así: `// comentario //` o `/* comentario */`.

Color de sintaxis

La sintaxis de los documentos JSON (instancias o esquemas JSON o Avro[767]) se puede revisar con el comando **XML | Comprobar formato XML (F7)**. El resultado de la revisión aparece en la ventana Mensajes (*imagen siguiente*).



En el mensaje de error podemos ver que se utilizó un espacio en una posición del documento donde se esperaba una coma o una llave.

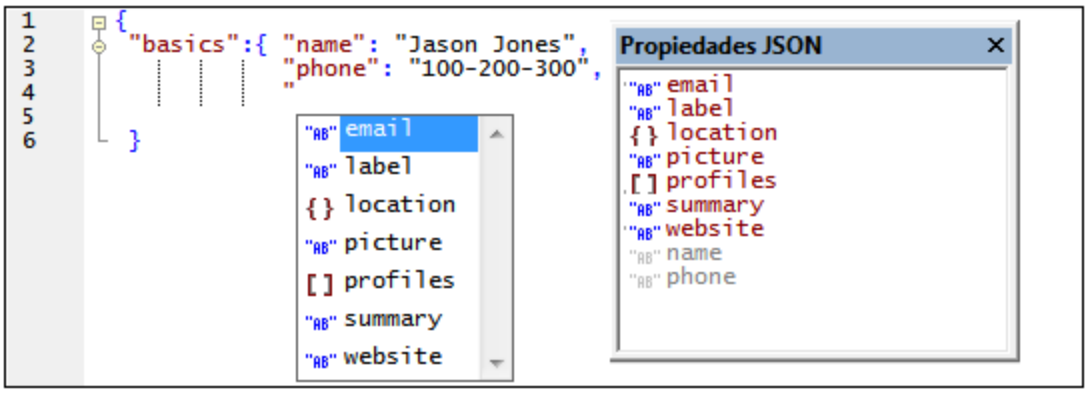
Finalización automática

La finalización automática se habilita si el documento JSON (instancia o esquema JSON o Avro) que estamos editando tiene asociado un esquema.

- Si el documento es un esquema JSON, entonces la finalización automática se basa en la versión del esquema indicada por la palabra clave `$schema`[671]. Para más información consulte también el apartado Versión del esquema JSON[715].
- Si el documento es una instancia JSON/JSON5, debe tener asociado un esquema JSON[754] para que la finalización automática funcione.
- Si el documento es un documento de datos Avro en formato JSON, entonces debe tener asociado un esquema Avro[754] para que la finalización automática funcione.
- Si el documento es un esquema Avro[769], entonces se asocia automáticamente con el esquema para Avro Schema y la finalización automática se basa en dicho esquema.

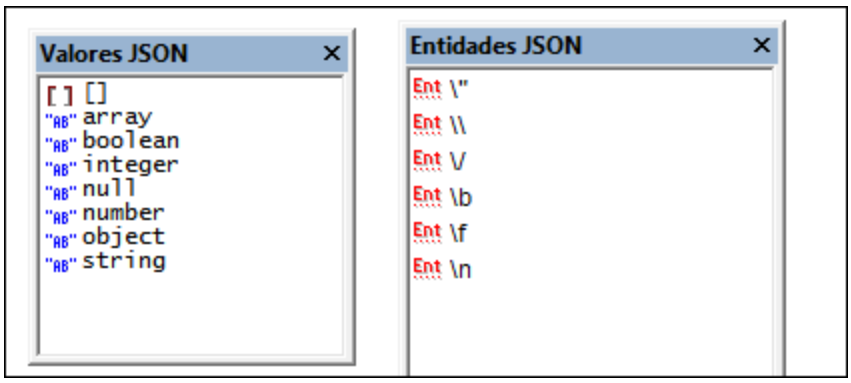
La finalización automática sugiere opciones de entrada correctas en la posición del cursor. Estas opciones aparecen en (i) recuadros emergentes en la ventana principal y (ii) en los ayudantes de entrada (*imagen siguiente*). Tanto los recuadros emergentes como los ayudantes de entrada incluyen una lista con todas las opciones de entrada que son válidas en la posición donde está el cursor. Para navegar por la lista de opciones basta con usar las teclas de dirección. Si el esquema contiene una descripción de una entrada (en la palabra clave `description` de la entrada), ésta aparece junto a la entrada resaltada. Para insertar una entrada basta con seleccionarla en el recuadro emergente o hacer doble clic en ella en el ayudante de entrada.

© 2016-2022 Altova GmbH



Por ejemplo, la imagen anterior muestra la edición de un documento de instancia. El recuadro emergente y el ayudante de entrada Propiedades JSON aparecen cuando colocamos el cursor después de las comillas que indican el inicio del nombre de una propiedad. El ayudante de entrada muestra todas las propiedades que se permiten en esa posición. Las propiedades que ya se han utilizado aparecen en gris y están deshabilitadas. El recuadro emergente muestra solamente las propiedades que se permiten en esa posición.

La vista ofrece dos ayudantes de entrada más: Valores JSON y Entidades JSON (*imagen siguiente*), que muestran respectivamente los valores permitidos de los pares `clave:valor` y las entidades necesarias para escapar los caracteres de las cadenas JSON. Por ejemplo, en la imagen siguiente, el ayudante de entrada Valores JSON muestra los valores permitidos para la palabra clave `type`. La última entrada del ayudante Entidades JSON, `\u00FF`, es un marcador de posición que representa un carácter Unicode. Basta con reemplazar la parte señalada en azul con el código del carácter Unicode que desee utilizar.



Además, si el esquema especifica enumeraciones, descripciones, número de repeticiones obligatorias y valores predeterminados, estos elementos también serán sugeridos por la finalización automática.

Guardar una cadena cifrada en base 64 como imagen

Para guardar una cadena cifrada en base 64 en formato imagen haga clic con el botón derecho del ratón en el tecto cifrado y seleccione el comando **Guardar como imagen**. En el cuadro de diálogo que aparece, seleccione la ubicación donde quiere guardar la imagen y asigne un nombre al archivo de imagen. La extensión del archivo (`.png`, `.gif`, `.svg`, etc.) se detecta automáticamente en el cifrado en base 64 y aparece en el cuadro de diálogo **Guardar**. Haga clic en Guardar para terminar.

También puede ejecutar esta acción con el comando de menú **Edición | Guardar como imagen**.

© 2016-2022 Altova GmbH

13.5 Vista Cuadrícula JSON

La vista Cuadrícula JSON sirve para ver la estructura del documento JSON (instancia o esquema JSON o Avro[769]) en forma de tabla, lo que permite editar el documento más fácilmente, y modificar la estructura de los bloques. Es decir, ofrece una perspectiva del documento y unas funciones de edición que no ofrece la vista Texto y que son especialmente prácticas a la hora de trabajar con documentos de gran tamaño y complejidad. Por ejemplo, los documentos JSON pueden estar formados por matrices y objetos anidados en otros objetos y matrices en múltiples niveles. A continuación puede ver código JSON en la vista Texto y su representación en la vista Cuadrícula.

Nota: la única edición compatible con **Avro** es la edición **Enterprise Edition**.

Fragmento de código JSON en la vista Texto

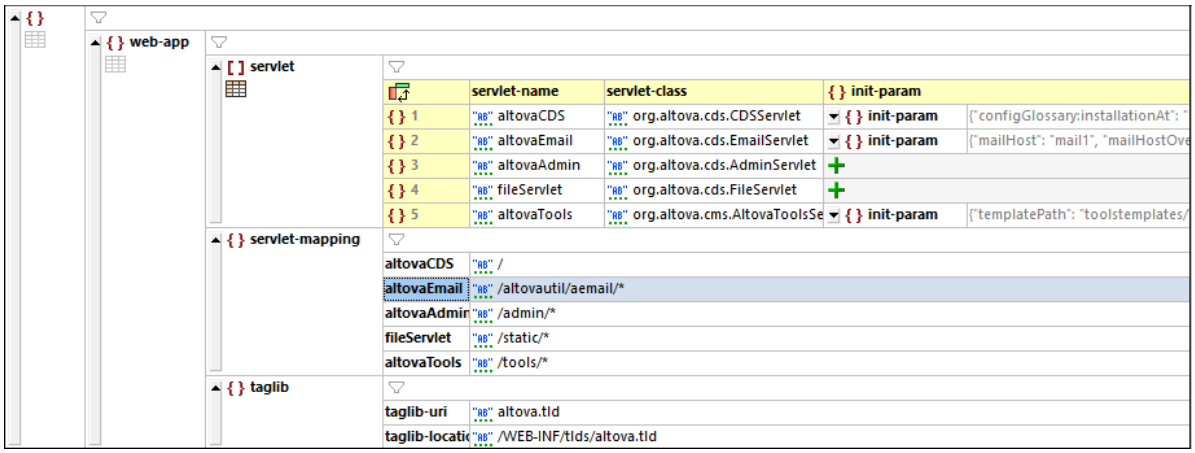

```
{
  "web-app": {
  "servlet": [
    {
        "servlet-name": "altovaCDS",
        "servlet-class": "org.altova.cds.CDSServlet",
        "init-param": {
        "configGlossary:installationAt": "Philadelphia, PA",
        "configGlossary:adminEmail": "ksm@pobox.com",
        "configGlossary:poweredBy": "Altova",
        "configGlossary:poweredByIcon": "/images/altova.gif",
        "configGlossary:staticPath": "/content/static",
        "templateProcessorClass": "org.altova.WysiwygTemplate",
        "templateLoaderClass": "org.altova.FilesTemplateLoader",
        "templatePath": "templates",
        "templateOverridePath": "",
        "defaultListTemplate": "listTemplate.htm",
        "defaultFileTemplate": "articleTemplate.htm",
        "useJSP": false,
        "jspListTemplate": "listTemplate.jsp",
        "jspFileTemplate": "articleTemplate.jsp",
        "cachePackageTagsTrack": 200,
        "cachePackageTagsStore": 200,
        "cachePackageTagsRefresh": 60,
        "cacheTemplatesTrack": 100,
        "cacheTemplatesStore": 50,
        "cacheTemplatesRefresh": 15,
        "cachePagesTrack": 200,
        "cachePagesStore": 100,
        "cachePagesRefresh": 10,
        "cachePagesDirtyRead": 10,
        "searchEngineListTemplate": "forSearchEnginesList.htm",
        "searchEngineFileTemplate": "forSearchEngines.htm",
        "searchEngineRobotsDb": "WEB-INF/robots.db",
        "useDataStore": true,
        "dataStoreClass": "org.altova.SqlDataStore",
        "redirectionClass": "org.altova.SqlRedirection",
        "dataStoreName": "altova",
        "dataStoreDriver": "com.microsoft.jdbc.sqlserver.SQLServerDriver",
```

```
      "dataStoreUrl": "jdbc:microsoft:sqlserver://LOCALHOST:1433;DatabaseName=goon",
      "dataStoreUser": "sa",
      "dataStorePassword": "dataStoreTestQuery",
      "dataStoreTestQuery": "SET NOCOUNT ON;select test='test';",
      "dataStoreLogFile": "/usr/local/tomcat/logs/datastore.log",
      "dataStoreInitConns": 10,
      "dataStoreMaxConns": 100,
      "dataStoreConnUsageLimit": 100,
      "dataStoreLogLevel": "debug",
      "maxUrlLength": 500
    }
    }, {
      "servlet-name": "altovaEmail",
      "servlet-class": "org.altova.cds.EmailServlet",
      "init-param": {
        "mailHost": "mail1",
        "mailHostOverride": "mail2"
      }
    }, {
      "servlet-name": "altovaAdmin",
      "servlet-class": "org.altova.cds.AdminServlet"
    }, {
      "servlet-name": "fileServlet",
      "servlet-class": "org.altova.cds.FileServlet"
    }, {
      "servlet-name": "altovaTools",
      "servlet-class": "org.altova.cms.AltovaToolsServlet",
      "init-param": {
        "templatePath": "toolstemplates/",
        "log": 1,
        "logLocation": "/usr/local/tomcat/logs/AltovaTools.log",
        "logMaxSize": "",
        "dataLog": 1,
        "dataLogLocation": "/usr/local/tomcat/logs/dataLog.log",
        "dataLogMaxSize": "",
        "removePageCache": "/content/admin/remove?cache=pages&id=",
        "removeTemplateCache": "/content/admin/remove?cache=templates&id=",
        "fileTransferFolder": "/usr/local/tomcat/webapps/content/fileTransferFolder",
        "lookInContext": 1,
        "adminGroupID": 4,
        "betaServer": true
      }
    }
  ],
  "servlet-mapping": {
    "altovaCDS": "/",
    "altovaEmail": "/altovautil/aemail/*",
    "altovaAdmin": "/admin/*",
    "fileServlet": "/static/*",
    "altovaTools": "/tools/*"
  },
  "taglib": {
```


 © 2016-2022 Altova GmbH

```
        "taglib-uri": "altova.tld",
        "taglib-location": "/WEB-INF/tlds/altova.tld"
      }
    }
  }
```

En la vista Texto es difícil comprender a primera vista la estructura del documento, mientras que en la vista Cuadrícula (imagen siguiente) podemos comprender la estructura de un solo vistazo.



En la vista Cuadrícula podemos modificar la estructura con solo añadir, eliminar o mover objetos por la tabla. Podemos reorganizar bloques enteros de texto (ordenándolos o moviéndolos, por ejemplo). Además, en esta vista también podemos editar el contenido del documento, lo que ahora es incluso más fácil gracias a los comandos dentro de celdas individuales [686].

Barra de herramientas de la vista Cuadrícula

Puede acceder a los comandos relacionados con la edición de la vista Cuadrícula desde la barra de herramientas de la vista Cuadrícula (imagen siguiente), que se encuentra en la parte superior de la ventana de la aplicación. Pase el cursor del ratón sobre los iconos para ver su nombre y sus teclas de acceso rápido.



Nota: estos comandos también están disponibles en el menú JSON [1351].

En esta sección

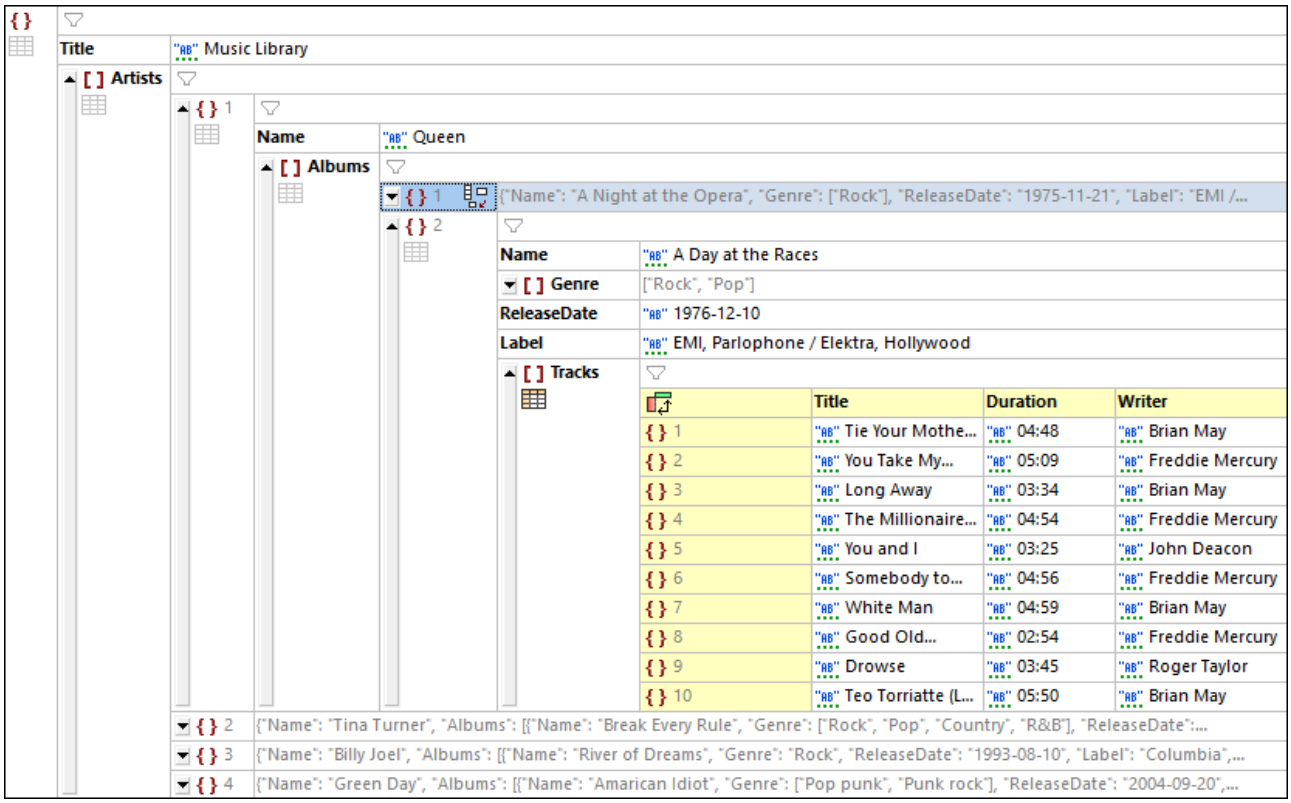
Esta sección contiene los apartados que enumeramos a continuación. En cada uno de ellos se explican características específicas de la vista Cuadrícula JSON. Para ver una descripción de otras características más generales de la vista Cuadrícula, como mover objetos en la cuadrícula, consulte la sección Vista Cuadrícula [169].

- Cómo se muestra un documento JSON en la vista Cuadrícula [682]
- Editar la estructura de un documento [685]

- Editar el contenido de un documento 686
- Arrastrar y colocar en la vista Cuadrícula 689
- Mostrar tablas 692
- Usar filtros XQuery y fórmulas 696
- Imágenes en la vista Cuadrícula 702
- Gráficos en la vista Cuadrícula 703
- Menú contextual en la vista Cuadrícula 706
- Configurar la vista Cuadrícula 709

13.5.1 Mostrar documentos en la Vista Cuadrícula

La Vista Cuadrícula JSON muestra el documento JSON de forma jerárquica dentro de una cuadrícula (*imagen siguiente*).



Cada línea muestra un tipo de estructura de datos. A continuación indicamos el símbolo que corresponde a cada una. Tenga en cuenta que tanto las matrices como los objetos pueden contener componentes secundarios que pueden ser objetos, matrices o valores atómicos.

{ }	Un objeto (*ver definición* 668). Los objetos contienen pares `key:value`.

© 2016-2022 Altova GmbH

[]	Una matriz (*ver definición*[668]). Las matrices contienen elementos que suelen ser objetos o valores.
	Un par `key:value` (*ver definición*[668]).

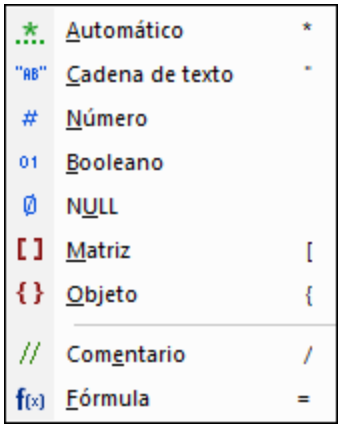
En la imagen anterior se puede ver un objeto raíz que contiene dos pares `key:value`: (i) **`"Title":"Music Library"`**, (ii) **`"Artists":[Array]`**. La matriz de **`Artists`** contiene cuatro elementos que son objetos.

Tenga en cuenta que:

- puede acercar y alejar la cuadrícula usando **Ctrl + rueda del ratón** o **Ctrl+[más]** y **Ctrl+[menos]**.
- los objetos, las matrices y los elementos de matriz de un mismo nivel están numeradas (empezando por el 1). En la imagen anterior, por ejemplo, dentro de la matriz `Artists` los objeto están numerados del `1` al `4`. Eso ocurre también con las matrices `Albums` y `Tracks`. Tenga en cuenta que esta numeración no está en el documento original, sino que se trata de una característica de la vista Cuadrícula JSON para ayudarle a ver la estructura del documento.
- los objetos y las matrices se pueden expandir o contraer pulsando el icono en forma de flecha que aparece a la izquierda del símbolo de la estructura (*imagen anterior*). Cuando se contrae una estructura de datos, su contenido se muestra como texto en una sola línea; un ejemplo son las tres últimas líneas de la cuadrícula de la imagen anterior.
- si selecciona varios componentes del mismo nivel puede expandirlos o contraerlos todos pulsando **Mayús** y haciendo clic en una de las cabezas de flecha de los componentes seleccionados.
- cuando se expanden matrices u objetos: (i) el contenido de las matrices y de los objetos empieza en la línea siguiente a su símbolo, (ii) el nombre de una matriz (su clave) está en la misma línea que el símbolo de la matriz, mientras que los elementos de la matriz empiezan en la línea siguiente. Sin embargo, cuando se contrae una matriz o un objeto, su contenido aparece en la misma línea.
- observe que los pares `key:value` están representados en una sola línea, excepto cuando son parte de la Vista Diseño de tabla[692] (en cuyo caso la clave será el encabezado de la columna). Para ver un ejemplo de la Vista Diseño de tabla[692] consulte las imágenes de la matriz **`Tracks`** en las imágenes anteriores.

Símbolos de tipos de datos (o tipos)

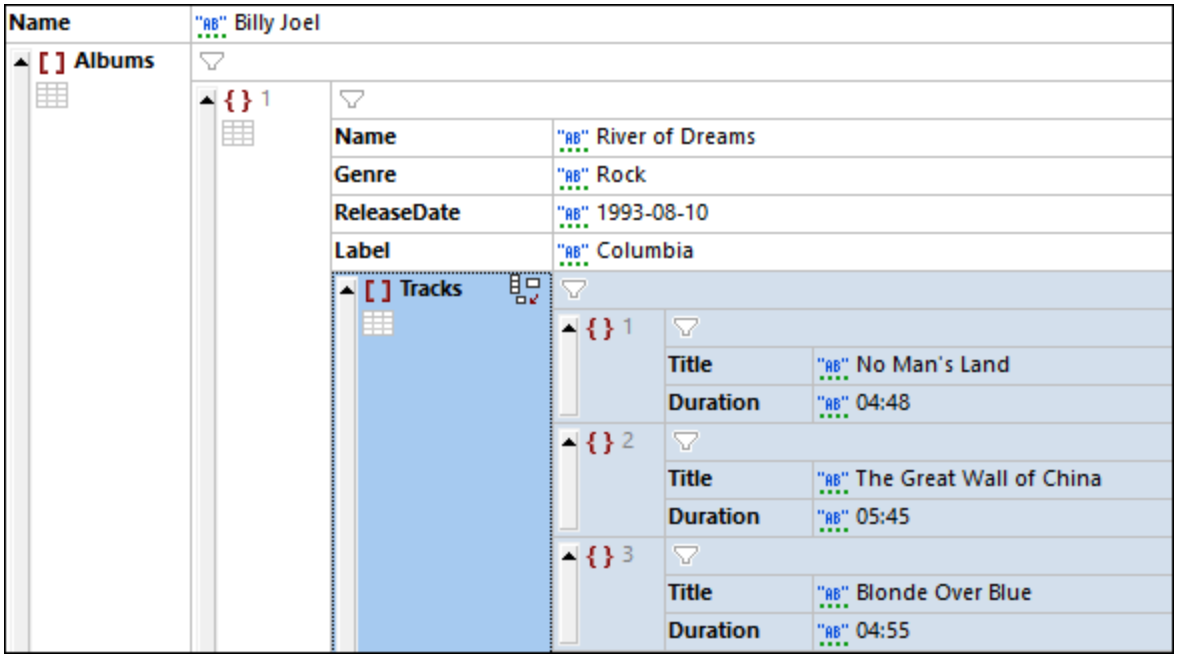
Los valores de propiedad de los objetos y los elementos de las matrices pueden ser de distintos tipos[669]; cada tipo se indica con un símbolo, como se muestra en la imagen siguiente. (En la imagen anterior observe que el tipo del valor de todos los pares `key:value` es el tipo `string`.)

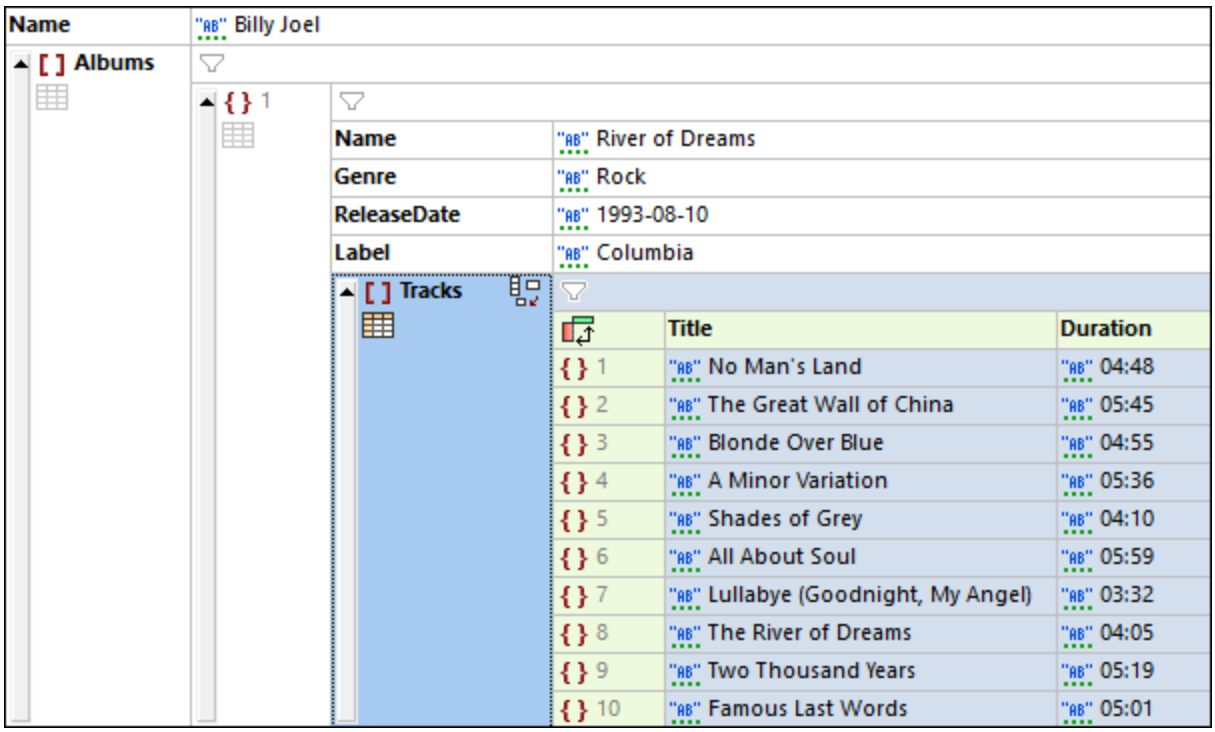


Nota: se admiten comentarios en JSONC y JSON5. Las fórmulas[696] son específicas de XMLSpy.

Mostrar tablas

Los objetos y las matrices que contienen al menos un objeto o una matriz se pueden ver como lista (*imagen izquierda*) o como tabla (*imagen derecha*). Se puede alternar entre la vista de lista y la de tabla para visualizar objetos y matrices individuales.





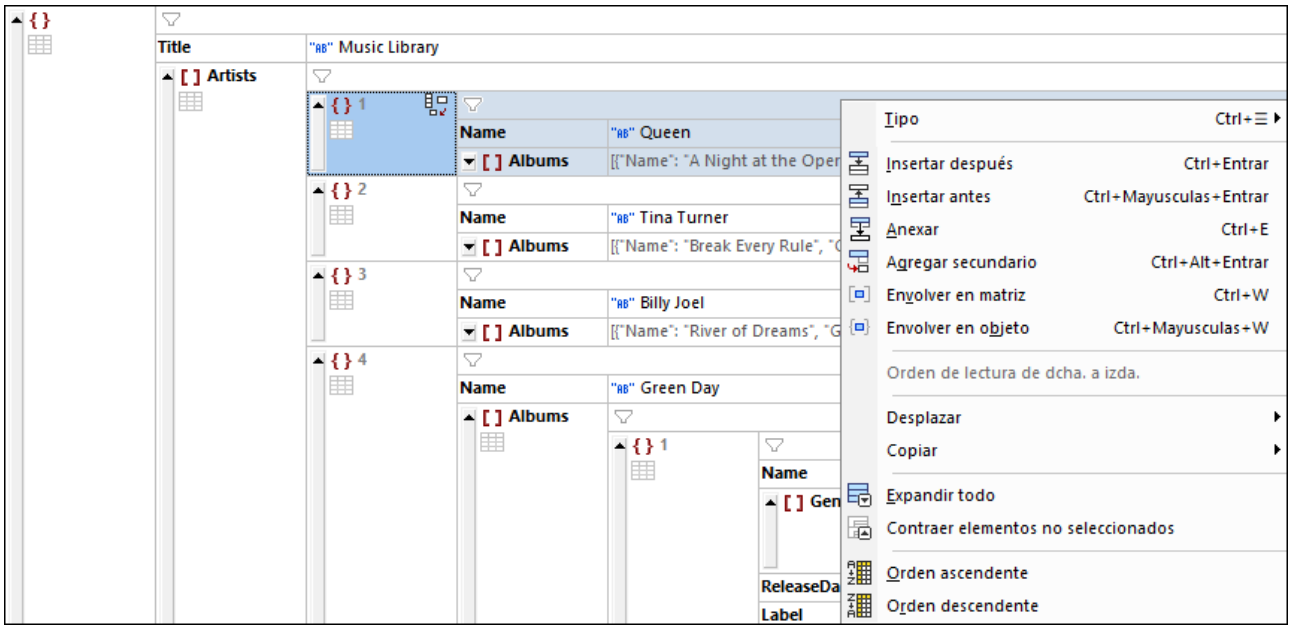
© 2016-2022 Altova GmbH

Para ver una descripción más detallada de la tabla de la vista Cuadrícula JSON consulte el apartado Mostrar tablas 692.

13.5.2 Editar la estructura de un documento

Para editar la estructura del documento JSON puede añadir componentes nuevos, eliminar o mover los bloques seleccionados de la cuadrícula. Puede editar la estructura del documento de varias formas:

- seleccione un componente y haga clic con el botón derecho para acceder a los comandos del menú contextual (*imagen siguiente*). Consulte la descripción de los comandos en el apartado Menú contextual de la vista Cuadrícula 706.
- use los botones de edición que hay dentro de las celdas. Estos botones se describen más abajo en el apartado Botones de comandos dentro de las celdas 686.
- Seleccione un componente y arrástrelo hasta una ubicación nueva para modificar la estructura.



Estos son los accesos rápidos de los comandos para editar una estructura:

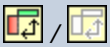
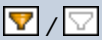
Comando	Acceso rápido
Insertar (mismo tipo de componente) después	Ctr+Entrar
Insertar (mismo tipo de componente) antes	Ctr+Mayús+Entrar
Anexar (mismo tipo de componente)	Ctr+E
Agregar secundario	Ctr+Alt+Entrar

Envolver en matriz	Ctr+W
Envolver en objeto	Ctr+Mayús+W

Nota: si se agrega un componente, por ejemplo como propiedad (insertado, anexado o agregado como secundario) y quiere cambiar su tipo a objeto o matriz, haga clic en el símbolo del tipo de componente y seleccione el tipo que prefiera[686].

Botones de comandos dentro de las celdas

En la vista Cuadrícula JSON los botones que hay dentro de las celdas de la cuadrícula permiten realizar acciones en el contexto de la celda en cuestión. Es más rápido usar estos botones de comando para las acciones que se usan más a menudo. En la tabla siguiente describimos los comandos de estos botones. Los comandos relacionados con la visualización tienen fondo azul. Los comandos para editar el documento (estructura y contenido) tienen fondo beige. Si necesita otros comandos en el contexto de alguna de las celdas puede usar el menú contextual[706] de la celda. Nota: si pasa el cursor del ratón por encima de un botón aparece información rápida con el nombre del comando del botón en cuestión.

	Optimiza el ancho de las columnas para que se ajuste a su contenido.
	Si aparece en color, la Vista Diseño de tabla[692] está activada, si está desactivada aparece en gris. Haga clic en el icono para cambiar la vista
	Este comando cambia las filas por las columnas y viceversa
	Si está naranja significa que hay un filtro[696] para la tabla, el objeto o la matriz en cuestión. Si el icono está blanco, el filtro está desactivado. Para activar/desactivar el filtro, haga clic en el icono. Para editar la expresión, haga doble clic en el icono.
	Este comando activa y desactiva el ajuste automático de línea. Para activarlo o desactivarlo haga clic en el icono. Por defecto solo se aplica este formato al contenido de los elementos. Si quiere que se aplique también a los nombres de los elementos, elija esa opción en la lista desplegable que abre el icono.
	En una matriz: anexar un elemento matriz a la lista o tabla En un objeto: anexar un par `key:value` (como fila en una lista o, en la vista Diseño de tabla, como una celda de una columna de tabla nueva)
	Añadir un par `key:value` vacío; el tipo de ese valor es `string` por defecto
	Seleccionar el tipo de datos del valor de una propiedad o introducir un comentario o una fórmula

Nota: para ver una descripción de otros comandos de la barra de herramientas consulte el apartado Menú contextual de la vista Cuadrícula[706].

13.5.3 Editar el contenido de un documento

Editar el contenido en la vista Cuadrícula JSON es muy sencillo: haga doble clic dentro de la celda que quiere editar y cambie el contenido. La Vista Cuadrícula JSON también contiene estas características:

© 2016-2022 Altova GmbH

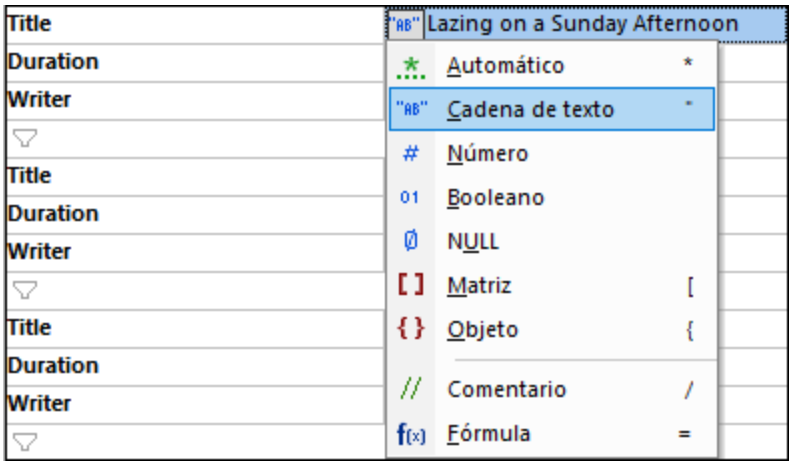
- selección de tipo, detección automática del tipo y conversión de valores
- finalización automática en documentos JSON basados en un esquema JSON

Selección de tipo y detección automática del tipo

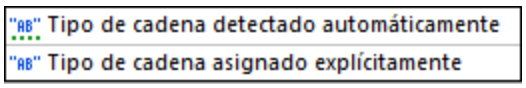
La Vista Cuadrícula JSON distingue entre estas categorías de tipo:

- *Tipos simples:* cadena de texto, número, booleano, null
- *Tipos simples especiales:* automático (se detecta por el valor), comentario (JSONC, JSON5)
- *Tipos de contenedor:* objeto, matriz
- *Tipos de contenedor especiales:* fórmula

El tipo de un valor viene indicado por un símbolo que aparece junto a él (*imagen siguiente*). Para cambiar un tipo haga clic en su símbolo y seleccione el tipo que prefiera del menú que aparece. Estos son los símbolos y accesos rápidos de los tipos:



Los tipos que se detectan automáticamente están subrayados con puntos verdes (*imagen siguiente*). Un tipo explícito es el que usted asigna.



Hay dos situaciones en las que ocurren acciones relacionadas con el tipo:

- cuando se carga un documento JSON: todos los tipos simples se convierten en `Auto`, que se detecta automáticamente a partir de los valores. Por ejemplo, **`"MyString"`** se detecta automáticamente como cadena de texto, **`123`** como número, **`true`** como booleano y **`null`** como valor nulo. Si una cadena de texto es ambigua, seleccione el tipo de forma explícita.
- cuando se introducen una estructura de datos o un valor nuevos: los tipos simples se detectan y el tipo se asigna automáticamente. Puede cambiar el tipo más tarde si quiere.

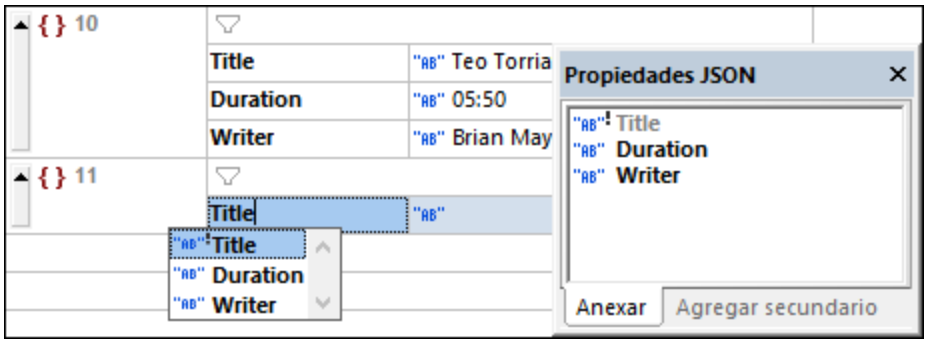
Nota: las opciones de la vista Cuadrícula JSON [709] permiten indicar (i) cómo se deben tratar los cambios de tipo si se seleccionan varias celdas y (ii) cómo se deben tratar los valores de los tipos atómicos cuando el tipo cambie a matriz u objeto.

Finalización automática

La finalización automática se habilita si el documento JSON (instancia/esquema JSON o esquema Avro) que se está editando está asociado a un esquema.

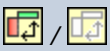
- si el documento es un esquema JSON, entonces la finalización automática se basa en la versión del esquema que indique la palabra clave `$schema`[671]. Para más información consulte también Versión del esquema JSON[715].
- si el documento es una instancia JSON/JSON5, entonces se debe asignar un esquema JSON a la instancia[754] para que se habilite la finalización automática.
- si el documento es un documento de datos Avro en formato JSON, entonces se debe asignar un esquema Avro a la instancia[754] para que funcione la finalización automática.
- si el documento es un esquema Avro[769], entonces está automáticamente asociado al esquema de Avro Schema y la finalización automática se basa en ese esquema.

La finalización automática sugiere opciones de entrada en el punto en que se encuentre el cursor. Esas opciones las sugiere (i) mediante menús emergentes en la ventana principal y (ii) con los ayudantes de entrada (*imagen siguiente*). Los menús emergentes y los ayudantes de entrada muestran una lista de entradas que son válidas para el punto en que se encuentre el cursor. Para pasar de una opción a otra en el menú emergente, use las teclas de flechas. Seleccione una entrada de la ventana emergente o haga doble clic en una entrada del ayudante de entradas para insertarla.

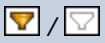


Botones de comando dentro de las celdas

En la vista Cuadrícula JSON los botones que hay dentro de las celdas de la cuadrícula permiten realizar acciones en el contexto de la celda en cuestión. Es más rápido usar estos botones de comando para las acciones que se usan más a menudo. En la tabla siguiente describimos los comandos de estos botones. Los comandos relacionados con la visualización tienen fondo azul. Los comandos para editar el documento (estructura y contenido) tienen fondo beige. Si necesita otros comandos en el contexto de alguna de las celdas puede usar el menú contextual[706] de la celda. Nota: si pasa el cursor del ratón por encima de un botón aparece información rápida con el nombre del comando del botón en cuestión.

	Optimiza el ancho de las columnas para que se ajuste a su contenido.
	Si aparece en color, la Vista Diseño de tabla[692] está activada, si está desactivada aparece en gris. Haga clic en el icono para cambiar la vista
	Este comando cambia las filas por las columnas y viceversa

© 2016-2022 Altova GmbH

	Si está naranja significa que hay un filtro[696] para la tabla, el objeto o la matriz en cuestión. Si el icono está blanco, el filtro está desactivado. Para activar/desactivar el filtro, haga clic en el icono. Para editar la expresión, haga doble clic en el icono.
	Este comando activa y desactiva el ajuste automático de línea. Para activarlo o desactivarlo haga clic en el icono. Por defecto solo se aplica este formato al contenido de los elementos. Si quiere que se aplique también a los nombres de los elementos, elija esa opción en la lista desplegable que abre el icono.
	En una matriz: anexar un elemento matriz a la lista o tabla En un objeto: anexar un par `key:value` (como fila en una lista o, en la vista Diseño de tabla, como una celda de una columna de tabla nueva)
	Añadir un par `key:value` vacío; el tipo de ese valor es `string` por defecto
	Seleccionar el tipo de datos del valor de una propiedad o introducir un comentario o una fórmula

Nota: para ver una descripción de otros comandos de la barra de herramientas consulte el apartado Menú contextual de la vista Cuadrícula[706].

Validar al modificar

El modo *Validar al editar*, está activado por defecto, permite validar un documento y comprobar si su formato es el correcto mientras lo modifica en la vista Cuadrícula JSON. Para validar un documento JSON (además de comprobar su formato) debe asignar un esquema JSON a ese documento. Si hay errores, el texto que los contiene aparece en rojo y el lugar del error se marca con un signo rojo de exclamación. Para más información consulte Validar documentos JSON[754].

El modo *Validar al editar* se puede activar y desactivar (i) con el comando de menú **XML | Validar al editar**[1344], (ii) el botón **Validar al editar** de la barra de herramientas o (iii) con la opción *Al editarlos* de las las opciones de validación del cuadro de diálogo "Opciones"[1584].

Buscar y reemplazar

Puede usar los comandos **Buscar** (**Ctrl+F**) y **Reemplazar** (**Ctrl+H**) del menú Edición para encontrar texto dentro de la vista Cuadrícula JSON y reemplazarlo. Las celdas que coincidan con la búsqueda se resaltan en gris y si hay componentes que contienen que coincidan con esa búsqueda, estos se resaltan en naranja. En caso de que las celdas contengan cadenas de texto largas solamente se resalta la celda.

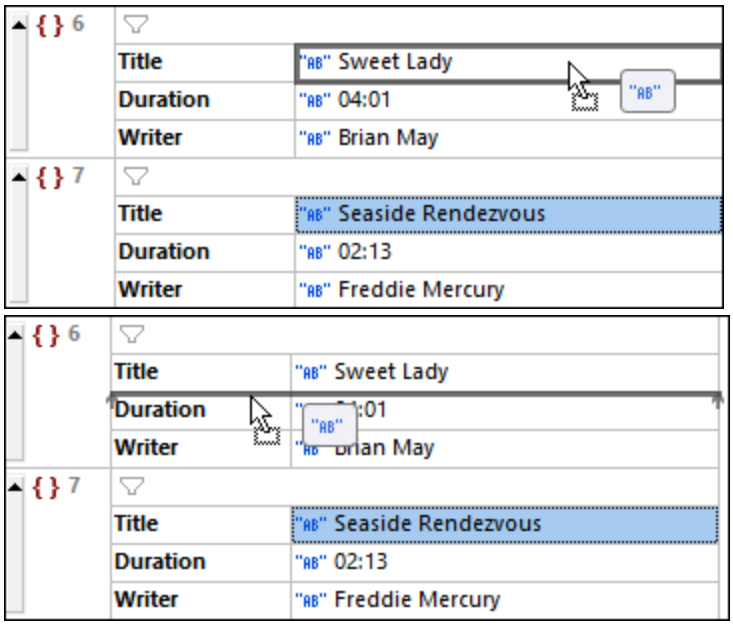
13.5.4 Arrastrar y colocar en la vista Cuadrícula

La vista Cuadrícula permite arrastrar fragmentos de documentos JSON a la vista Cuadrícula desde un documento que esté abierto en XMLSpy, una aplicación externa o incluso un sitio web.

Al colocar el cursor sobre el nodo de destino aparece una capa que no solo ofrece información sobre lo que se está arrastrando, sino también sobre cómo se creará ese elemento cuando se coloque. Por ejemplo, en la imagen siguiente se está arrastrando el valor **`Seaside Rendezvous`**. La ventana emergente **AB** indica que se trata de un valor de cadena.

- Si pasa el cursor sobre un campo de valor (como en la imagen, a la izquierda) se marca todo el campo, lo que indica que el valor de ese campo será reemplazado por el valor del campo que se está arrastrando.
- Si pasa el cursor del ratón sobre un nombre de nodo (como en la imagen, a la derecha, con `Duration`), entonces aparece una línea de inserción que indica que la cadena de valor se colocará allí para crear el nodo de valor de un par clave-valor. La clave en este caso será una cadena vacía.

Nota: puede deshacer cualquier acción con **Ctrl+Z** o el comando de menú **Edición | Deshacer**.



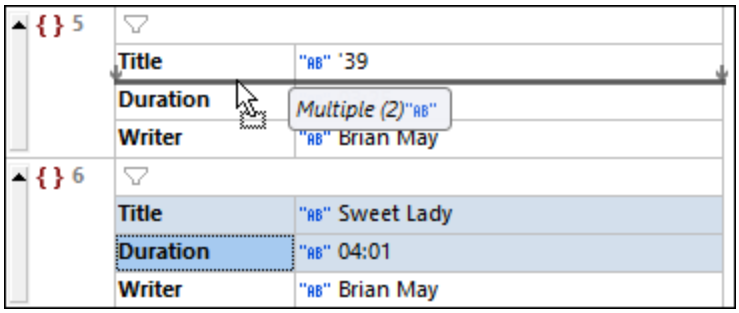
Información de la capa de arrastre

Estos son los tipos de información de la capa de arrastre.

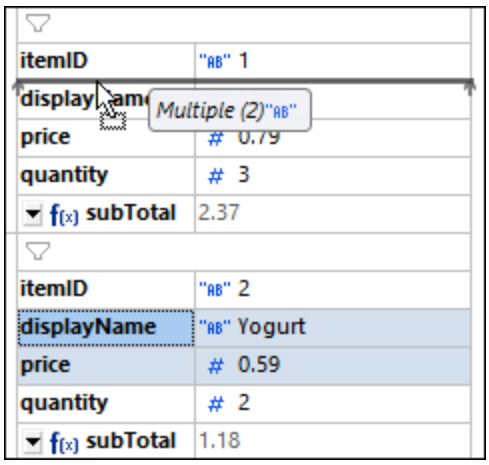
Vista Cuadrícula normal

- El valor que se coloca en un campo de valor mueve el valor de origen para sobrescribir el valor del campo de destino (*imagen anterior a la izquierda*).
- El valor que se coloca en la estructura como nodo crea el nodo de valor de un par clave-valor (*imagen anterior a la derecha*). La clave en este caso será una cadena vacía.
- Si hay varios nodos de un mismo tipo se muestran el tipo de nodo que se va a colocar y el número de nodos. Tenga en cuenta que para seleccionar un nodo (y no su valor) debe hacer clic en el nombre del nodo. En la imagen siguiente la información de la capa de arrastre indica que se van a colocar dos pares clave-valor de tipo cadena.

© 2016-2022 Altova GmbH

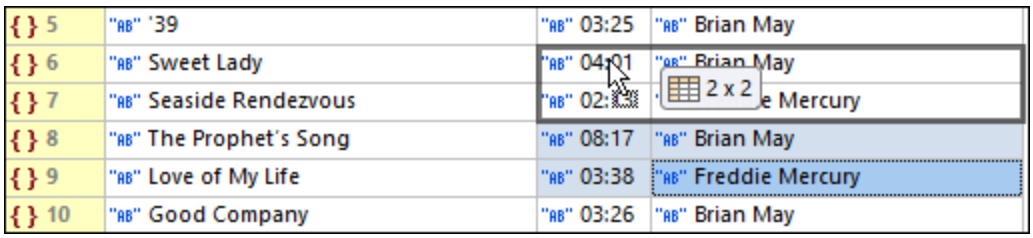


- Si hay varios nodos de distintos tipos se muestra la información de que se van a crear nodos nuevos con un tipo predeterminado de cadena (*imagen siguiente*). Puede cambiar el tipo de nodos que coloque haciendo clic en el icono *Editar tipo* correspondiente. Tenga en cuenta que para seleccionar un nodo (y no su valor) debe hacer clic en el nombre del nodo.



Vista Diseño de tabla

- Al arrastrar valores, el número de celdas seleccionadas viene indicado por `columna x filas` (*imagen siguiente*). La matriz de celdas arrastradas reemplaza a la matriz correspondiente. La celda de destino (en la que coloca la matriz) recibe el valor de la celda superior izquierda de la matriz. El resto de celdas rellena las celdas de destino hacia la derecha y hacia abajo. Por ejemplo, en la imagen siguiente las celdas azules se arrastran hasta colocarlas en la celda `Duration` de la fila 6. Las celdas reemplazan las celdas que se habían marcado. Si la matriz de celdas que se arrastran supera los límites de la tabla se añaden las filas o columnas necesarias para que quepan todas las celdas. En este caso los límites que se han tenido que ampliar aparecen como líneas discontinuas.



- Si el destino es una estructura de nodos, el nodo que se añade se indica junto con (i) las columnas que se van a añadir (entre corchetes) y (ii) el número de instancias del elemento nuevo. Por ejemplo,

en la imagen siguiente se añade una matriz que contiene dos objetos, cada uno de los cuales tiene un par clave-valor **`Title`**, **`Duration`** y **`Writer`**.



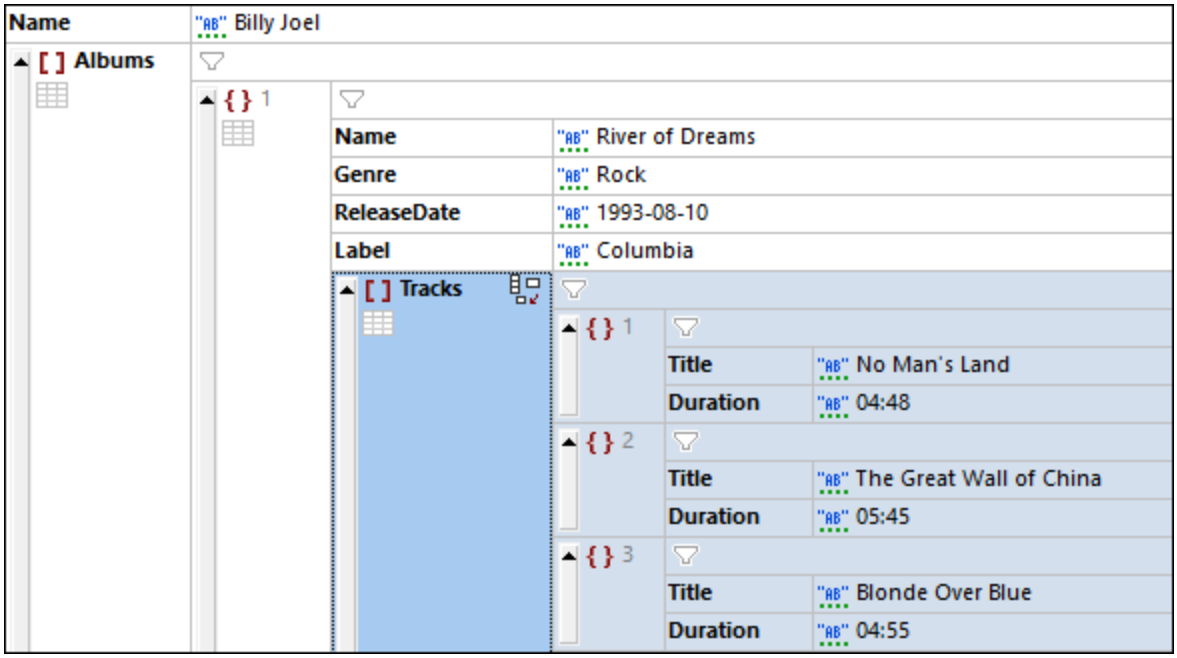
- Puede mover columnas de tabla seleccionándolas (para ello haga clic en su encabezado) y arrastrándolas hasta el encabezado de columna adyacente al que quiere mover. Una línea de inserción le indicará a qué lado de la columna de destino se colocará la columna que arrastra.

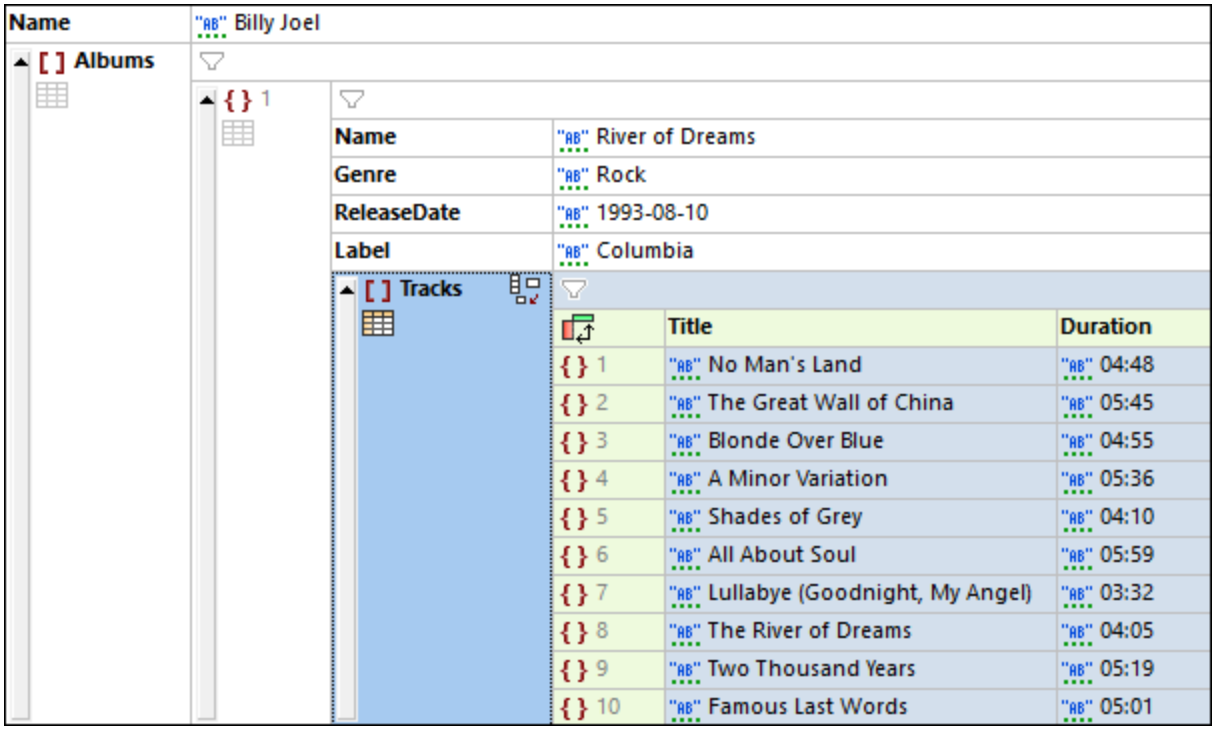
13.5.5 Vista Diseño de tabla

Vista Diseño de tabla

Los objetos y las matrices que contienen al menos un objeto o una matriz se pueden visualizar como una lista (*resaltada en la imagen izquierda*) o como una tabla (*resaltada en la imagen derecha*). Para cambiar de una vista a otra haga clic en el icono de la tabla situado bajo el icono del objeto o de la matriz (*imagen siguiente*). En las tablas de matriz los elementos de matriz están colocados en filas. Por ejemplo, en la imagen de la derecha la matriz `Tracks` aparece como una tabla. Esta matriz consiste en objetos secundarios que aparecen como filas. Las propiedades de cada uno de los objetos () aparecen como columnas, con los nombres de las propiedades comunes como encabezados de las columnas. Para convertir las filas en columnas y viceversa haga clic en el icono que hay en la celda de la parte superior izquierda de la tabla.

© 2016-2022 Altova GmbH

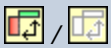




Iconos para ver y editar en la vista Diseño de tabla

Estos iconos están disponibles en la vista Diseño de tabla y representan funciones de visualización y edición.

	Optimiza el ancho de las columnas para que se ajuste a su contenido.

	Si aparece en color, la Vista Diseño de tabla 692 está activada, si está desactivada aparece en gris. Haga clic en el icono para cambiar la vista
	Este comando cambia las filas por las columnas y viceversa
	Si está naranja significa que hay un filtro 696 para la tabla, el objeto o la matriz en cuestión. Si el icono está blanco, el filtro está desactivado. Para activar/desactivar el filtro, haga clic en el icono. Para editar la expresión, haga doble clic en el icono.
	Este comando activa y desactiva el ajuste automático de línea. Para activarlo o desactivarlo haga clic en el icono. Por defecto solo se aplica este formato al contenido de los elementos. Si quiere que se aplique también a los nombres de los elementos, elija esa opción en la lista desplegable que abre el icono.
	En una matriz: anexar un elemento matriz a la lista o tabla En un objeto: anexar un par `key:value` (como fila en una lista o, en la vista Diseño de tabla, como una celda de una columna de tabla nueva)
	Añadir un par `key:value` vacío; el tipo de ese valor es `string` por defecto
	Seleccionar el tipo de datos del valor de una propiedad o introducir un comentario o una fórmula

Editar en la vista Diseño de tabla

A continuación explicamos las acciones que puede realizar en la vista Diseño de tabla.

Añadir filas a una tabla
Puede añadir filas de varias maneras:

- *Añadir un elemento fila de nivel inferior*. Seleccione la tabla y, en el menú contextual, seleccione el comando **Agregar secundario (Ctrl+Mayús+Entrar**). Otra opción es hacer clic en el icono **Anexar secundario** de la tabla (*véase la lista más arriba*) para anexar el elemento fila al final de la tabla.
- *Añadir un elemento fila del mismo nivel cuando se selecciona una fila*. Seleccione una fila y, en el menú contextual, seleccione el comando **Anexar** (**Ctrl+Entrar**) para anexar un elemento fila al final de la tabla.
- *Insertar un elemento fila del mismo nivel encima de la fila seleccionada. Seleccione la fila y, en el menú contextual, seleccione el comando* **Insertar (Ctrl+Alt+Entrar**).

Introducir o editar el valor de una propiedad
Seleccione la celda de la tabla en la que esté el valor de la propiedad. Los tipos cadena de texto, número, booleano y null se detectan automáticamente. En los casos ambiguos se aplica el tipo `string` (cadena). Puede sobrescribir la selección automática con el icono de selección de tipo de datos (*véase la lista de iconos más arriba*). También puede usar un acceso rápido (*véase la tabla siguiente*).

*	Automático (detecta si el tipo es cadena de texto, número, booleano o null y elige la configuración apropiada)
"	Cadena de texto
[	Matriz
{	Objeto

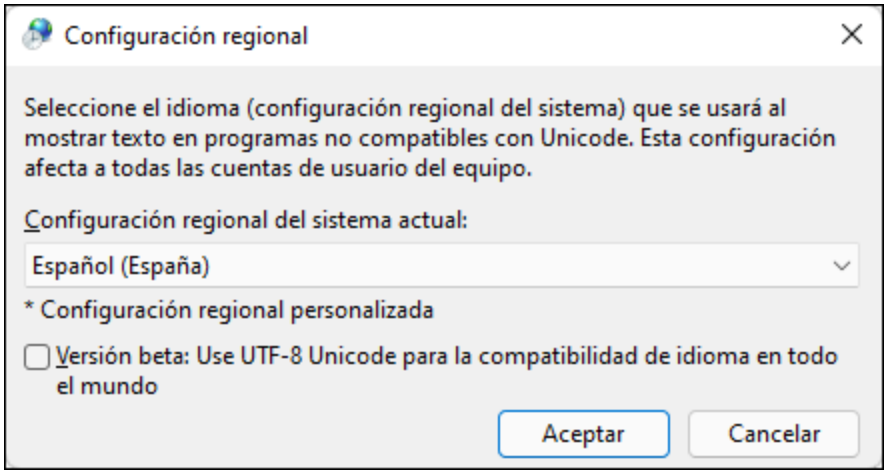
© 2016-2022 Altova GmbH

/	Comentario (el documento debe ser JSONC o JSON5)
=	Fórmula (el documento debe ser JSONC o JSON5)

Añadir una celda vacía nueva en una columna nueva
Puede añadir una celda nueva a una fila. La celda nueva será parte de una columna nueva. Para añadir la celda nueva seleccione la fila que prefiera y, en el menú contextual, seleccione el comando Agregar secundario (**Ctrl+Mayús+Entrar**). Otra opción es hacer clic en el icono **Anexar celda en la fila** (*ver la lista de iconos más arriba*). La celda se crea en una columna nueva. El tipo del valor de la celda es `string` por defecto. Introduzca el valor de la propiedad en la celda y el nombre de la propiedad como encabezado de la columna.

Ordenar las filas de una tabla según los valores de la columna que seleccione
Puede ordenar las filas de una tabla según los valores relativos de uno de sus nodos secundarios (una columna). Por ejemplo, puede ordenar filas en base a la columna *LastName* para obtener los elementos de la tabla que se repiten en orden alfabético. Para ordenar una columna seleccione su encabezado y haga clic en el comando **Orden ascendente** u **Orden descendente** de la barra de herramientas de la vista Cuadrícula 681. Estos comandos también están disponibles en el menú JSON 1353.

Sort order in some languages, especially those with non-Latin alphabets, may benefit from enabling the beta Unicode UTF-8 support in the Language Region Settings dialog of Windows 10 (or later). Do this as follows: Go to your Windows *Settings* dialog and search for *Language Settings*. Under *Related Settings*, click *Administrative language settings*. In the Region dialog that appears, go to the *Administrative* tab, and, under *Language for non-Unicode programs*, click *Change system locale*. In the Region Settings dialog that appears (*screenshot below*), select the option *Beta: Use Unicode UTF-8 for worldwide language support* and click **OK**.



Mover datos de la vista Tabla a aplicaciones externas y viceversa

Puede sacarle partido a la estructura tabular de los datos de la vista Tabla para intercambiar datos entre esta vista y una aplicación de hojas de cálculo. Para mover los datos de la vista Tabla a otra aplicación, seleccione los nodos pertinentes de la tabla y use la opción Copiar como texto estructurado 1282 para copiar y pegar los datos directamente en una hoja de cálculos de, por ejemplo, Excel. (Puede seleccionar los nodos de la vista Tabla haciendo clic en celdas, encabezados de columna, encabezados de fila o en la tabla entera). Si hace clic en la tabla entera o en encabezados de columna, también se copiará el texto de los encabezados de columna.)

item (6)

	= partNum	{} productName	{} quantity	{} price	{} shipDate	{} *ipo:*comment
1	833-AA	Lapis necklace	2	99.95	1999-12-05	Need this for the holidays!
2	748-OT	Diamond heart	1	248.90	2000-02-14	Valentine's day packaging.
3	783-KL	Uncut diamond	7	79.90	2000-01-07	
4	238-KK	Amber ring	3	89.90	2000-01-07	With no inclusions, please.
5	229-OB	Pearl necklace	1	4879.00	1999-12-05	
6	128-UL	Jade earring	5	179.90	2000-02-14	

La imagen anterior muestra seis elementos `item` en forma de tabla en la vista Tabla, donde se seleccionaron dos filas. Para copiar estas dos filas y pegarlas en una hoja de Excel, copie las filas como texto estructurado y péguelas en la hoja de Excel. El resultado será el que aparece en la imagen siguiente. Para más información consulte la descripción del comando en la *Comandos de menú.*

	A	B	C	D	E	F
1	238-KK	Amber ring	3	89.9	01/07/2000	With no inclusions, please.
2	229-OB	Pearl necklace	1	4879	12/05/1999	
3						
4						

El intercambio de datos funciona también en la dirección inversa. Es decir, puede copiar datos de cualquier aplicación de hoja de cálculo e insertarlos directamente en una tabla de la vista Tabla. Para ello, siga estos pasos:

1. Seleccione un rango de datos en la aplicación externa y cópielo (al portapapeles, **Ctrl+C** en Windows)
2. Seleccione una celda de la vista Tabla de su documento XML.
3. Pegue los datos con **Ctrl+V**.

Los datos se pegarán en la tabla de XMLSpy con una estructura equivalente a la estructura original y empezando en la celda seleccionada en la vista Tabla. Cabe destacar que:

- si ya existen datos en estas celdas de la vista Tabla, los datos nuevos sobrescribirán los datos originales.
- si no hay suficientes filas para alojar los datos nuevos, se crearán las filas que hagan falta.
- si no hay suficientes columnas para alojar los datos nuevos, **no se crearán** columnas nuevas.
- los datos de las celdas se convierten en el contenido de los elementos representados por las celdas correspondientes.

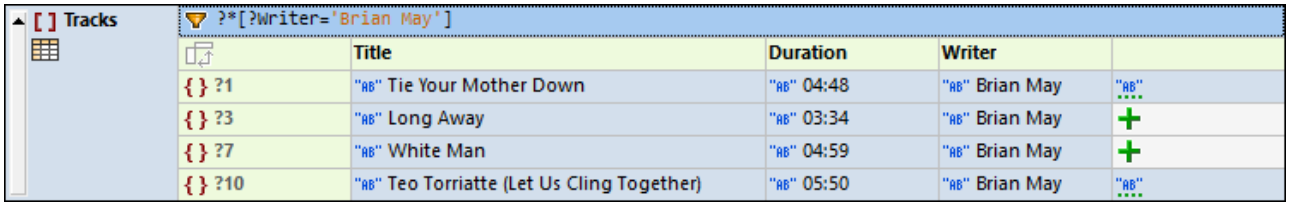
Si desea realizar tareas de intercambio de datos más complejas, XMLSpy también ofrece funciones únicas de conversión que le ayudarán a importar o exportar datos XML[1452] desde cualquier archivo de texto, documento Word o base de datos.

13.5.6 Filtros

Los filtros permiten filtrar qué objetos y matrices se ven en la presentación. Por ejemplo, en la imagen siguiente se ha aplicado un filtro (que es una expresión XQuery 3.1) a una matriz de forma que solo se vean las

© 2016-2022 Altova GmbH

canciones escritas por Brian May. En términos JSON, solo se ven los objetos que son secundarios de **Tracks** que tienen una propiedad **Writer** que contiene el valor de cadena `Brian May`. La expresión XQuery del filtro busca todos los objetos secundarios de **Tracks** y seleccione aquellos en los que la propiedad **Writer** coincide con la cadena `'Brian May'`.



Cada filtro funciona de forma independiente y no se ve afectado por el resto de filtros o fórmulas.

Para más información sobre generar expresiones XQuery para documentos JSON consulte el apartado Expresiones XQuery para JSON[760].

Nota: al introducir expresiones para filtros y fórmulas en la vista Cuadrícula puede que quiera ver una expresión en varias líneas para poder leerla mejor. Para añadir una línea nueva en una expresión pulse **Ctrl+Entrar**.

Configurar un filtro

Para configurar un filtro haga clic con el botón derecho en el elemento que quiere filtrar, seleccione **Filtro** en el menú contextual que aparece, introduzca la expresión XQuery y haga clic en **Entrar**. La celda *Filtro* viene indicada por el icono (*imagen siguiente*). Puede activar/desactivar el filtro haciendo clic en el icono *Filtro*.

Tenga en cuenta estos aspectos sobre los filtros:

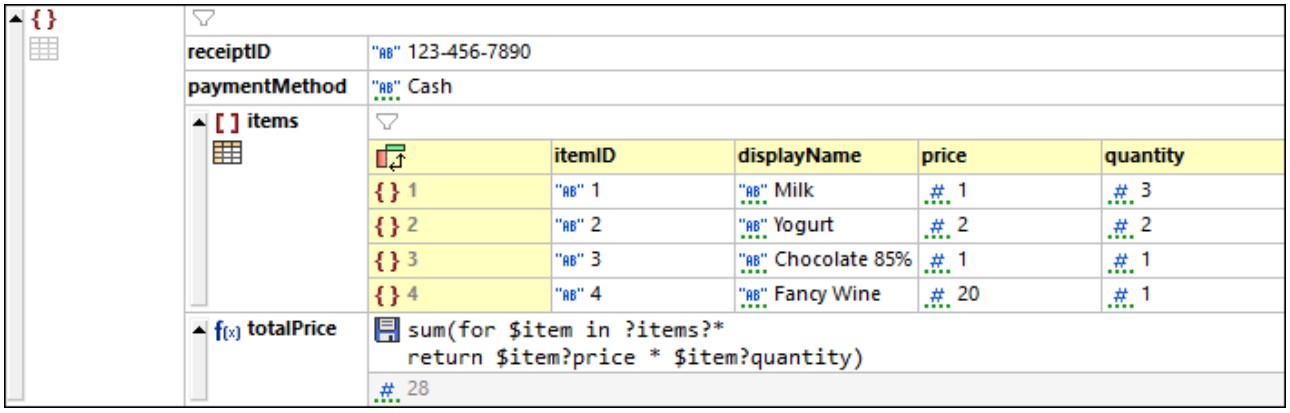
- Los filtros solo se pueden aplicar a objetos y matrices.
- El nodo de contexto de la expresión XQuery del filtro es el nodo actual. En la imagen anterior, por ejemplo, el nodo de contexto de la expresión XQuery del filtro es el nodo **Tracks**.
- Los filtros **no se almacenan** en el documento JSON con contenido JSON, pero se pueden conservar en un archivo de metadatos para su uso posterior. Si se guardan los filtros, estos se aplican automáticamente al objeto o a la matriz correspondientes cuando el documento se vuelve a abrir en la vista Cuadrícula JSON.
- Los filtros se pueden anidar. Se puede aplicar un filtro anidado al contenido filtrado del filtro principal.
- Para añadir una línea nueva en una expresión pulse **Ctrl+Entrar**.

13.5.7 Fórmulas

Una fórmula es una expresión XQuery 3.1 que genera unos resultados (un conjunto de nodos o un cálculo) que aparecen en la vista Cuadrícula JSON. En la imagen siguiente, por ejemplo, se calcula el precio total de los elementos 1 a 4 y el resultado (`28`) aparece en una línea aparte. Cada fórmula funciona de forma independiente y no se ve afectada por el resto de filtros o fórmulas.

Para más información sobre generar expresiones XQuery para documentos JSON consulte el apartado Expresiones XQuery para JSONJSON[760].

Nota: al introducir expresiones para filtros y fórmulas en la vista Cuadrícula puede que quiera ver una expresión en varias líneas para poder leerla mejor. Para añadir una línea nueva en una expresión pulse **Ctrl+Entrar**.



Si el documento es JSON5 o JSONC:

- La fórmula se escribe en el documento como comentario JSON
- El **resultado** de la fórmula se puede almacenar como código (sin comentarios) en el documento JSON. Para ello haga clic en el icono del disco de la fórmula (*imagen siguiente*). Si la fórmula no se almacena como contenido JSON se guarda solamente como metadatos del documento para su visualización en la vista Cuadrícula. Tenga en cuenta que el icono del disco solo está disponible en documentos JSON5 y JSONC, pero no en otros tipos de documentos JSON.

Los documentos JSON de la imagen anterior, incluidos la fórmula almacenada y sus resultados.

Tenga en cuenta que la ***fórmula*** se almacena como comentario JSON, pero el ***resultado*** lo hace directamente como código JSON. *El código siguiente (donde el resultado de la fórmula se almacena) es el resultado de hacer clic en el icono del disco de la fórmula.*

```
{
	"receiptID": "123-456-7890",
	"paymentMethod": "Cash",
	"items": [
		{
			"itemID": "1",
			"displayName": "Milk",
			"price": 1,
			"quantity": 3
		},
		{
			"itemID": "2",
			"displayName": "Yogurt",
			"price": 2,
			"quantity": 2
		},
		{
			"itemID": "3",
			"displayName": "Chocolate 85%",
			"price": 1,
			"quantity": 1
```


© 2016-2022 Altova GmbH

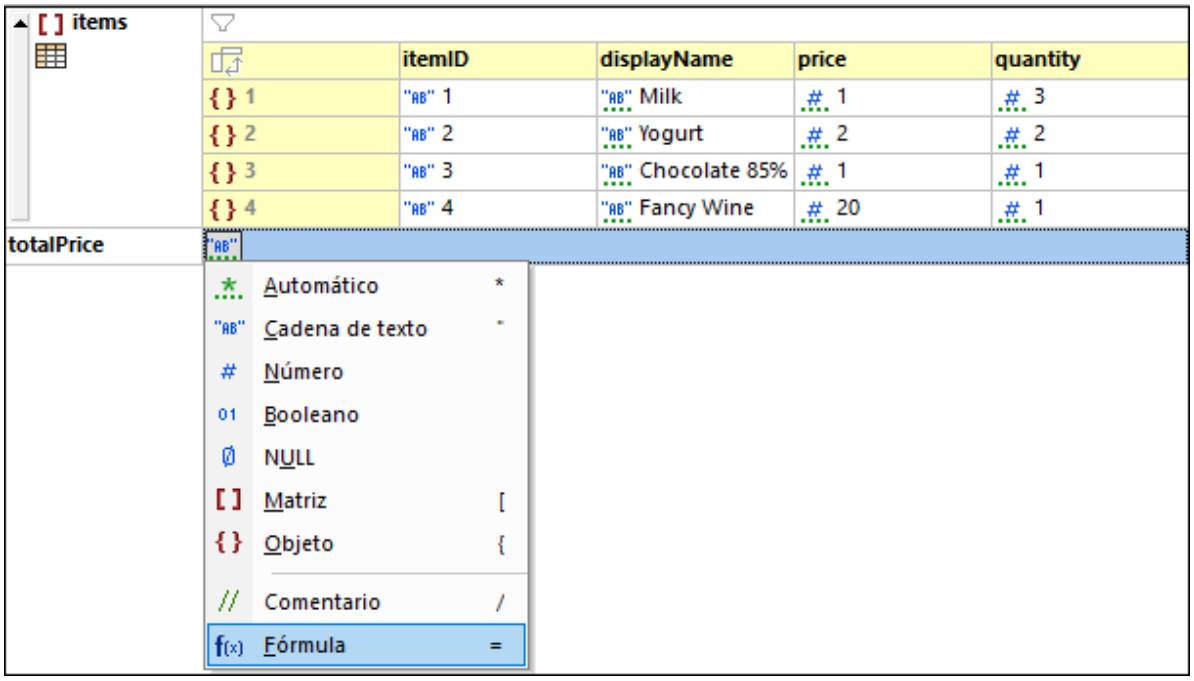
```
                },
                {
                        "itemID": "4",
                        "displayName": "Fancy Wine",
                        "price": 20,
                        "quantity": 1
                }
        ],
        //(:altova_xq_embed:)totalPrice(:altova_xq_key:)sum(for $item in ?items?* return
$item?price * $item?quantity)
        "totalPrice": 28
        //(:altova_xq_end:)
}
```

En la imagen anterior la fórmula calcula el total de los elementos de una secuencia. Cada uno de estos elementos es el producto de los valores **price** y **quantity** de cada uno de los objetos que contiene la matriz **items**. La iteración para seleccionar cada uno de los objetos y asignarlos a su vez a la variable **$item** se indica con: **for $item in ?items?***. En este punto es importante anotar el nodo de contexto, que es el elemento principal de la fórmula y, por consiguiente, el elemento principal del nodo **items**. Para obtener cada producto se buscan los nodos secundarios **price** y **quantity** del objeto que esté actualmente en la variable **$item** y se multiplican esos dos valores. Los productos que se obtienen de esta forma son los elementos de la secuencia, que luego se suman para obtener el precio total.

Crear una fórmula

Siga estos pasos para crear una fórmula:

1. Seleccione el nodo al que quiere añadir la fórmula, sea como elemento del mismo nivel o como secundario. Haga clic con el botón derecho y añada el elemento del mismo nivel o secundario. Al decidir dónde quiere que aparezca la fórmula (como elemento del mismo nivel o secundario) tenga en cuenta que el nodo de contexto de la expresión XQuery de la fórmula será el nodo principal de esa fórmula. Por ejemplo, en la expresión XQuery de la imagen anterior el nodo de contexto es el nodo principal de la fórmula (**totalPrice**) y de su elemento del mismo nivel, la matriz **items**. Para crear la fórmula de la imagen anterior se anexó un nodo del mismo nivel a la matriz **items** (*imagen siguiente*).
2. Cambie el tipo de nodo a Fórmula (*imagen siguiente*).



3. Haga doble clic en la celda que contiene el icono **f(x)** e introduzca el nombre de la fórmula (*imagen siguiente).*). Si el documento es JSON5 o JSONC aparece un icono de disco. Puede hacer clic en este icono para guardar los resultados de la fórmula en el documento.



4. Por defecto, la expresión XQuery es la cadena **'XQuery'**, por lo que el resultado será la cadena **XQuery** (que aparece en la celda bajo la expresión). Haga doble clic en la celda de la expresión e introduzca la expresión, después pulse **Entrar**. La fórmula se evalúa y aparece el resultado. Otras fórmulas no cambian.
5. Si hay varias fórmulas en el documento haga clic en el comando de menú **JSON | Volver a evaluar todo** para actualizar los resultados de todas las fórmulas. Este comando es especialmente útil si las fórmulas del documento buscan datos que cambian de forma dinámica (como tasas de cambio).

Tenga en cuenta estos aspectos sobre las fórmulas:

- El nodo de contexto de la expresión XQuery de la fórmula es el nodo principal del nodo de la fórmula.
- Las fórmulas se almacenan en los metadatos del documento (que está en un archivo JSON de metadatos especial que sirve para toda la aplicación). Si el documento es JSON5 y JSONC, entonces la fórmula se almacena también como comentario en el contenido JSON.
- El resultado generado por una fórmula aparece en la celda bajo la expresión XQuery de la fórmula. En el caso de JSON5 y JSONC, el resultado se puede guardar en el documento haciendo clic en el icono de disco que hay junto a la expresión XQuery. Haga o no clic en el disco, el resultado de la fórmula se calcula y se almacena en los metadatos del documento.

	El resultado de la fórmula no se ha guardado en el contenido JSON; haga clic para guardarlo. *Solo en JSON5 y JSONC.*

© 2016-2022 Altova GmbH

	El resultado de la fórmula se ha guardado en el contenido JSON; haga clic para revertir la acción guardar. *Solo en JSON5 y JSONC.*

- Si el resultado de la fórmula es un cálculo que se almacena en el contenido, se almacena como una propiedad con el nombre que haya asignado a la fórmula. Por ejemplo, en la fórmula `totalPrice` mencionada anteriormente, el resultado se almacena así: `"totalPrice": 28`.
- Observe que: En documentos JSON5 y JSONC las fórmulas se guardan como comentarios JSON y sus resultados se guardan como propiedades JSON.

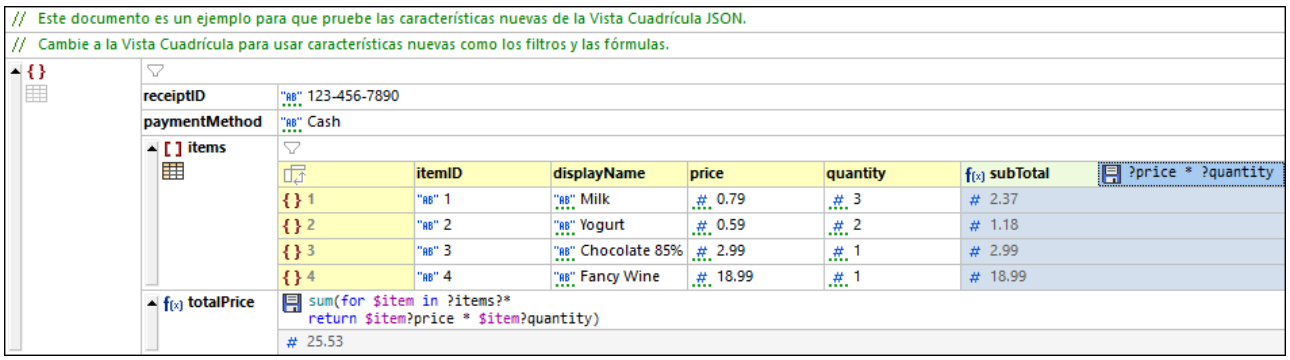
Nota: en el cuadro de diálogo Configurar la vista Cuadrícula 709 puede indicar si quiere guardar las fórmulas en los documentos JSON5 y JSONC o no. La opción predeterminada es que las fórmulas sí se guardan.

Nota: en el cuadro de diálogo Configurar la vista Cuadrícula JSON 709 también puede: (i) definir las opciones de los metadatos (si las fórmulas se guardan automáticamente en el archivo de metadatos, solo tras preguntar o si no se guardan).

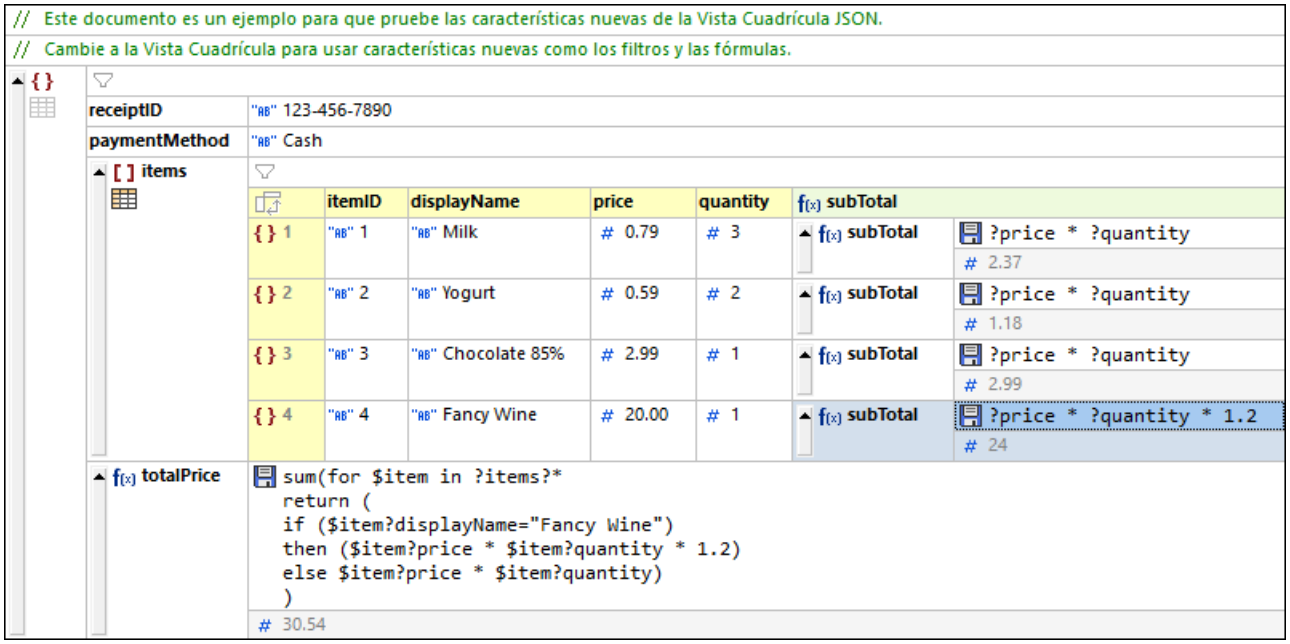
Fórmulas en tablas

Si todas las celdas de una columna de tabla (en Vista Diseño de tabla 692) contienen la misma fórmula, entonces la fórmula aparece solamente una vez, en el encabezado de la columna (*imagen siguiente*). Los *resultados* del cálculo de la fórmula aparecen en las celdas correspondientes.

La fórmula del encabezado de la columna es una representación de la vista Cuadrícula. En el contenido del documento JSON (en la vista Texto), la fórmula se repite por cada elemento fila de tabla.



Si alguna de las fórmulas es diferente (como en la celda resaltada de la imagen siguiente), cada fórmula aparece en su respectiva celda.



Volver a evaluar todas las fórmulas
Para actualizar los resultados de todas las fórmulas del documento haga clic en el comando de menú **JSON | Volver a evaluar todo**.

Procedimiento para acciones relevantes

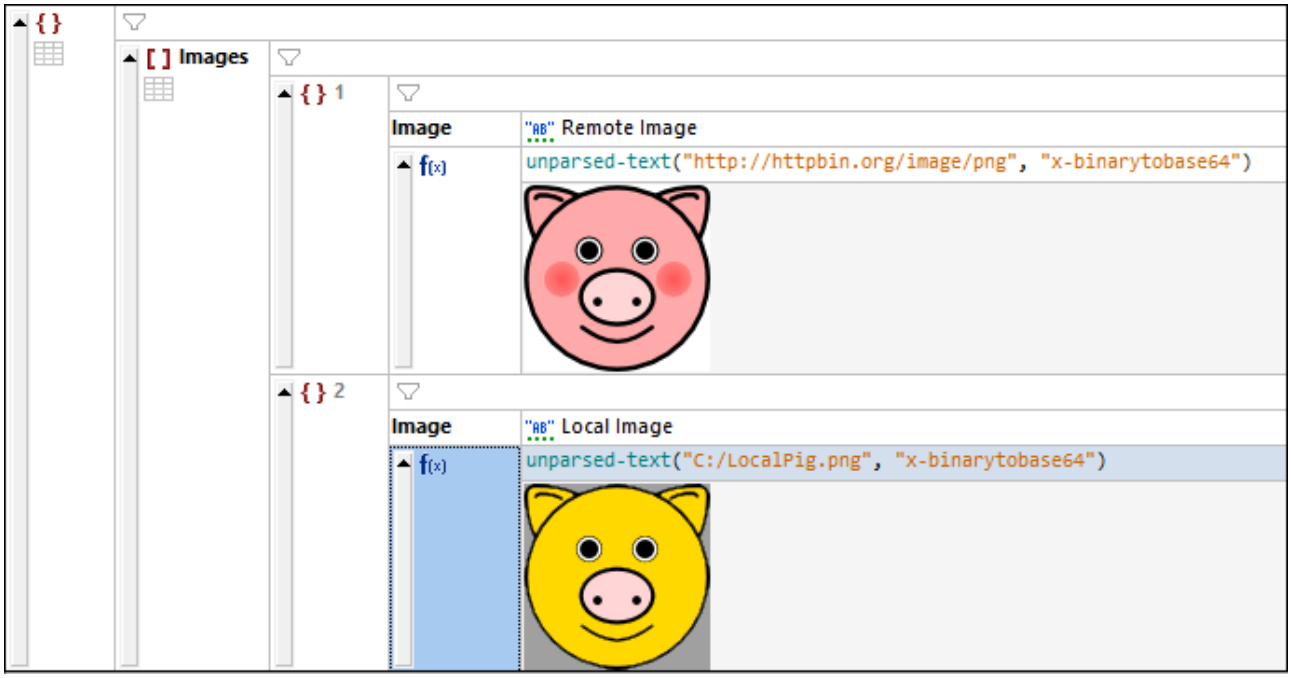
- Para añadir una columna nueva vacía a la tabla: Cambie de la vista Diseño de tabla a la Vista Lista, haga clic con el botón derecho en cualquier par `clave:valor` de la lista y anexe o inserte un nuevo par `clave:valor` desde el menú contextual del elemento. Al cambiar de vuelta a la vista Diseño de tabla se crea una columna nueva para el par `clave:valor` que anexó/insertó. Ahora puede editar esta columna en la vista Cuadrícula.
- Su todas las fórmulas de una columna de tabla son iguales que la que aparece en el encabezado y quiere crear una distinta para una sola celda, cambie a la Vista Lista y edite la fórmula de esa celda. Al cambiar a la vista Diseño de tabla las fórmulas aparecen cada una en su correspondiente celda.
- Para añadir una línea nueva en una expresión pulse **Ctrl+Entrar**.

13.5.8 Imágenes

Se pueden mostrar imágenes en la vista Cuadrícula JSON (*imagen siguiente*). La imagen se cifra automáticamente en Base 64.

Para cargar una imagen puede usar esta expresión: `unparsed-text("<Image-URL>", "x-binarytobase64")`

 © 2016-2022 Altova GmbH



La URL de las imágenes pueden tener estos formatos (también como rutas relativas):

- `http://httpbin.org/image/png`
- `file:///c:/LocalPig.png`
- `C:/LocalPig.png`
- `LocalPig.png`

Guardar una cadena cifrada en base 64 como imagen

Para guardar una cadena cifrada en base 64 en formato imagen haga clic con el botón derecho del ratón en la imagen o su celda y seleccione el comando **Guardar como imagen**. (Tenga en cuenta que la imagen está cifrada como una cadena en base 64 en el archivo activo pero aparece en la vista Cuadrícula como una imagen.) En el cuadro de diálogo que aparece, seleccione la ubicación donde quiere guardar la imagen y asigne un nombre al archivo de imagen. La extensión del archivo (`.png`, `.gif`, `.svg`, etc.) se detecta automáticamente en el cifrado en base 64 y aparece en el cuadro de diálogo **Guardar**. Haga clic en Guardar para terminar.

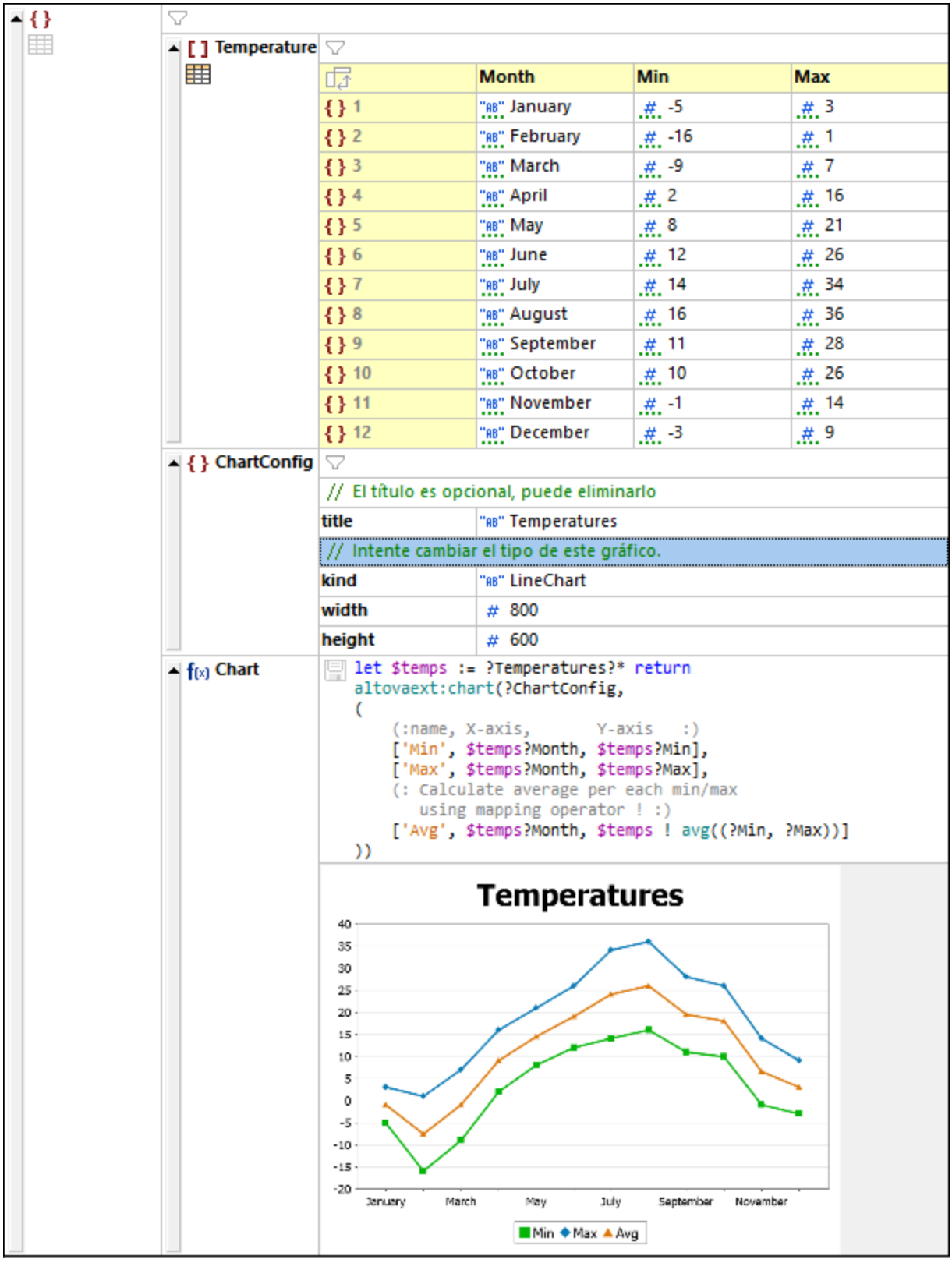
También puede ejecutar esta acción con el comando de menú **Edición | Guardar como imagen**.

13.5.9 Gráficos

Se pueden crear gráficos en la vista Cuadrícula JSON con la función de extensión XPath/XQuery de Altova **`altovaext:chart`** (*imagen siguiente*), que describimos más abajo. También viene explicada, junto con otras funciones de extensión para gráficos, en el apartado Funciones de extensión para gráficos 2200. La extensión **`altovaext:chart`** de la imagen siguiente se usa en una expresión XQuery `Let` definida dentro de una fórmula de la vista Cuadrícula JSON 697. El gráfico aparece como imagen bajo la fórmula. (Este gráfico de ejemplo está

en el archivo , que está en la carpeta *Examples* de su carpeta (Mis) Documentos[36] y a la que también puede acceder desde el *proyecto Examples*[122]).

© 2016-2022 Altova GmbH



{ }

[ ] Temperature

	Month	Min	Max
{ } 1	"AB" January	# -5	# 3
{ } 2	"AB" February	# -16	# 1
{ } 3	"AB" March	# -9	# 7
{ } 4	"AB" April	# 2	# 16
{ } 5	"AB" May	# 8	# 21
{ } 6	"AB" June	# 12	# 26
{ } 7	"AB" July	# 14	# 34
{ } 8	"AB" August	# 16	# 36
{ } 9	"AB" September	# 11	# 28
{ } 10	"AB" October	# 10	# 26
{ } 11	"AB" November	# -1	# 14
{ } 12	"AB" December	# -3	# 9

{ } ChartConfig

```
// El título es opcional, puede eliminarlo
title "AB" Temperatures
// Intente cambiar el tipo de este gráfico.
kind "AB" LineChart
width # 800
height # 600
```

f(x) Chart

```
let $temps := ?Temperatures?* return
altovaext:chart(?ChartConfig,
(
    (:name, X-axis,       Y-axis   :)
    ['Min', $temps?Month, $temps?Min],
    ['Max', $temps?Month, $temps?Max],
    (: Calculate average per each min/max
       using mapping operator ! :)
    ['Avg', $temps?Month, $temps ! avg((?Min, ?Max))]
))
```

Muestra de expresión XPath para generar el gráfico de la imagen anterior

```
let $temps := ?Temperatures?* return
altovaext:chart(?ChartConfig,
(
    (:name, X-axis,       Y-axis   :)
    ['Min', $temps?Month, $temps?Min],
    ['Max', $temps?Month, $temps?Max],
    (: Calculate average per each min/max
       using mapping operator ! :)
    ['Avg', $temps?Month, $temps ! avg((?Min, ?Max))]
))
```

Usar la extensión de gráficos de Altova

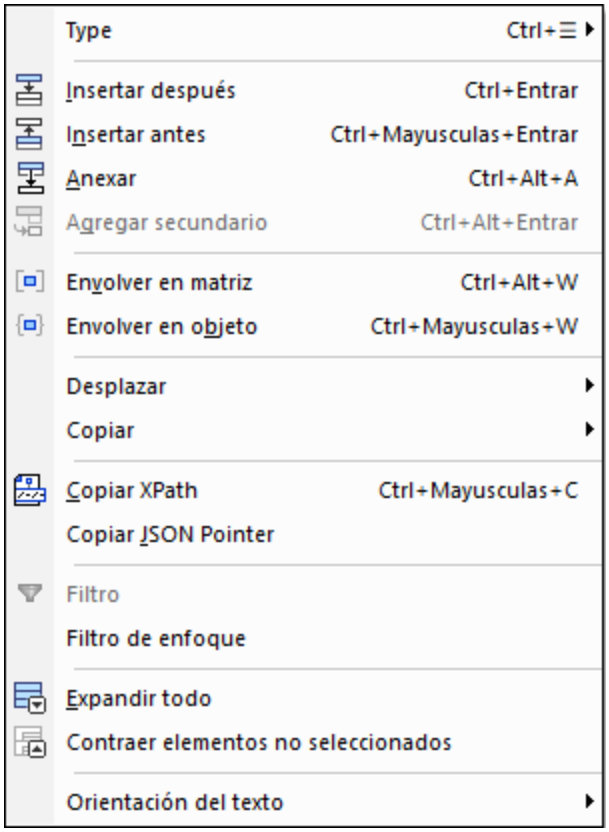
- la función de extensión para gráficos **altovaext:chart** debe usar el prefijo de espacio de nombres **altovaext:**.
- la función **altovaext:chart** toma dos argumentos: (i) información de configuración del gráfico y (ii) información sobre la serie de datos del gráfico.
- la información de configuración del gráfico es el primer argumento de **altovaext:chart**. Se trata de una serie sin ordenar de cuatro pares clave-valor. Estos pares son para (i) el título del gráfico, (ii) el tipo de gráfico, como barras, circulares, etc. (la clave es **kind**; consulte Funciones de extensión para gráficos 2200 para ver otros tipos), (iii) el ancho del gráfico en píxeles (solo números enteros; la clave es **width**) y (iv) el alto del gráfico en píxeles (solo números enteros; la clave es **height**). En la imagen anterior la información de la configuración se almacena en el objeto **ChartConfig**, al que se hace referencia desde la función **altovaext:chart**.
- la serie de datos del gráfico es segundo argumento de **altovaext:chart**. Cada serie de datos es una matriz de tamaño 3: (i) el nombre de la serie, (ii) los valores del eje x, (iii) los valores del eje y. Si quiere crear varias series (por ejemplo, que cada serie represente una línea, como en el ejemplo anterior) debe crear una secuencia de matrices. El ejemplo anterior tiene tres series de datos; para la temperatura mínima, máxima y media. Los datos de los ejes X e Y son referenciados desde las matrices **Temperatures**.

Para más información consulte Funciones de extensión para gráficos 2200.

13.5.10 Menú contextual de la vista Cuadrícula

Cuando hace clic con el botón derecho en un componente en la vista Cuadrícula JSON aparece un menú contextual (*imagen siguiente*) con comandos que puede usar para modificar la vista y editar el documento. También se puede acceder al menú contextual con la tecla Menú. Estos comandos se describen a continuación.

© 2016-2022 Altova GmbH



Tipo
Al pasar el cursor del ratón por encima del comando **Tipo** se abre un submenú en el que se puede seleccionar el tipo de componente[686]. También puede acceder al submenú Tipo pulsando las teclas **Ctrl+Tecla de menú**.

Insertar antes/después, anexar, agregar secundario
Los comandos **Insertar** y **Anexar** añaden un elemento al mismo nivel. El elemento nuevo se añade, respectivamente, antes (**Insertar antes**) o después (**Insertar después**) del elemento seleccionado, o como último elemento del mismo nivel del elemento seleccionado (**Anexar**). El comando **Agregar secundario** anexa un elemento nuevo como elemento secundario Si ya existen uno o más elementos secundarios, entonces el tipo del elemento nuevo será el mismo que el del último secundario. Si no existen otros secundarios, entonces el elemento nuevo será un par `vkey:value` vacío. Por ejemplo, si el elemento seleccionado es una matriz con secundarios que son objetos, entonces el elemento secundario nuevo se anexará como el último objeto secundario.

Envolver en matriz, Envolver en objeto
La parte seleccionada de la tabla se puede envolver tanto en una matriz como en un objeto.

Subir/Bajar/Desplazar a la izquierda/Desplazar a la derecha
Si se puede mover un componente en cualquiera de esas direcciones dentro de la cuadrícula, se habilita el comando correspondiente. Seleccione el que necesite para mover el componente.

Copiar
Estos comandos se pueden usar para copiar la selección en el portapapeles. Solo se habilitan las opciones que se pueden aplicar al componente en cuestión.

Comando	Descripción
Copiar como texto JSON	La selección actual se serializa como lenguaje de marcado JSON
Copiar como texto estructurado	La selección actual se serializa como TSV (valores separados por tabulador)
Copiar como imagen	La celda de imagen seleccionada se copia como imagen

Tenga en cuenta que:

- *Copiar como imagen* exporta una cadena cifrada en base 64 a un archivo de imagen. Debe asegurarse de que el nombre que asigna al archivo de imagen corresponde con el tipo de imagen indicado en el cifrado en base 64. Si la cadena que se exporta no está cifrada en base 64, entonces se creará un archivo con el nombre que haya indicado, pero este no tendrá contenido.
- Para insertar texto en una celda debe copiar el texto y pegarlo dentro de la celda.
- Para importar desde un archivo, por ejemplo una image, use el comando **Editar | Insertar | Archivo externo codificado**[1284]. Este comando inserta una imagen como cadena cifrada en base 64 y la muestra como imagen en la vista Cuadrícula JSON. Los formatos de imagen compatibles con la importación son: `PNG`, `JPEG`, `BMP`, `GIF`, `TIFF`.

Copiar XPath
Este comando copia en el portapapeles una expresión XPath 3.1 de ruta, empezando por el documento raíz, que ubica el nodo seleccionado.

Copiar JSON Pointer
Este comando copia en el portapapeles una expresión JSON Pointer que ubica el nodo seleccionado. Por ejemplo: **`/Artists/1/Albums/1/Tracks`**. Para más información **Copiar XPointer/Pointer JSON**[1283].

Filtro, Filtro de enfoque
El comando **Filtro** se puede activar y desactivar. Con él puede añadir un filtro al elemento que seleccione o desactivar un filtro que ya existe. Una vez que añada un filtro con este comando, recuerde que tiene que introducir una expresión de filtro o el filtro desaparecerá automáticamente. Una vez que se haya añadido un filtro puede desactivarlo volviendo a seleccionar este comando o haciendo clic en el filtro. Para eliminar el filtro haga clic en el comando *Eliminar todos los filtros* en la barra de herramientas.

El comando **Filtro de enfoque** (i) genera un filtro en el elemento raíz que contiene una expresión XPath para ubicar el nodo seleccionado y (ii) filtra los datos de la vista Cuadrícula para mostrar el nodo seleccionado (y sus descendientes) directamente bajo el elemento raíz. Esta opción es útil si quiere ver solamente el nodo seleccionado en la cuadrícula. Para quitar el filtro haga clic en el icono Filtro de la celda del elemento raíz.

Expandir todo
Este comando se habilita si el componente seleccionado o cualquiera de sus descendientes se contraen y sirve para expandir el componente y todos sus descendientes.

Contraer elementos no seleccionados
Este comando contrae todos los componentes excepto el componente seleccionado y sus antecesores.

Orden ascendente, orden descendente
Estos comandos se pueden usar para ordenar los componentes de una tabla según el contenido de la columna de tabla seleccionada. También puede ordenar una selección de filas de tablas, siempre que las filas sean adyacentes.

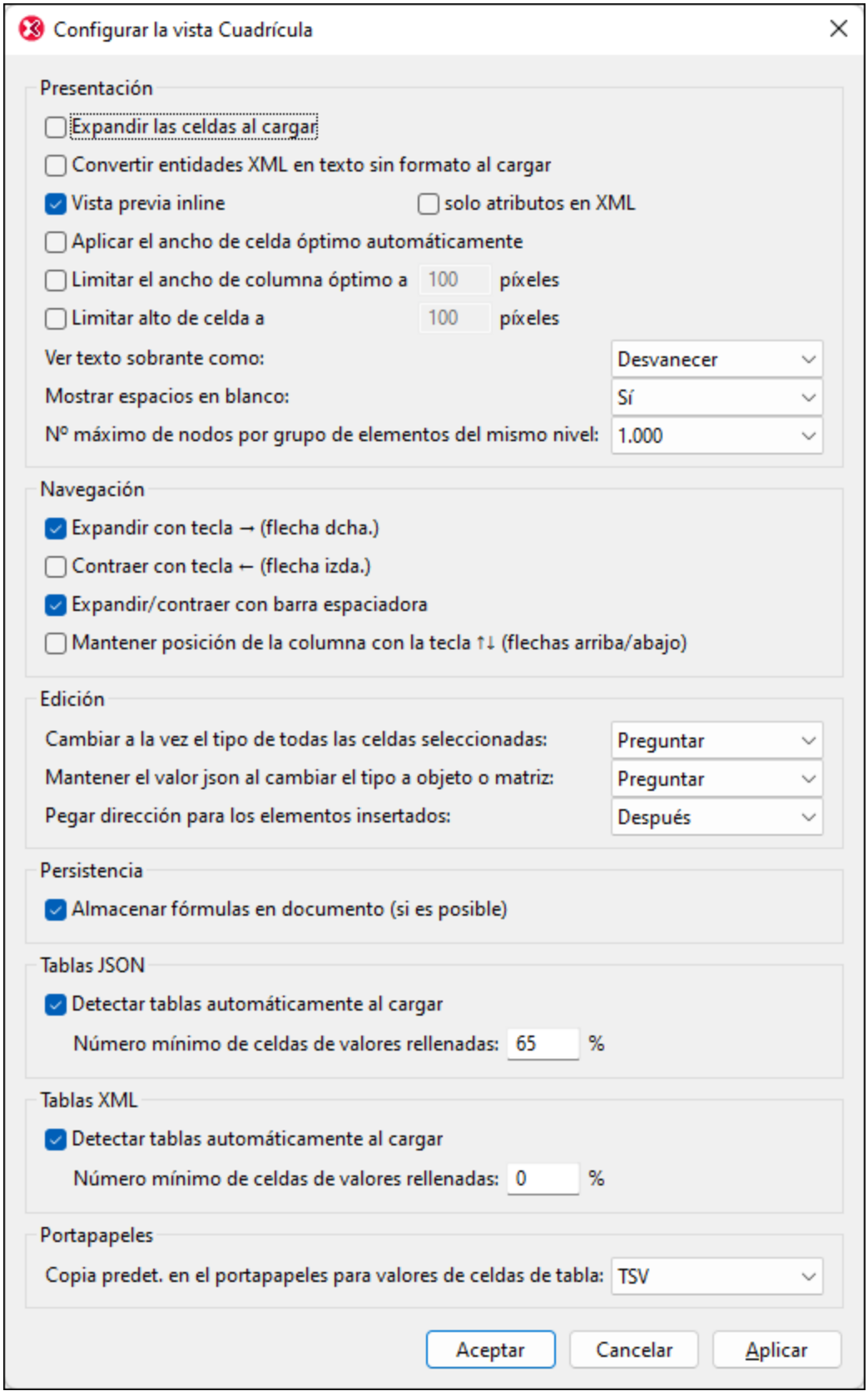
© 2016-2022 Altova GmbH

13.5.11 Configurar la vista Cuadrícula

Las opciones de la vista Cuadrícula se definen en el cuadro de diálogo Configurar la vista Cuadrícula (*imagen siguiente*), al que se accede desde el icono Configuración de la barra de herramientas de la vista Cuadrícula 681 .

	Icono Configuración de la barra de herramientas de la vista Cuadrícula 681

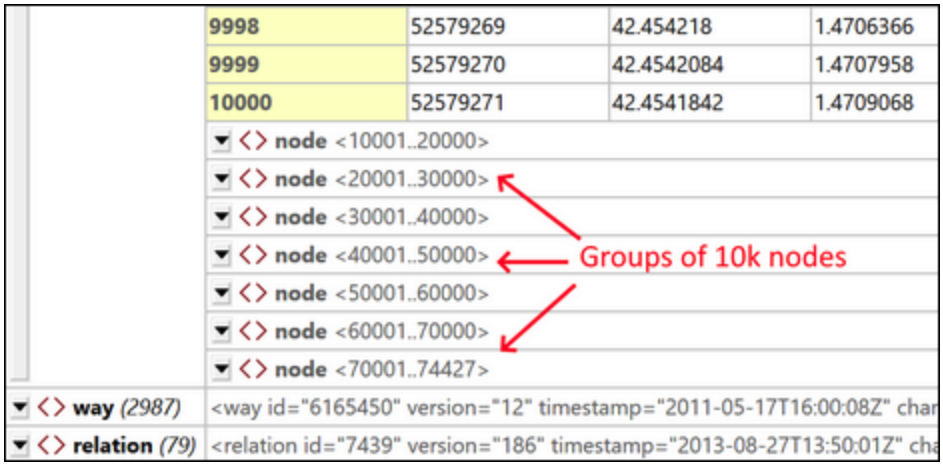
A continuación explicamos las opciones de configuración de la Vista Cuadrícula. Tenga en cuenta que estas opciones afectan a la vista Cuadrícula de todos los documentos (XML, JSON, DTD).



© 2016-2022 Altova GmbH

Presentación
Las opciones de la sección *Presentación* son bastante sencillas. No obstante, a continuación incluimos un par de aclaraciones.

- Si no se expanden todas las celdas al cargar, el nodo raíz y todos sus descendientes están contraídos. En este caso debe ir expandiendo los nodos conforme navega por el documento.
- Si se marca la opción *Convertir entidades XML en texto sin formato al cargar* las entidades XML se cargan en la vista Cuadrícula como texto sin formato de la entidad correspondiente; no se resuelven en sus respectivas representaciones de glifo.
- Si no se marca la opción *Vista previa inline*, en lugar de una vista previa de la celda solo aparece el número de índice del elemento de esa celda. Si se habilita esta opción, puede decidir si quiere que la vista previa contenga (i) el contenido y los atributos del elemento o (ii) solo los atributos. Si prefiere esta última opción, marque la casilla *solo atributos* en XML. Si quiere ver ambos, desmárquela. Recuerde que solamente se ve la primera parte del contenido inline de una celda; pase el cursor del ratón sobre la etiqueta de inicio para ver todo su contenido.
- Si activó el ancho de tabla óptimo podrá ver la cuadrícula entera. Para ello algunas celdas aplicarán el ajuste automático de línea.
- Si el texto es mayor que el tamaño de la celda puede elegir entre que el texto sobrante se desvanezca o sea reemplazado por puntos suspensivos.
- Puede mostrar u ocultar los espacios en blanco de las celdas. Los espacios aparecen como puntos a media altura y las tabulaciones como flechas. El final de línea se indica con un retorno de carro dentro de la celda.
- Los nodos del mismo nivel se pueden mostrar en grupos de 100, 1.000, 10.000 nodos (*imagen siguiente*). Esto es útil por dos motivos: (i) si quiere disponer de más espacio en la presentación de archivos que tienen un muchos nodos del mismo nivel; (ii) si solo quiere procesar un subconjunto de nodos del mismo nivel (porque ahorra tiempo), puede expandir los grupos de nodos del mismo nivel que quiere procesar y ejecutar la acción únicamente en esos nodos; por ejemplo si quiere usar la acción Buscar y reemplazar o si quiere convertir el tipo de un nodo (consulte *Edición*, más abajo).



Navegación
Ahora puede usar las flechas para navegar por la cuadrícula. A continuación explicamos las opciones que existen al usar estas teclas.

- *Expandir con la flecha derecha:* si un elemento de celda está contraído, la tecla *Flecha derecha* lo expande dentro de la celda. Si el elemento no está contraído, la tecla *Flecha derecha* pasa a la siguiente celda a la derecha (esto incluye elementos secundarios). Si no se activa esta opción, la tecla *Flecha derecha* se detiene en una celda contraída. La opción *Expandir con la flecha derecha* no

afecta a las celdas que están dentro de las tablas; en esas celdas la flecha derecha simplemente le llevará a la siguiente celda a la derecha.
- *Contraer con la flecha izquierda:* al moverse hacia la *izquierda* con esta tecla llega un momento en que tiene que moverse hacia arriba en la jerarquía del documento. Si se activa esta opción, lo elementos que se puedan contraer se contraen al pulsar la tecla *Flecha izquierda*; de lo contrario estos elementos no se contraen aunque la celda activa para al elemento superior. La opción *Expandir con la flecha izquierda* no afecta a las celdas que están dentro de las tablas; en esas celdas la flecha izquierda simplemente le llevará a la siguiente celda a la izquierda.
- *Expandir/contraer con la barra espaciadora:* la barra espaciadora sirve para expandir/contraer un elemento. Es decir, también puede usarla para navegar por la cuadrícula.
- *Mantener la posición de la columna con las teclas de flecha hacia arriba y abajo:* estas dos teclas suben y bajan por las celdas de la cuadrícula, incluidos los elementos secundarios y principales, que se encuentran en distintos niveles de la jerarquía y, por tanto, en distintas columnas. Si se selecciona esta opción puede saltarse los niveles que no estén en la columna activa. Esto puede ser útil, por ejemplo, si se encuentra en la columna del elemento **`subject/course/books/book/title`**. Al seleccionar esta opción puede usar las flechas arriba y abajo para navegar solamente por los títulos de los libros (sin entrar en las columnas book, books, course o subject y sin entrar en las columnas de elementos descendientes de **`Title`**.).

Edición
Las opciones de la sección *Presentación* son bastante sencillas. No obstante, a continuación incluimos un par de aclaraciones.

- Al cambiar el tipo de varias celdas seleccionadas puede escoger si llevar a cabo la acción: *Siempre*, *nunca* o *tras preguntar* (para que decida el usuario).
- Al cambiar un tipo JSON de atómico a objeto o matriz también puede escoger si la acción se lleva a cabo: (i) tras *Preguntar* (si el valor del tipo atómico se debe conservar como el valor de un par **`clave:valor`** secundario sin nombre), (ii) *Siempre* (se conserva el valor en un par **`clave:valor`** secundario sin nombre) o (iii) *Nunca* (nunca se debe conservar el valor).
- La opción *Pegar dirección* determina si la selección del portapapeles se pega encima o debajo de la celda seleccionada.

Persistencia
Los filtros y las fórmulas se pueden almacenar en los metadatos de documentos XML y JSON. Las fórmulas se almacenan como instrucciones de procesamiento en documentos XML y como comentarios en algunos tipos de documentos JSON. Estas opciones indican cómo guardar los filtros y las fórmulas. En el caso de las fórmulas, la terminología *cuando sea posible* hace referencia al hecho de que esas fórmulas no se pueden guardar como comentarios en todos los documentos JSON (solo en documentos JSON5 y JSONC).

Tablas JSON, tablas XML
Si se activa la opción que detecta tablas de la vista Cuadrícula JSON automáticamente al cargar, entonces puede seleccionar cuál es el porcentaje mínimo de celdas de tabla rellenadas para que una tabla se reconozca como tal. Si el número de celdas rellenadas no sobrepasa esta cantidad, la estructura aparece como una cuadrícula normal con los elementos extensibles uno bajo el otro en una lista.

Clipboard
También puede escoger si el contenido del portapapeles se debe restaurar como valores separados por comas (TSV) o como XML/JSON (en función del tipo de documento). Esta es una opción muy útil: Si quiere pegar un

 © 2016-2022 Altova GmbH

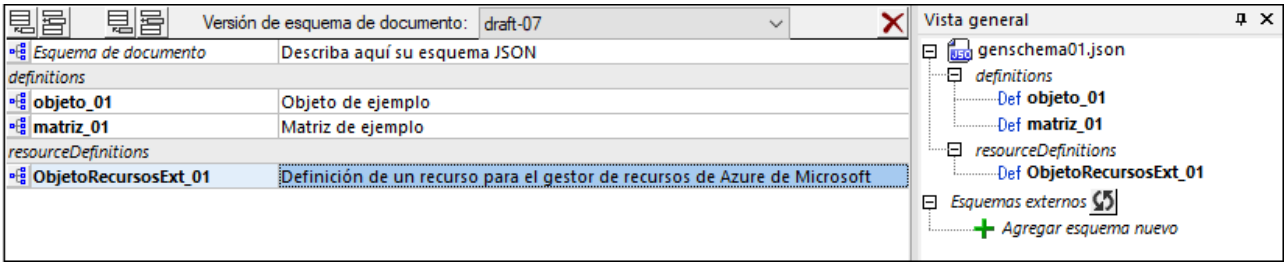
formulario de tabla desde el portapapeles en otro documento, esta opción permite elegir si la tabla copiada se guarda como TSV o con marcado. (Para ver la diferencia, intente pegar una tabla en un editor de texto después de copiarla en el portapapeles en los distintos formatos).

13.6 Vista Esquema JSON

En la vista Esquema JSON puede ver y editar documentos de esquema JSON. Estos son sus componentes:

- Una ventana principal que cuenta con dos modos: **Cuadrícula de definiciones**[717] y **Vista de diseño**[723].
- Tres ayudantes de entrada[718] (situados por defecto a la derecha de la ventana principal) llamados *Vista general*, *Detalles* y *Restricciones*.
- La ventana Mensajes (situada por defecto debajo de la ventana principal).
- La ventana Información[754] (situada por defecto en la esquina inferior izquierda de la ventana de la aplicación)

En esta imagen puede ver la ventana principal y el ayudante de entrada Vista general.

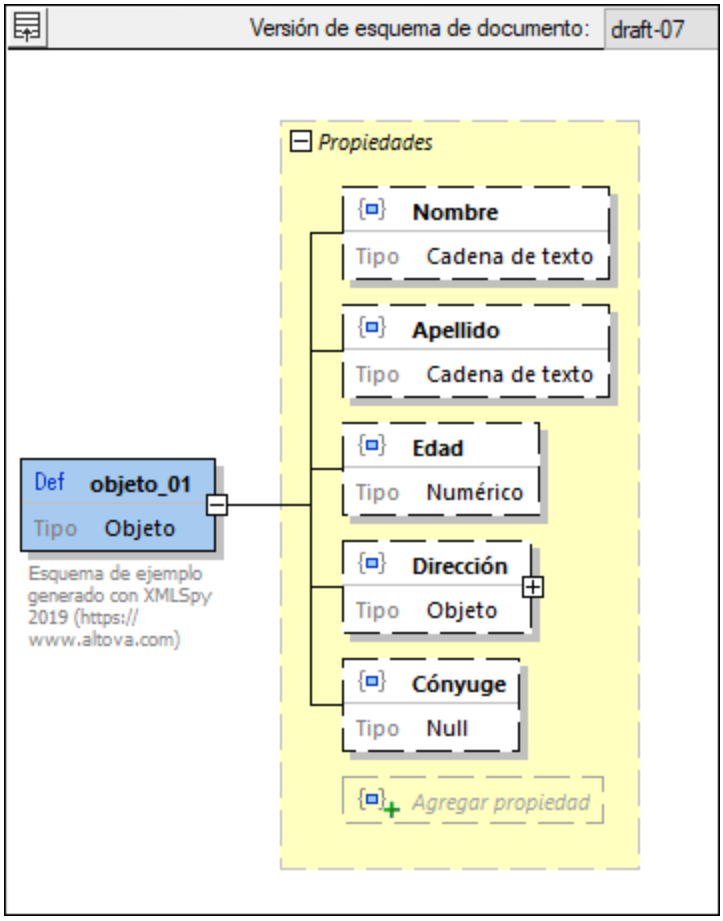


La ventana principal

La ventana principal tiene dos modos que se pueden alternar: la **Cuadrícula de definiciones**[717] (*imagen anterior*) y la **Vista de diseño**[723] (*imagen siguiente*). La **Cuadrícula de definiciones** muestra el esquema principal del documento actual (que recibe el nombre de `Esquema de documento`) y todas las definiciones que se añadieron al esquema. Una *definición* es una descripción de una estructura de datos JSON. En la imagen anterior, por ejemplo, `objeto_01` y `matriz_01` son definiciones de un objeto y de una matriz respectivamente. Las definiciones también aparecen en el ayudante de entrada Vista general.

La **Cuadrícula de definiciones** ofrece un resumen del esquema JSON pero no muestra el contenido de las definiciones. Para cambiar a la **Vista de diseño** y ver y editar una definición, haga clic en el icono de la definición o doble clic en la definición en el ayudante de entrada Vista general.

© 2016-2022 Altova GmbH



Para volver a la **Cuadrícula de definiciones** desde la **Vista de diseño** basta con hacer clic en el icono **Cambiar a Cuadrícula de definiciones** situado en la esquina superior izquierda de la vista de diseño (*imagen anterior*). Para configurar la **Vista de diseño** haga clic en el comando de menú **Diseño de esquemas | Configurar opciones de vista** 749.

Los ayudantes de entrada

Ambos modos de la Vista Esquema JSON (Cuadrícula de definiciones y Vista de diseño) ofrecen tres ayudantes de entrada llamados *Vista general*, *Detalles* y *Restricciones*. Estos ayudantes de entrada sirven para: (i) ver información sobre el esquema y sus definiciones y (ii) introducir información y valores relacionados con las definiciones. Consulte el apartado Ayudantes de entrada: Vista general, Detalles y Restricciones 718 para obtener más información sobre ellos.

13.6.1 Versión del esquema JSON

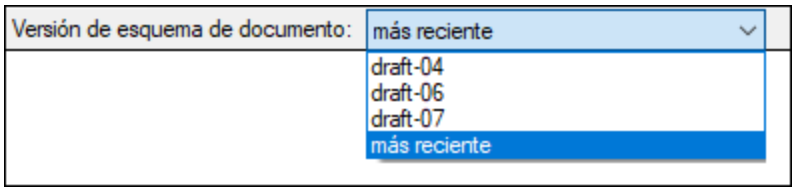
Un esquema JSON se escribe con sintaxis JSON y se suele guardar en un archivo con extensión `.json`. La palabra clave **`$schema`** indica que se trata de un esquema JSON y debe ser la primera palabra clave del objeto de nivel superior y tener como valor el URI de la versión del esquema JSON que quiera usar. Estos son dos ejemplos de cómo usar la palabra clave **`$schema`**:

```
{
   "$schema": "http://json-schema.org/draft-07/schema#",
   ...
}
```

```
{
   "$schema": "http://json-schema.org/schema#",
   ...
}
```

En el primero de los ejemplos anteriores la versión del esquema se menciona de forma explícita. Si falta el nombre de la versión, como en el segundo de los ejemplos anteriores, entonces eso indica que la versión del esquema que se debe usar es la más reciente (actualmente **2020-12 and 2019-09-hyper**).

En la vista JSON Schema puede cambiar a versión en el cuadro combinado de la barra que se encuentra en la parte superior de la ventana principal (*imagen siguiente*).



Opciones de las versiones nuevas del esquema que no existen en versiones anteriores

Si usa una opción de una de las versiones más recientes del esquema y después cambia a una versión anterior del mismo que no contiene esa opción, entonces ocurre lo siguiente:

- aparece un mensaje preguntando si quiere eliminar/convertir esa opción o mantenerla
- el componente que corresponda a la versión no compatible aparecerá resaltado en color naranja. Por ejemplo, si se define un valor para la palabra clave **const** (nueva en la versión **draft-06**) y cambia la versión del esquema a **draft-04**, entonces el valor de **const** aparecerá en naranja.

Versiones del esquema JSON

Para más información sobre las especificaciones del esquema JSON, en concreto sobre opciones específicas de cada versión, consulte estos enlaces:

Especificación JSON Schema: http://json-schema.org/specification.html
Notas sobre el borrador Draft-06: http://json-schema.org/draft-06/json-schema-release-notes.html
Notas sobre el borrador Draft-07: http://json-schema.org/draft-07/json-schema-release-notes.html
Borrador Draft 2019-09 (antiguo Draft-08): http://json-schema.org/specification-links.html#2019-09-formerly-known-as-draft-8
Borrador Draft 2020-12: http://json-schema.org/specification-links.html#2020-12

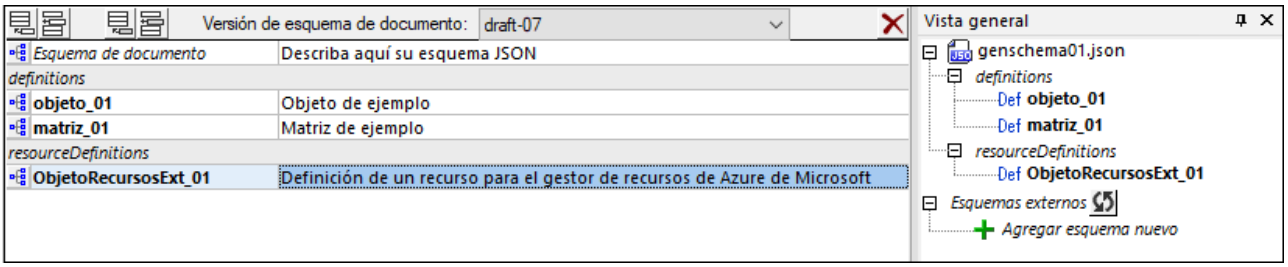
También están disponibles estos hiperesquemas: **draft-04-hyper**, **draft-06-hyper**, **draft-07-hyper**, **2019-09-hyper**.

© 2016-2022 Altova GmbH

Aquí puede encontrar enlaces a los esquemas nucleares y a los hiperesquemas: http://json-schema.org/specification-links.html.

13.6.2 Agregar definiciones globales

La **Cuadrícula de definiciones** 714 de la ventana principal (*imagen siguiente*) muestra una lista de las definiciones globales del esquema. Se trata (i) de la definición de *esquema de documento principal*, (ii) de las definiciones (o *$defs* en esquemas posteriores) de *tipos JSON globales,* como objetos, matrices, cadenas, etc. que son tipos de JSON Schema y (iii) de las definiciones de *tipos JSON externos o definidos de forma personalizada*. Por ahora solamente están disponibles las definiciones que tienen lugar dentro de un contenedor llamado `resourceDefinitions`. Se trata del contenedor que utiliza Microsoft Azure Resource Manager para definiciones JSON. Añada una sección *resourceDefinitions* nueva al documento de esquema con el icono **Anexar sección de definiciones** o **Insertar sección de definiciones** de la barra de herramientas de la cuadrícula (*imagen siguiente*).

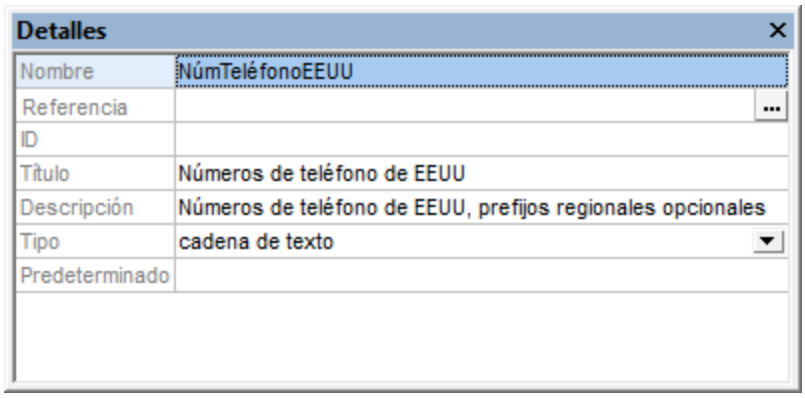


Se recomienda definir un tipo JSON 672 de forma global si necesita volver a utilizarlo dentro del mismo esquema o en otro esquema distinto. Por ejemplo, puede definir un tipo de cadena JSON para números de teléfono de EE UU en un esquema JSON (p. ej. una biblioteca de definiciones) y después hacerle referencia desde el mismo esquema o desde otros esquemas JSON.

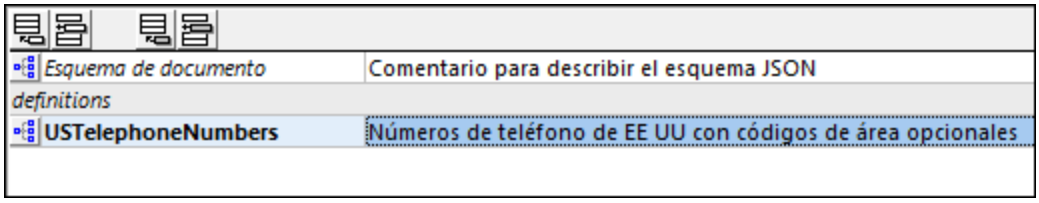
Agregar una definición

Estas son las acciones que se pueden llevar a cabo en la Cuadrícula de definiciones:

- *Para agregar una definición:* haga clic en el icono **Anexar definición de esquema con nombre** o **Insertar definición de esquema con nombre** (*situados en la esquina superior izquierda de la cuadrícula*). En la cuadrícula se crea una definición nueva vacía, con un nombre predeterminado. La nueva definición también aparecerá en el ayudante de entrada Vista general.
- *Para cambiar el tipo de una definición:* todas las definiciones nuevas se crean con el tipo `Cualquiera`. Puede cambiar el tipo en el ayudante de entrada Detalles (*imagen siguiente*) o editando la definición en la Vista de diseño 723.



- *Para cambiar el nombre de una definición:* haga doble clic en el nombre y edítelo. Otra manera es editar el campo *Nombre*en el ayudante de entrada Detalles[719].
- *Para insertar una descripción de la definición:* edite el campo *Descripción* en el ayudante de entrada Detalles[719]. La descripción aparece en la Cuadrícula de definiciones junto al nombre de la definición (*imagen siguiente*). Otra manera es hacer doble clic en el campo Definición de la cuadrícula.



- *Para hacer referencia a una definición:* consulta la descripción del ayudante de entrada Vista general[718] y la sección Definiciones globales[721]y locales.
- *Para editar una definición:* haga clic en el icono de la definición en la Cuadrícula de definiciones o doble clic en la definición en el ayudante de entrada Vista general[718]. Esto abre la definición en la Vista de diseño.

13.6.3 Ayudantes de entrada: Vista general, Detalles y Restricciones

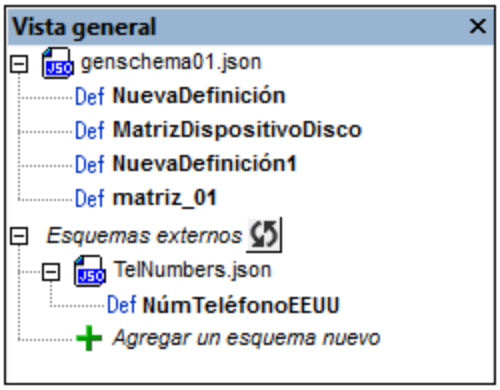
Los ayudantes de entrada de la vista Esquema JSON están situados por defecto en el lateral derecho de la ventana de la aplicación. Los ayudantes están disponibles en ambos modos de la ventana principal[714]: (i) Cuadrícula de definiciones y (ii) Vista de diseño. Puede cambiar la posición de los ayudantes de entrada con solo hacer clic en su barra de título y arrastrarlos por la ventana de la aplicación. Haga doble clic en su barra de título para acoplar el ayudante de entrada o para hacerlo flotar. Consulte el apartado Ayudantes de entrada[124] para obtener más información.

Vista general

El ayudante de entrada Vista general (imagen siguiente) enumera la definición del esquema actual y todas sus definiciones globales. Si hace doble clic en una definición, ésta se abre en la Vista de diseño[723], donde se puede editar. Si desea utilizar definiciones de esquemas externos, basta con añadir el esquema externo y después reutilizar la definición correspondiente.

© 2016-2022 Altova GmbH

Agregar esquemas externos
Para añadir un esquema externo haga clic en el icono **Agregar esquema nuevo** del ayudante de entrada y busque el esquema en el sistema de archivos. Las definiciones del esquema recién añadido aparecerán en el ayudante de entrada. Por ejemplo, en la imagen siguiente puede ver que se añadió el esquema `TelNumbers.json` y que este esquema tiene una definición llamada **`NúmTeléfonoEEUU`**. Puede añadir tantos esquemas externos como desee.



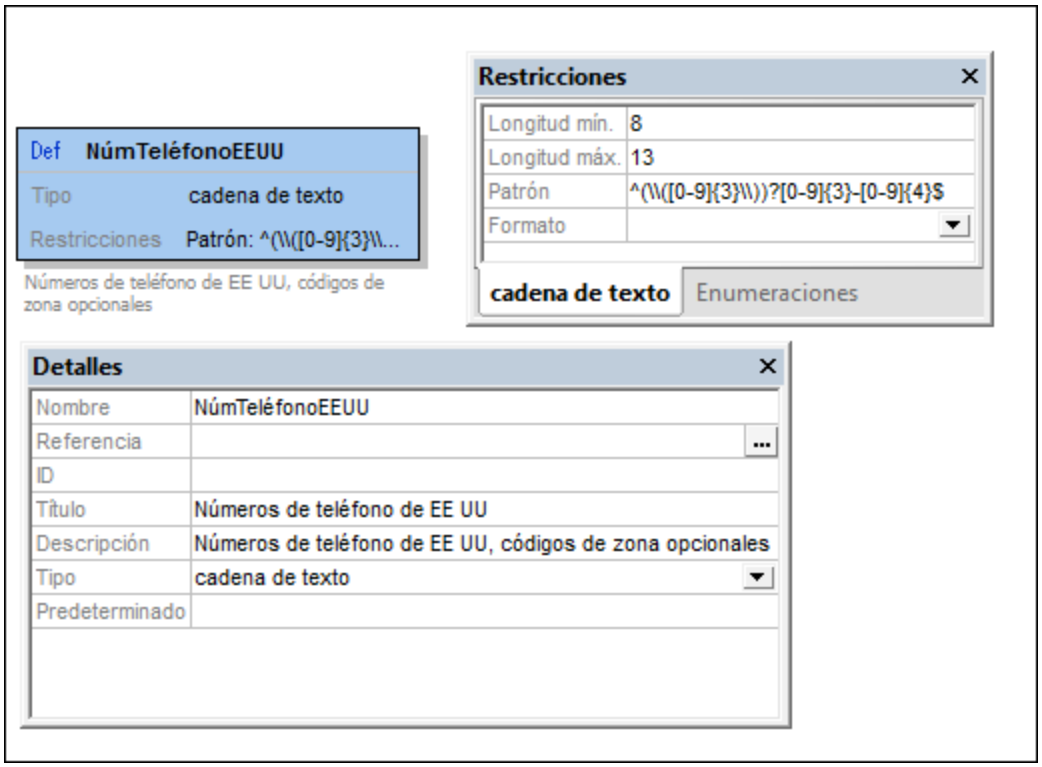
Reutilizar definiciones externas
Una vez añadido el esquema externo, sus definiciones se podrán reutilizar en las definiciones del esquema al que se importaron. Cuando una definición reutiliza otra definición (por medio de referencias), toma todas sus propiedades. La referencia a otra definición puede hacerse de dos maneras:

- *en el modo Vista de diseño:* arrastrando la definición desde el ayudante de entrada Vista general hasta la definición donde se desea utilizar.
- *en los modos Cuadrícula de definiciones y Vista de diseño:* desde el campo Referencia del ayudante de entrada Detalles de la definición donde se desea utilizar. Para más información consulte la descripción del ayudante de entrada Detalles 719.

Nota: el icono **Actualizar** situado junto a la entrada *`Esquemas externos`* del ayudante Vista general sirve para actualizar todos los esquemas externos añadidos. Recuerde que si no reutiliza ninguna definición de algún esquema externo, este esquema desaparecerá de la lista cuando haga clic en **Actualizar**.

Detalles

Las propiedades de una definición se pueden introducir en los ayudantes Detalles y Restricciones cuando se selecciona la definición en ambos modos de la vista Cuadrícula de definiciones o Vista de diseño 714. En la imagen siguiente puede ver la definición de `NúmTeléfonoEEUU` en la Vista de diseño 723 junto con los ayudantes de entrada Detalles y Restricciones. Observe que la información de los dos ayudantes de entrada también aparece en el recuadro azul de la definición en la Vista de diseño. Más abajo puede ver qué propiedades pueden configurarse en estos dos ayudantes de entrada.



Estas son las propiedades que se pueden editar en el ayudante de entrada Detalles:

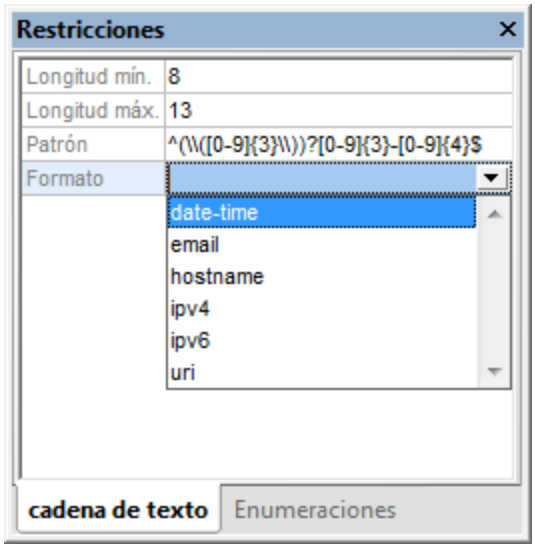
- *Nombre:* el nombre de la definición.
- *Referencia:* si desea que una definición reutilice otra definición, haga clic en el botón **Examinar** del campo Referencia (*imagen anterior*). Esto abre el cuadro de diálogo "Editar referencia" donde se enumeran todas las definiciones disponibles (tanto del esquema actual como de esquemas externos). Seleccione qué definición desea reutilizar, marque la opción *Convertir ruta de la referencia en relativa* si quiere usar una ruta relativa y haga clic en **Aceptar**. Para más detalles consulte Definiciones globales y locales 721.
- *Tipo:* seleccione el tipo de datos de la definición en la lista desplegable del cuadro combinado. Tenga en cuenta que si cambia el tipo de la definición se eliminarán las palabras clave que sean propias del tipo anterior. Si desea volver a las definiciones anteriores, pulse las teclas **Deshacer (Ctrl-Z)**. Los tipos se explican en los apartados Datos JSON 669 y Selectores de tipo (cualquiera, múltiple, etc.) 740.
- *ID:* se trata de una palabra clave opcional que sirve para definir un URI para el esquema. Este URI se puede usar para hacer referencia al esquema y se usa como URI de base para otras referencias de URI dentro del esquema. El valor de esta propiedad debe ser una cadena de texto que sea un URI. Recuerde que en el validador JSON de Altova el desreferenciado es canónico solamente. Consulte la especificación JSON para obtener más información.
- *Anclaje (nuevo en draft-2019-09):* se trata de un elemento clave de identificación opcional que indica un fragmento de nombre simple (y no un URI, como ocurre con ID). El valor de Anclaje debe ser una cadena de texto tal y como se describe en el borrador correspondiente.
- *Título y Descripción:* los valores de estas dos palabras claves sirven para describir las definiciones dirigidas a los usuarios finales.
- *Comentario (nuevo en draft-07):* destinado a los programadores del esquema, al contrario que la descripción, que está destinada a los usuarios finales.
- *Const (nuevo en draft-06):* un valor constante, como una enumeración de un valor.
- *Predeterminado:* el valor predeterminado de la definición.

© 2016-2022 Altova GmbH

- *De solo lectura, De solo escritura (nuevo en draft-07):* indican, respectivamente, campos que son de solo lectura o de solo escritura. Un ejemplo de un campo de solo escritura sería el campo en el que se teclea la contraseña.
- *Obsoleto (nuevo en draft-2019-09):* un indicador de que la definición seleccionada se va a eliminar en un futuro. Las aplicaciones tienen métodos especiales para tratar con este tipo de definiciones.

Restricciones

Las restricciones de una definición dependen del tipo que tenga. A continuación describimos las restricciones de cada tipo. (Consulte también el apartado sobre Tipos atómicos 738.)



Los tipos que faltan en esta lista se han excluido porque no pueden tener restricciones. Sin embargo, recuerde que puede definir enumeraciones para todos los tipos:

- *Cadena de texto:* la longitud de la cadena y su patrón. El patrón se especifica por medio de una expresión regular. En el campo `Formato` puede seleccionar uno de los formatos de cadena de texto que se definen en la especificación JSON (*imagen anterior, que muestra los formatos disponibles en la versión draft-04*); existen más formatos definidos en versiones posteriores. *Tipo de contenido* y *Codificación del contenido* (ambos nuevos en draft-07) seleccionan el tipo de soporte y de codificación de los datos no JSON codificados en cadenas JSON.).
- *Numérico:* el rango de los valores permitidos.
- *Matriz:* el número de elementos permitidos en la matriz y si estos deben ser únicos.
- *Objeto*: el número de propiedades permitidas.

El ayudante de entrada Restricciones siempre incluye la pestaña enumeraciones. En ella puede especificar qué elementos están permitidos para ese tipo de definición. Asimismo, existe una pestaña *Ejemplos* (nueva en draft-06) para todos los tipos excepto *Prohibido*. Se trata de una serie de ejemplos sin validación; el valor de `default` se puede usar como ejemplo sin repetirlo bajo esta palabra clave.

13.6.4 Definiciones globales y locales

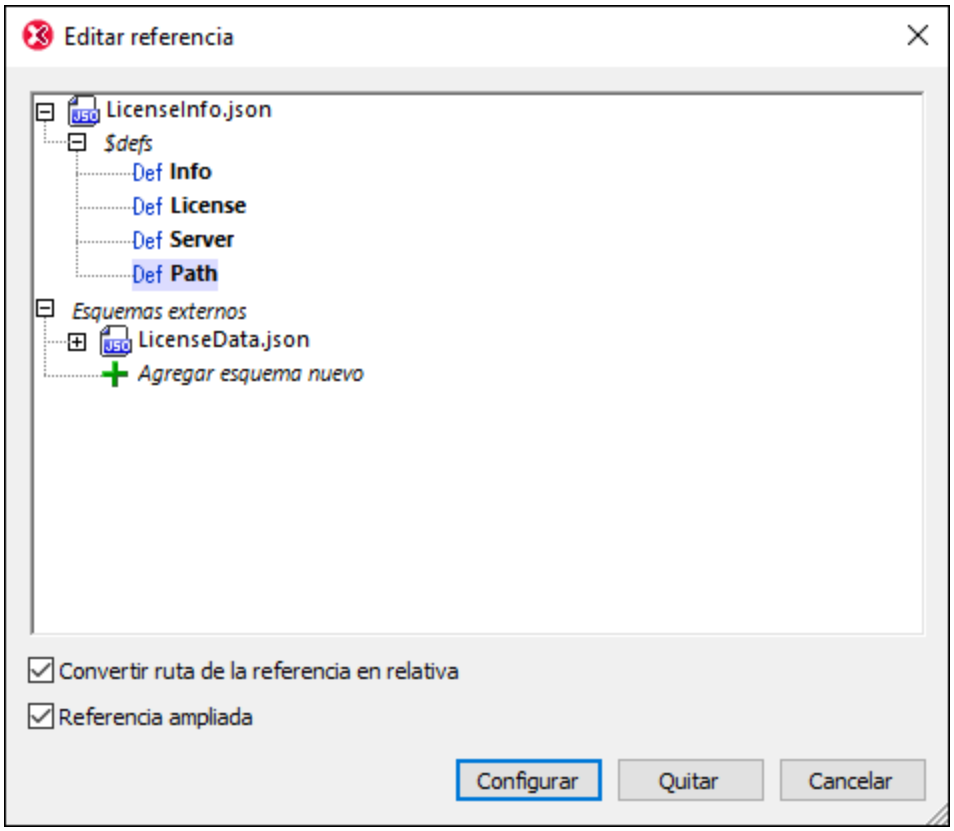
Las definiciones del esquema JSON se pueden crear global o localmente:

- Las **definiciones globales** se crean en la Cuadrícula de definiciones[714] de la ventana principal añadiendo una definición y especificando sus propiedades[717]. A una definición global se puede hacer referencia desde otras definiciones del mismo esquema o desde definiciones de otros esquemas. Esto permite reutilizar definiciones en todo el proyecto. Todas las definiciones globales del esquema actual aparecen en la Cuadrícula de definiciones[714]. Para poder reutilizar definiciones globales de otros esquemas basta con añadir los esquemas externos[718] al ayudante de entrada Vista general.
- Las **definiciones locales** se crean dentro de definiciones globales. Esto se hace añadiendo definiciones de un nivel inferior o del mismo nivel a la definición global.

Referencias a una definición global

Si quiere hacer referencia a una definición global desde otra definición tiene dos opciones:

- En la vista Diseño arrastre la definición global desde el ayudante de entrada Vista general[718] hasta la definición donde quiere usarla.
- En la vista Diseño[714] haga clic con el botón derecho en la definición para la que quiere hacer referencia a una definición global y seleccione **Editar referencia**. (También puede, habiendo seleccionado la definición en la vista Diseño, ir al ayudante de entrada Detalles y hacer clic en el botón **Otro cuadro de diálogo** del campo *Referencia*.) En el cuadro de diálogo *Editar referencia* que aparece (*imagen siguiente*) seleccione la referencia global que necesita. Otra manera de acceder a este cuadro de diálogo es seleccionando la definición de referencia, yendo al campo Referencia del ayudante de entrada Detalles y haciendo clic en el botón **Otro cuadro de diálogo** del campo y seleccione la definición global que quiere reutilizar (*véase la descripción del Ayudante de entrada Detalles*).



 © 2016-2022 Altova GmbH

Nota: una definición puede (i) hacer referencia a una definición global y no contener ninguna definición local o restricción local o (ii) hacer referencia a una definición global y también contener definiciones/restricciones locales (de las versiones *draft-2019-09* y posteriores). En el segundo caso, la referencia se conoce como **referencia ampliada**. En el cuadro de diálogo Editar referencia puede crear una referencia a una definición global como referencia ampliada si marca la casilla *Referencia ampliada*. Las definiciones globales que se crean como referencias ampliadas siempre aparecen como último elemento en la lista de definiciones de componentes de referencia. Si se ha definido un valor para la palabra clave `description` del componente de referencia, ese valor aparece bajo el componente de referencia.

Nota: si cambia el nombre de una definición global después de hacerle referencia desde otra definición del mismo esquema, el nombre también cambiará en la referencia. Sin embargo, las referencias desde otros esquemas deberán editarse manualmente para que sean correctas.

Convertir definiciones locales en globales

Haga clic con el botón derecho en una definición local en la Vista de diseño[723] y seleccione **Convertir en global**. Esto crea una definición global e inserta una referencia a ella en la definición local. El nombre de la definición global se genera de forma automática, pero si lo edita el cambio se pasa automáticamente a la referencia de la definición local.

Cambiar referencias a una definición global por una definición local

Una referencia a una definición global puede darse tanto en definiciones locales como globales. Para quitar la referencia y hacer que sus propiedades sean locales, haga clic con el botón derecho en la definición (local o global) en la Vista de diseño[723] y seleccione **Convertir en local**. Las propiedades de la definición global se crean de forma local en la definición.

13.6.5 Vista de diseño

En el modo Vista de diseño puede especificar la estructura y los valores permitidos de cada definición global del esquema. Las definiciones se especifican con ayuda de estos mecanismos o componentes de la interfaz gráfica:

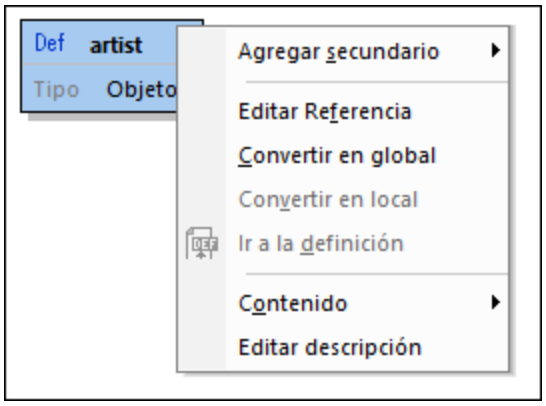
- el ayudante de entrada Detalles[719] (también disponible en la Cuadrícula de definiciones)
- el ayudante de entrada Restricciones[721] (también disponible en la Cuadrícula de definiciones)
- el menú contextual de la definición (que se abre haciendo clic con el botón derecho en el recuadro de la definición)

Las definiciones que se pueden especificar con los ayudantes de entrada Detalles y Restricciones se describen en detalle en el apartado Ayudantes de entrada: Vista general, Detalles y Restricciones[718]. Algunas de estas propiedades también se pueden especificar dentro del recuadro de la definición directamente. En este apartado (y en los tres siguientes) describimos el mecanismo gráfico que ofrece la ventana principal.

Nota: si necesita deshacer cambios realizados por error o no deseados basta con pulsar **Ctrl+Z**.

Menú contextual

El menú contextual de una definición (*imagen siguiente*) ofrece comandos para diseñar su estructura y editar sus propiedades.

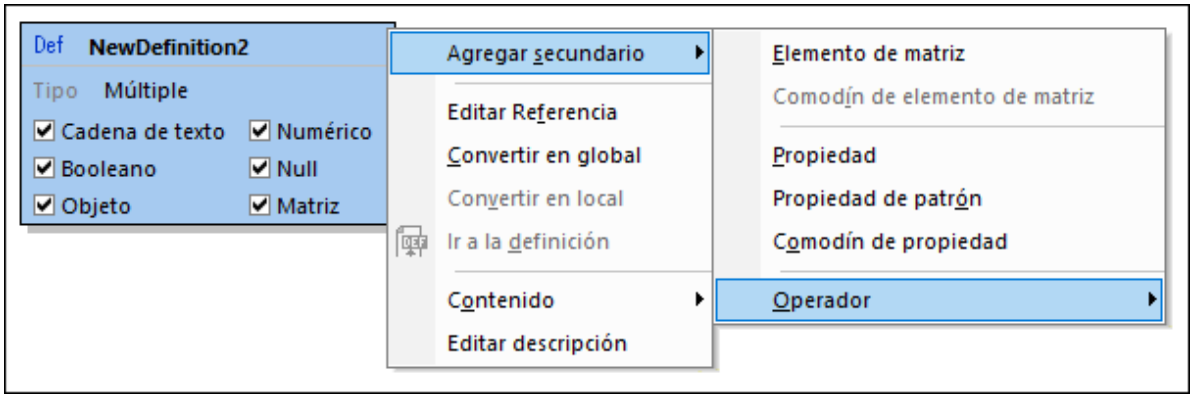


Estos son los comandos que ofrece el menú contextual:

- **Agregar secundario:** dependiendo del tipo de definición se pueden agregar unos secundarios u otros (consulte el apartado *Agregar secundarios: crear la estructura* 724).
- **Editar referencia:** permite que la definición haga referencia a una definición global y que tome las propiedades de dicha definición global. Este comando abre el mismo cuadro de diálogo "Editar la referencia" que se abre desde el ayudante de entrada Detalles 719 (consulte el apartado Ayudantes de entrada: Vista general, Detalles y Restricciones 719).
- **Convertir en global:** este comando se habilita si la definición es local 721. Convierte la definición seleccionada en una definición global 721 y añade una referencia a dicha definición global en la selección.
- **Convertir en local:** este comando se habilita si la definición es global 721. Convierte la definición seleccionada en una definición local 721 creando una referencia a la definición global original.
- **Ir al a definición:** este comando se habilita si la definición seleccionada está dentro de una definición que hace referencia a una definición global. Al hacer clic en este comando la vista le lleva a la definición global.
- **Contenido:** este comando abre un submenú con comandos para cortar, copiar y restaurar el contenido de la definición seleccionada.
- **Editar descripción:** habilita el campo `Descripción` de la definición para poder editarlo.

Agregar secundarios: crear la estructura

La estructura de una definición se crea añadiendo varios niveles de descendientes. Estos niveles se crean con el comando **Agregar secundario** del menú contextual. Los secundarios que se pueden agregar dependen del tipo que tenga la definición:



 © 2016-2022 Altova GmbH

- *Objeto*[725]*:* los objetos toman propiedades y operadores.
- *Matriz*[736]*:* las matrices toman elementos de matriz y operadores.
- *Tipo atómico (cadena de texto, numérico, booleano y null)*[738]*:* toman operadores.
- *Cualquiera*[740]*:* toma propiedades, elementos de matriz y operadores.
- *Múltiple*[740]*:* dependiendo del tipo que incluya toma todos los secundarios permitidos para los tipos seleccionados.
- *Operador*[746]*:* permite utilizar operadores lógicos para determinar la estructura de la definición.

Las estructuras que pueden crearse para cada tipo se describen en detalle en los apartados siguientes.

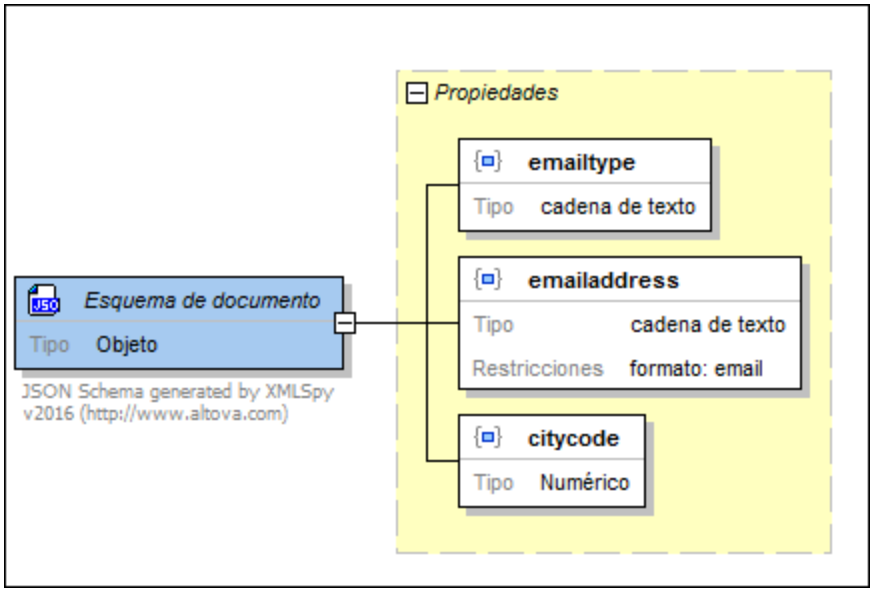
13.6.6 Objetos y propiedades

Los objetos se delimitan con llaves y asignan una clave a un valor, por ejemplo: `"MiClave": Valor`. La clave debe ser siempre una cadena de texto y, por tanto, debe ir entre comillas. El valor puede ser cualquier tipo de datos JSON[669]. Cada par `clave:valor` se conoce como propiedad del objeto (*imagen siguiente*).

Aquí puede ver un objeto con instancias que tiene tres propiedades:

```
{
  "emailtype": "home",
  "emailaddress": "contact01.home@altova.com",
  "citycode": 22
}
```

El esquema para este objeto sería más o menos este:



Debe tener en cuenta estos aspectos:

- Cada propiedad debe estar presente en la instancia. Si es así, la propiedad tiene un borde sólido. Si una propiedad es opcional, el borde es una línea discontinua. Puede definir si la propiedad es

obligatoria u opcional con los comandos del menú contextual de la propiedad o desde el ayudante de entrada Detalles.
- El orden en que deben aparecer las propiedades en la instancia no se puede definir en el esquema. Esto significa que el orden en el que se definen las propiedades en el esquema es totalmente irrelevante.
- El símbolo con un cuadrado azul entre llaves significa que se trata de una propiedad (en lugar de una propiedad de patrón o comodín de propiedad, que tienen *otros símbolos* 726).
- El tipo de una propiedad se puede editar haciendo doble clic en el tipo en el diagrama de la Vista de diseño y seleccionando una opción en la lista desplegable. También puede seleccionar el tipo en el ayudante de entrada Detalles.
- El valor de restricción de la propiedad `emailaddress` se define en el ayudante de entrada Restricciones.

Propiedades, propiedades de patrón, comodines de propiedad y esquemas de nombre de propiedad

Un objeto puede tener propiedades, propiedades de patrón, comodines de propiedad y esquemas de nombre de propiedad (nuevos en el draft-07 715). Estos elementos se añaden al objeto desde los menús contextuales (i) del objeto, (ii) del recuadro amarillo de la propiedad (haciendo clic con el botón derecho en el título del recuadro) y (iii) de cada propiedad. Ya hemos descrito los aspectos más importantes de las propiedades. Ahora vamos a describir las propiedades de patrón y los comodines de propiedad.

Propiedad de patrón

Una **propiedad de patrón** (*imagen siguiente*) define el nombre de la propiedad con una expresión regular. En la imagen siguiente, por ejemplo, la expresión regular indica que la propiedad (i) debe tener un nombre que empiece con un guión bajo y (ii) debe tener como valor un booleano. Las propiedades de patrón no necesitan restricciones y puede añadir tantas propiedades de patrón como quiera. Observe que el símbolo de las propiedades de patrón es {^$}.



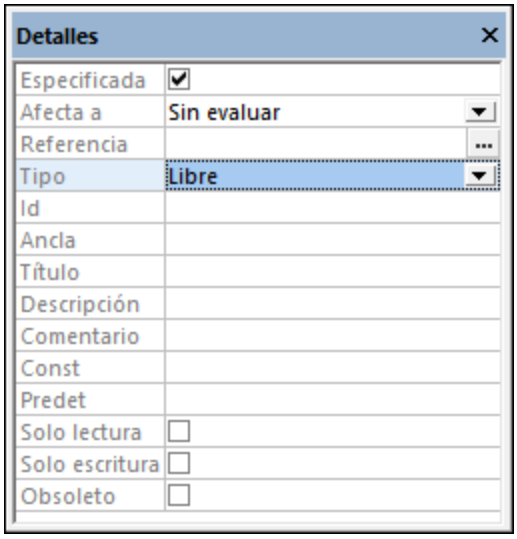
Comodín de propiedad

Un **comodín de propiedad** (*imagen siguiente*) especifica que además de las propiedades del conjunto de propiedades del objeto pueden darse además un número concreto de propiedades. El comodín puede definir el tipo que deben tener estas propiedades. Por ejemplo, en la imagen siguiente, el comodín de propiedad define que las propiedades pueden tener cualquier nombre pero deben tener valores numéricos. Por cada objeto solo puede haber un comodín de propiedad. Sin embargo, si el comodín tiene el tipo *Cualquiera*, entonces podrá definir restricciones para cada tipo en el ayudante de entrada Restricciones. Observe que el símbolo de los comodines de propiedad es {*}.



A partir del borrador draft-2019-09 los comodines de propiedad tienen una palabra nueva, `unevaluatedProperties`, que se procesa solamente si falta la palabra clave `additionalProperties`. Los valores de estas dos palabras clave se producen seleccionando los valores apropiados para las entradas *Especificado*, *Se aplica a* y *Tipo* en el ayudante de entradas Detalles (*imagen siguiente*).

 © 2016-2022 Altova GmbH



En la tabla siguiente puede ver qué efectos tienen las palabras clave `unevaluatedProperties` y `additionalProperties` (y viceversa, es decir, el efecto que la palabra clave tiene en los valores del ayudante de entrada del editor). La configuración de la imagen anterior, por ejemplo, sería `unevaluatedProperties=true`.

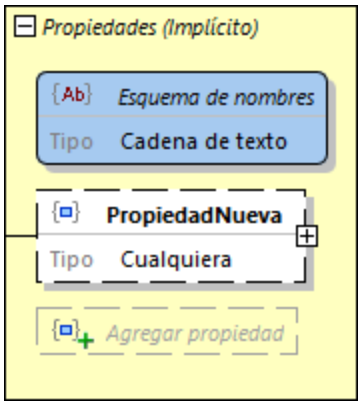
addtionalProperties	unevaluatedProperties		Especificado	Se aplica a	Tipo
--	--	<=>	false	Todos	Libre
true	omitido	<=>	true	Todos	Libre
false	omitido	<=>	--	--	--
Schema	omitido	<=>	true	Todos	Tipo de esquema
--	true	<=>	true	Sin evaluar	Libre
--	false	<=>	true	Sin evaluar	Prohibido
--	Schema	<=>	true	Sin evaluar	Libre

Tenga en cuenta que:

- Si `addtionalProperties` y `unevaluatedProperties` están presentes, `unevaluatedProperties` se omite.
- *Especificado*=false solo funciona con *Alcance*=Todos y *Tipo*=Libre.

Esquema de nombre de propiedad

Un **esquema de nombre de propiedad** (*imagen siguiente*) limita los nombres de las propiedades de ese un objeto en cuestión. (Esta opción es nueva en draft-07 715). Por ejemplo, en la imagen siguiente podemos ver que los nombres de las propiedades deben ser cadenas de texto. También podemos indicar otras limitaciones para esta propiedad con el ayudante de entrada Restricciones: por ejemplo, que el nombre de la propiedad esté dentro de cierta longitud de caracteres o que siga un patrón en concreto.



Nota: no se puede configurar un mínimo ni un máximo de repeticiones para las propiedades de patrón ni para los comodines de propiedad. Consulte el párrafo siguiente para obtener más información.

Cómo se validan las propiedades

Cuando en una instancia se encuentra una propiedad, la aplicación la valida de esta manera:

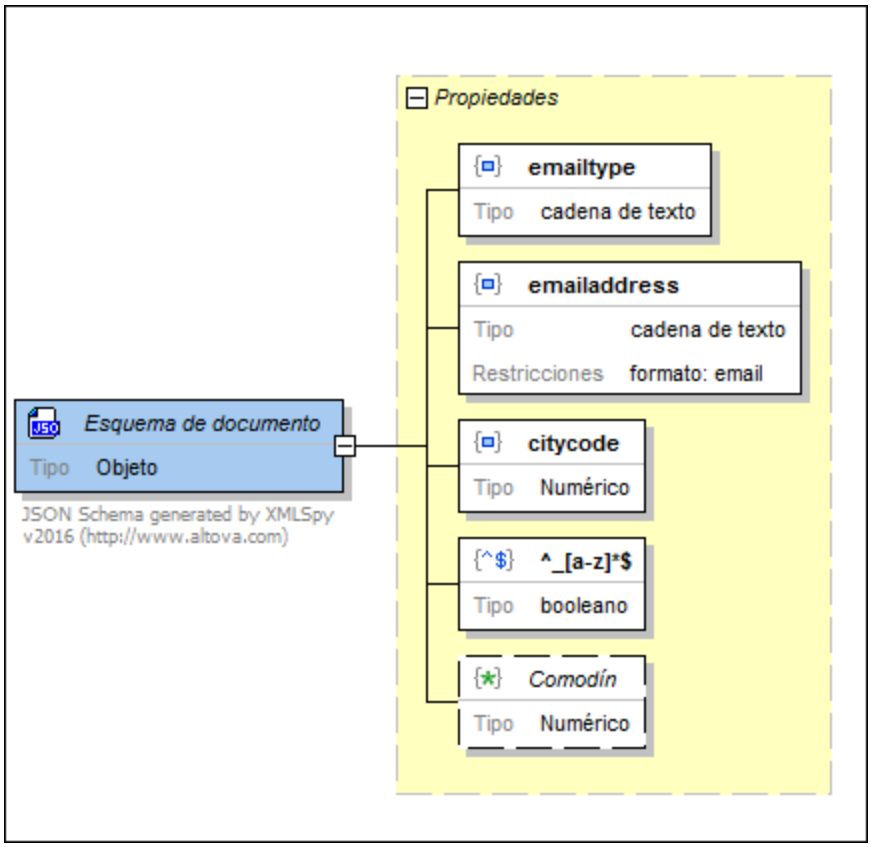
1. Se compara el nombre que tiene la propiedad en el esquema con todas las propiedades con nombre del objeto.
2. Si no hay coincidencias, el nombre se compara con todas las propiedades de patrón del conjunto de propiedades del objeto.
3. Si sigue sin haber coincidencias, entonces se invoca el comodín (si lo hay).
4. Si aún así no hay coincidencias, se emite un error de validación. Si el nombre coincide con una propiedad o una propiedad de patrón (o si existe un comodín), entonces se compara el valor con el valor de la definición de propiedad correspondiente.
5. Si el valor de instancia coincide con el tipo y las restricciones de la definición de propiedad correspondiente, entonces se considera que la propiedad es válida. De lo contrario, la propiedad no es válida.

Ejemplo

En el ejemplo de la imagen se define un objeto que:

- debe tener tres propiedades llamadas `emailtype`, `emailaddress` y `citycode`
- puede tener una propiedad o más cuyo nombre empiece con un guión bajo y cuyo valor sea un booleano (consulte la propiedad de patrón de la imagen)
- puede tener una propiedad adicional o más cuyo nombre y valor puede ser cualquiera

© 2016-2022 Altova GmbH



13.6.7 Propiedades sin especificar

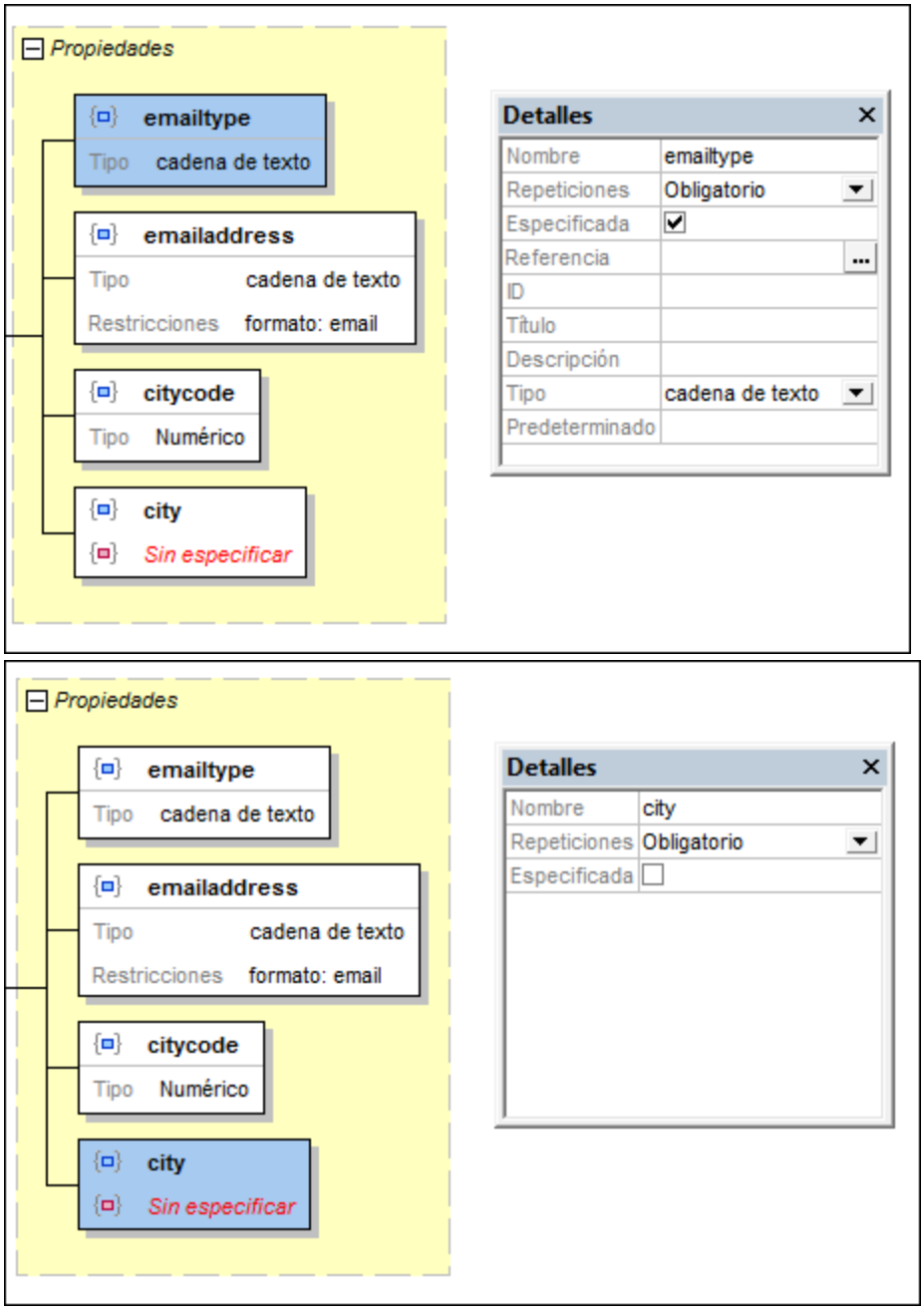
En el fragmento de código que aparece más abajo puede ver que la palabra clave **required** determina que es obligatorio que este objeto tenga cuatro propiedades. Sin embargo, solamente se han definido tres de las cuatro propiedades que son necesarias. La cuarta propiedad, **city**, está sin definir. El estado de las propiedades definidas es `Especificada`, mientras que la propiedad que está sin definir tiene el estado `Sin especificar`.

Fragmento de código: propiedades especificadas y sin especificar

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "JSON Schema generated by XMLSpy v2016 (http://www.altova.com)",
    "type": "object",
    "properties": {
        "emailtype": {
            "type": "string"
        },
        "emailaddress": {
            "type": "string",
            "format": "email"
        },
```

```
            "citycode": {
                "type": "number"
            }
        },
        "required": [
            "emailtype",
            "emailaddress",
            "citycode",
            "city"
        ],
        "additionalProperties": false
    }
```


© 2016-2022 Altova GmbH



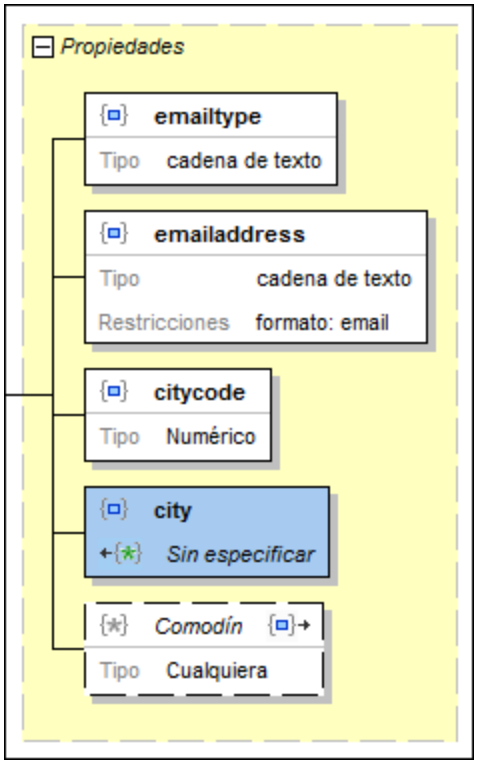
En la Vista de diseño la propiedad sin especificar se marca en rojo porque es obligatoria según el esquema pero no está definida. Aunque el esquema JSON es válido en sí mismo, los documentos de instancia que se validen con este esquema no lo serán. El motivo es que (i) si falta la propiedad `city`, el documento no será válido porque la propiedad `city` es obligatoria; (ii) si la propiedad `city` está presente, el documento no será válido porque la propiedad `city` está sin definir y no hay comodín de propiedad que permita su presencia (consulte el párrafo *Especificar una propiedad de forma implícita* 729) .

Para crear una definición para una propiedad que está sin especificar:

1. Seleccione la propiedad que está sin especificar en la Vista de diseño.
2. En el ayudante de entrada Detalles marque la casilla `Especificada` (*imagen anterior*). También puede activar/desactivar esta marca con el comando **Especificada** del menú contextual.
3. Modifique la definición de la propiedad según corresponda.

Especificar una propiedad de forma implícita

Puede especificar una propiedad de forma implícita añadiendo una propiedad de patrón o un comodín de propiedad adecuados. En la imagen siguiente puede ver que se añadió un comodín de propiedad. Con este comodín coincidirá una propiedad de la instancia cuyo nombre sea `city`. Por tanto, en el esquema, podemos decir que la propiedad `city` está especificada de forma implícita por medio del comodín. El documento de instancia será válido según el esquema si contiene la propiedad `city`.



Observe los símbolos que aparecen en los recuadros de la propiedad especificada de forma implícita y del comodín de propiedad. Cada símbolo es un vínculo a la otra propiedad. Haga doble clic en cualquiera de estos símbolos para seleccionar la otra propiedad.

13.6.8 Objetos y dependencias

Dentro de la definición de un objeto puede especificar que una propiedad determinada esté presente solamente si otra propiedad lo está. Según esto, la primera propiedad es dependiente de la segunda. Por ejemplo, imaginemos un objeto llamado `member` que tiene una propiedad llamada `credit_card`, definida como opcional. La propiedad `billing_address` del objeto puede definirse como dependiente de la propiedad `credit_card`. Es decir, la propiedad `billing_address` estará presente solamente si también lo está la propiedad `credit_card`.

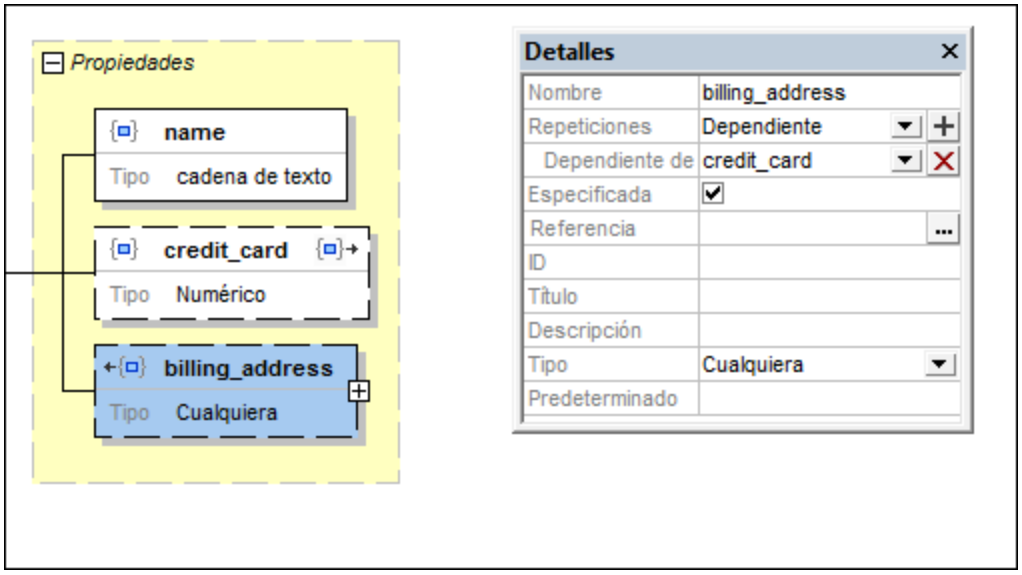
© 2016-2022 Altova GmbH

Este tipo de dependencia puede especificarse de dos maneras distintas:

- como dependencia de propiedad (la estructura dependiente es una propiedad) o
- como dependencia de esquema (la estructura dependiente es un esquema).

Dependencias de propiedad

En la siguiente imagen puede ver un objeto que tiene una propiedad llamada `name` que es obligatoria, una propiedad llamada `credit_card` que es opcional y una propiedad llamada `billing_address` que es dependiente. La propiedad `billing_address` depende de la propiedad `credit_card`. El código de esta definición de objeto JSON aparece justo debajo de la imagen. Más abajo explicamos cómo crear dependencias de propiedad.



Ejemplo de código de un objeto JSON con una dependencia de propiedad

```
{
   "type": "object",
   "properties": {
      "name": {
      "type": "string"
      },
      "credit_card": {
         "type": "number"
      },
      "billing_address": {
         "type": "string"
      }
   },
   "required": [ "name" ],
   "dependencies": {
      "credit_card": [ "billing_address" ]
   },
   "additionalProperties": false
}
```

Siga estos pasos para crear una dependencia de propiedad:

1. Haga clic con el botón derecho en la propiedad en la que se basará la dependencia (en este ejemplo, la propiedad `credit_card`).
2. En el menú contextual seleccione el comando **Agregar dependencia | Propiedad dependiente**. Esto añade una propiedad nueva que tiene el valor `Dependiente` en el campo *Repeticiones* (ayudante de entrada Detalles).
3. Defina el nombre y valor de esta propiedad. Después puede añadir detalles o restricciones a la propiedad.

Siga estos pasos para especificar que una propiedad dependa de otra:

1. Haga clic con el botón derecho en la propiedad que debe ser dependiente de otra (en este ejemplo, la propiedad `billing_address`).
2. En el menú contextual seleccione el comando **Dependiente**. Otra manera de hacerlo es seleccionar el valor `Dependiente` en el campo *Repeticiones* del ayudante de entrada Detalles.
3. En el ayudante de entrada Detalles abra la lista desplegable del campo *Dependiente de*. Esta lista desplegable incluye las demás propiedades del objeto. Seleccione la propiedad de la que debe depender la propiedad actual.

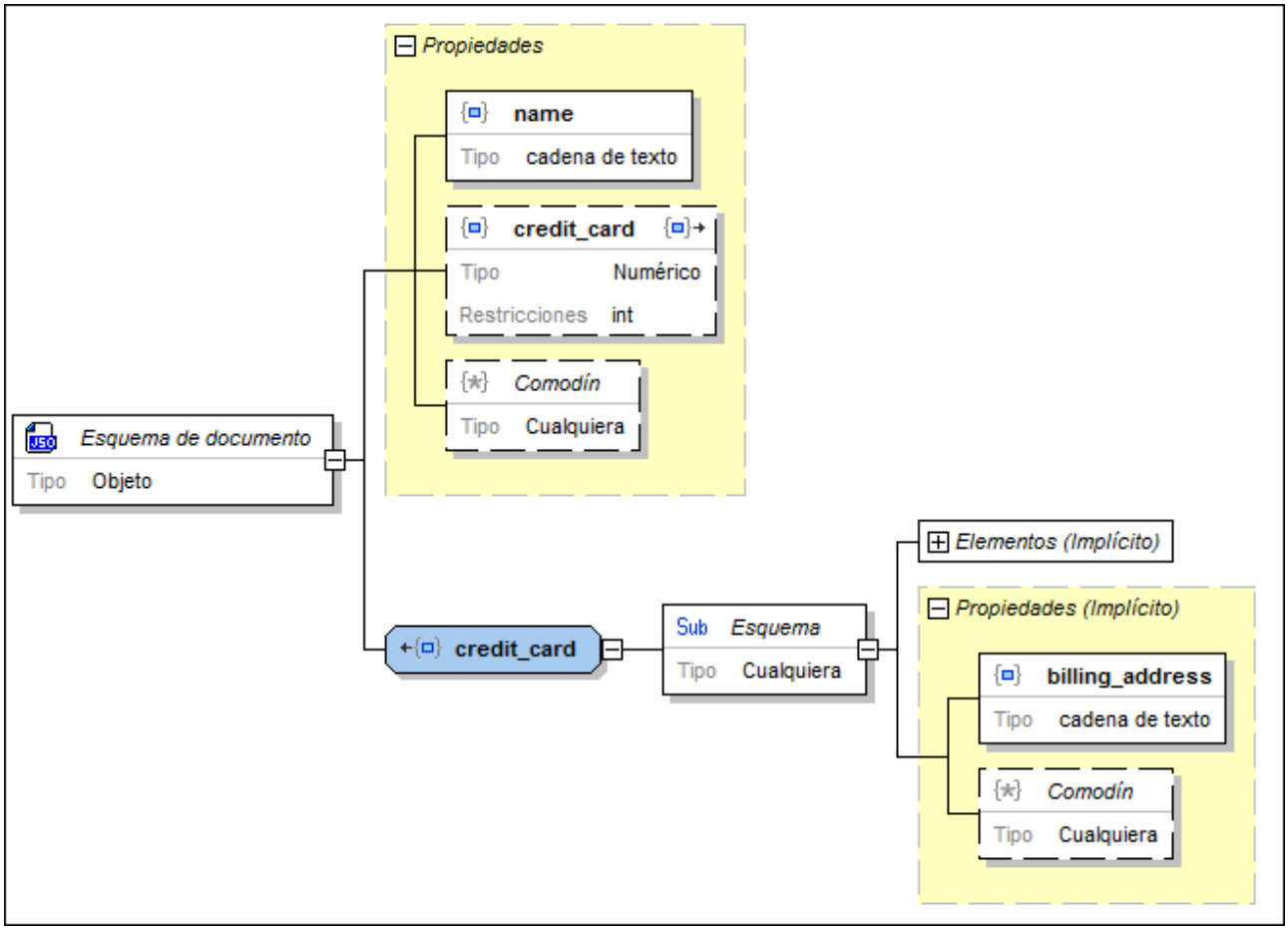
Nota: en los recuadros de las dos propiedades que tienen una relación de dependencia aparece un icono. Si hace doble clic en el icono de una propiedad, se selecciona la otra propiedad.

Nota: una propiedad puede tener más de una propiedad dependiente.

Dependencias de esquema

En la imagen siguiente aparece un objeto que describe la misma estructura de datos que el objeto descrito en el apartado anterior. Sin embargo, las definiciones de los dos objetos son distintas. Mientras que en la definición anterior se usaba una *dependencia de propiedad* para definir la propiedad `billing_address` como dependiente de `credit_card`, en esta definición se usa una dependencia de esquema para definir la dependencia. El código de esta definición de objeto JSON aparece justo debajo de la imagen. Más abajo explicamos cómo crear dependencias de esquema.

© 2016-2022 Altova GmbH



Ejemplo de código de un objeto JSON con una dependencia de esquema

```
{
   "type": "object",
   "properties": {
      "name": {
         "type": "string"
      },
      "credit_card": {
         "type": "integer"
      }
   },
   "required": [ "name" ],
   "dependencies": {
      "credit_card": {
         "properties": {
            "billing_address": {
               "type": "string"
            }
         },
         "required": [ "billing_address" ]
      }
   }
```

```
}
```

Siga estos pasos para crear una dependencia de esquema:

1. Haga clic con el botón derecho en la propiedad en la que se basará la dependencia (en este ejemplo, la propiedad `credit_card`).
2. En el menú contextual seleccione el comando **Agregar dependencia | Dependencia de esquema**. Esto crea una definición de objeto nueva, que tendrá el mismo nombre que la propiedad de la que depende (en este ejemplo, la propiedad `credit_card`) y tendrá un subesquema secundario.
3. Defina el subesquema como quiera, añadiendo detalles y restricciones.

Nota: en los recuadros de la propiedad y del objeto que tienen una relación de dependencia aparece un icono. Si hace doble clic en el icono de un recuadro, se selecciona el otro recuadro.

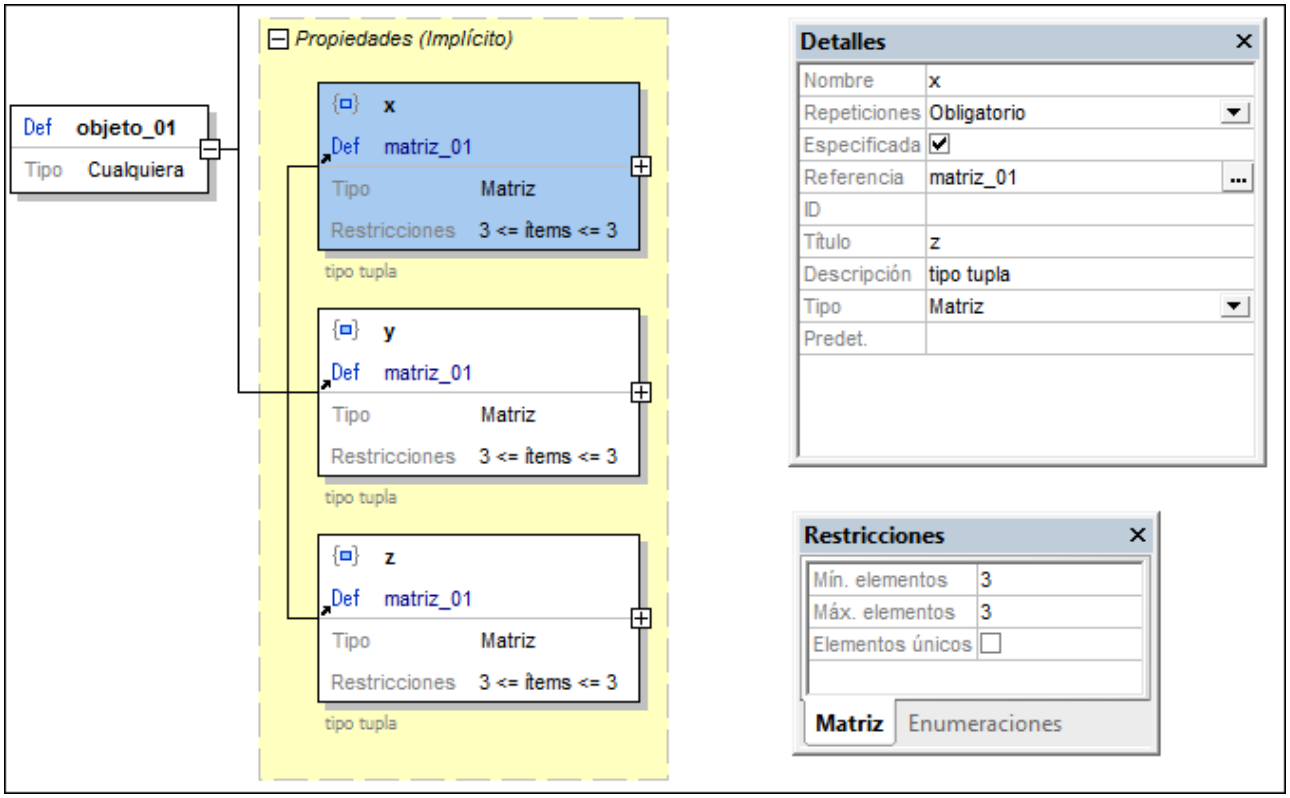
Nota: para establecer varias dependencias, créelas dentro del subesquema dependiente.

13.6.9 Matrices

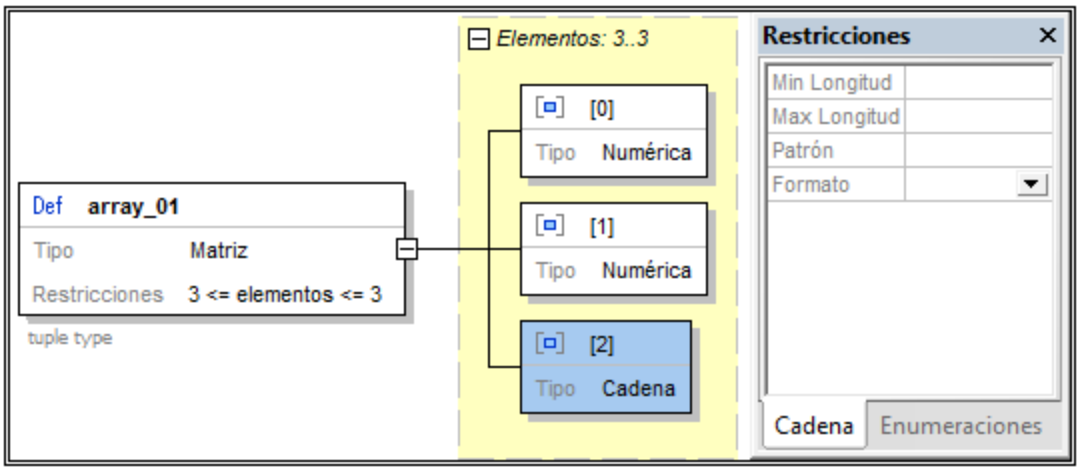
Una matriz es una lista de cero o más elementos ordenados delimitada por corchetes. Cada elemento de la lista tiene asignado un tipo. El código de instancia que aparece a continuación es de un objeto con tres propiedades. El valor de cada propiedad es una matriz (*sus delimitadores aparecen resaltados en amarillo*).

```
{
    "x": [ 1, 2, "abc" ],
    "y": [ 3, 4, "def" ],
    "z": [ 5, 6, "ghi" ]
}
```

Las tres matrices de este ejemplo tienen la misma definición. Cada una de ellas contiene tres elementos ordenados de esta forma: (i) primero un número, (ii) después otro número y (iii) por último una cadena de texto. Más abajo puede ver este objeto en el esquema. Como las tres matrices tienen la misma definición, se creó una definición en una matriz global llamada `matriz_01`. Cada una de las tres matrices (`x`, `y` y `z`) remite a la matriz global 721 `matriz_01`.



En la imagen anterior está seleccionada la matriz **x** (su recuadro está resaltado en azul) y sus detalles y restricciones aparecen en los correspondientes ayudantes de entrada. Observe que tiene una restricción en cuanto al número de elementos permitidos. El número se puede editar en el ayudante de entrada Restricciones y aparece en el diagrama del esquema. Los elementos de la matriz se pueden definir en la definición de la matriz, que es en este caso la definición global `matriz_01` (*imagen siguiente*).



Debe tener en cuenta estos aspectos:

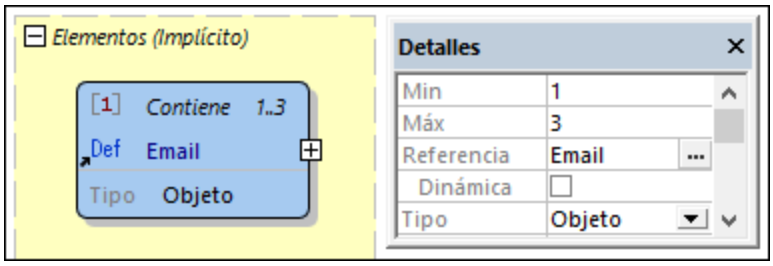
- La restricción `unique` indica que todos los elementos de la matriz deben ser únicos.
- La numeración de los elementos empieza por `0`.

- Las frases *3 <= elementos <= 3* y *Elementos: 3..3* que pueden verse en la imagen indican el máximo y mínimo de elementos permitidos. En este caso debe haber exactamente tres elementos.

Agregar elementos y comodines de elemento a una matriz y la palabra clave `contains`

Los elementos de matriz, los comodines de elemento de matriz y la palabra clave **`contains`** se agregan desde el menú contextual de una definición o desde un elemento de matriz.

- Un comodín de matriz permite incluir una mayor variedad de objetos en la matriz.
- La palabra clave **`contains`**, a su vez, indica que el valor de la palabra clave **`contains`** debe ser un esquema JSON válido y que al menos uno de los elementos de la matriz debe ser válido conforme al esquema dado. A partir del borrador 2019-09 aparecen las palabras clave **`minContains`** y **`maxContains`**. En el ayudante de entrada Detalles estas palabras clave se corresponden con las propiedades *Min* y *Max* de la caja Contains (*imagen siguiente*) y definen cuántos elementos pueden coincidir con el objeto de esquema referenciado.



13.6.10 Tipos atómicos

Hay cinco tipos JSON atómicos (es decir, tipos simples o primitivos): (i) `string`, (ii) `number`, (iii) `integer`, (iv) `boolean` y (v) `null`. Si desea que una definición tenga uno de estos tipos atómicos:

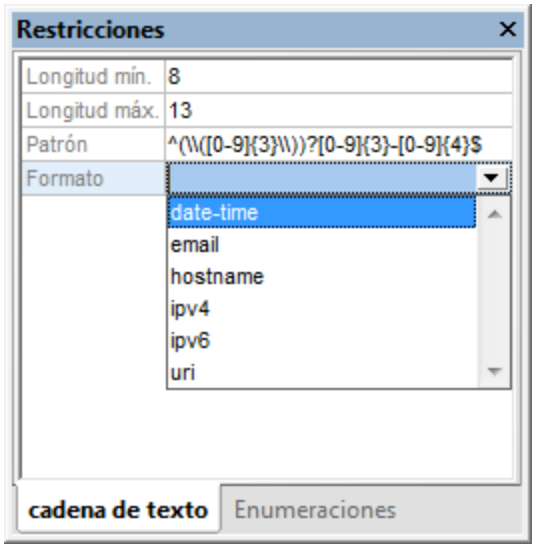
- Haga doble clic en el campo `Tipo` del recuadro de la definición y seleccione el tipo correspondiente.
- En el ayudante de entrada Detalles seleccione el tipo en la lista desplegable del campo `Tipo`.

A continuación describimos las restricciones de cada tipo atómico.

Cadena de texto

El tipo `cadena de texto` puede tener estas restricciones: (i) longitud de la cadena, (ii) un patrón definido por una expresión regular y (iii) un formato predefinido por la especificación JSON.

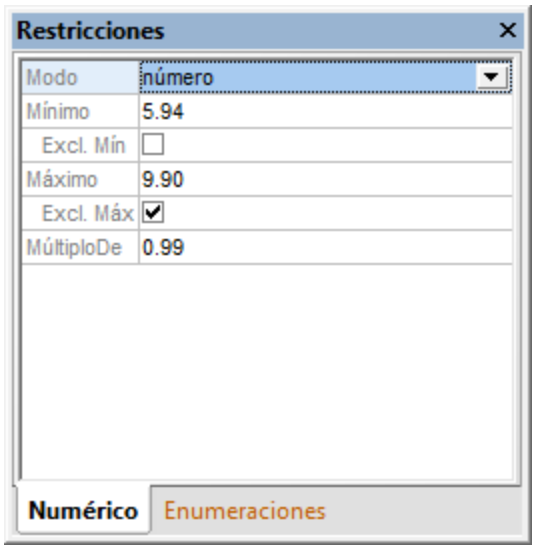
© 2016-2022 Altova GmbH



Nota: en las opciones de validación JSON[1595] de XMLSpy puede indicar si el formato de las cadenas de los documentos de instancia JSON se debe validar o no.

Numérico

El tipo numérico agrupa estos dos tipos: `número` y `entero`). El tipo propiamente dicho se establece en el campo `Modo` (cuyo valor predeterminado es `número`). La diferencia entre estos tipos es que el tipo `número` permite el uso de decimales, mientras que `entero` no los permite. Si ya existe un valor en el campo `Múltiplo de`, entonces el valor de instancia debe ser un entero que sea múltiplo del valor de `Múltiplo de`.



Por ejemplo, los valores del tipo numérico definido en la imagen anterior son: `5.94`, `6.93`, `7.92` y `8.91`.

Booleano y null

El tipo `booleano` toma los valores `true` o `false`. El tipo `null` toma el valor `null`. Ninguno de estos tipos puede tener restricciones.

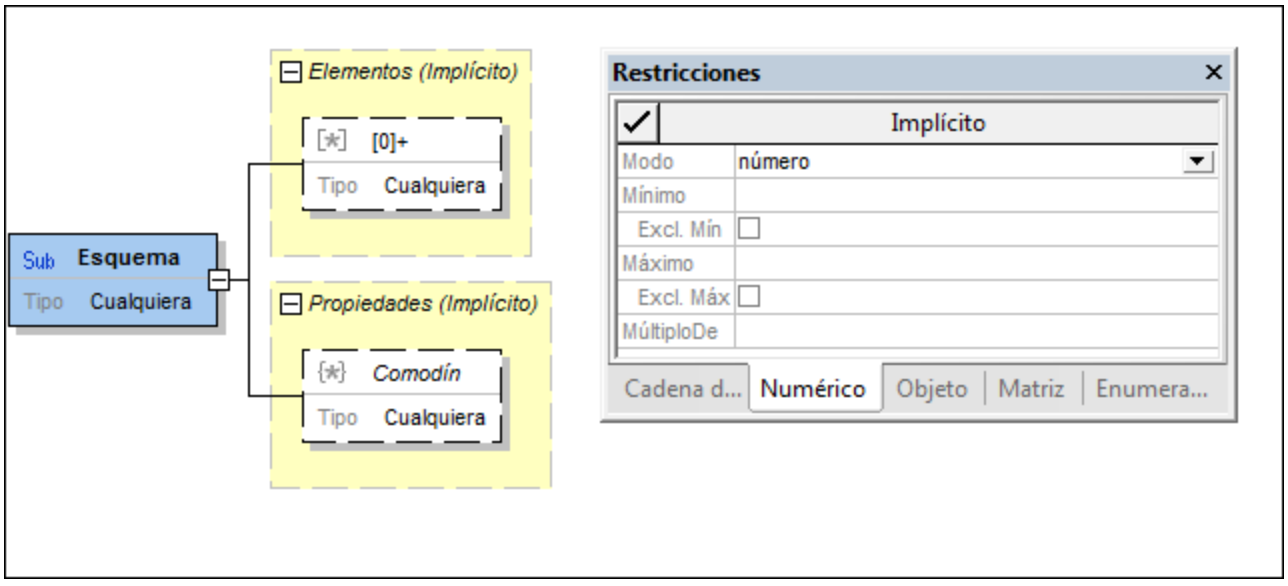
13.6.11 Selectores de tipo (cualquiera, múltiple, etc.)

En la lista desplegable de los cuadros combinados Tipo de la vista Esquema JSON aparecen cuatro tipos que no son tipos JSON: **`cualquiera`**, **`múltiple`**, **`libre`** y **`prohibido`**. Estos tipos son en realidad selectores de tipo.

- El selector de tipo **`cualquiera`** selecciona cualquier tipo JSON. Esto significa que en esa instancia cualquier tipo JSON es válido para una definición con ese nombre.
- El selector de tipo **`múltiple`** selecciona uno o varios tipos JSON. Esto significa que si el tipo de instancia es uno de los tipos JSON seleccionados en el esquema, entonces el tipo de la instancia será válido para la definición.
- El selector de tipo **`libre`** (nuevo en draft-06[715]) no aplica ninguna restricción al tipo JSON. Esto significa que en esa instancia cualquier tipo JSON es válido para una definición con ese nombre.
- El selector de tipo **`prohibido`** (nuevo en draft-06[715]) prohíbe cualquier tipo JSON, de manera que no puede existir ninguna definición con ese nombre.

El selector de tipo cualquiera

El selector de tipo `cualquiera` se puede seleccionar siempre que se pueda seleccionar un tipo. Cuando se añade una definición al esquema, el tipo predeterminado es `cualquiera`. Esto especifica que cualquier tipo JSON es válido y que el tipo de la instancia puede ser un objeto, una matriz o cualquier tipo atómico (`string`, `number`, `integer`, `boolean` y `null`).



Por ejemplo, el subesquema de la imagen anterior tiene el tipo `cualquiera`. Es decir, todos los tipos JSON son válidos para esta definición. Además observe que:

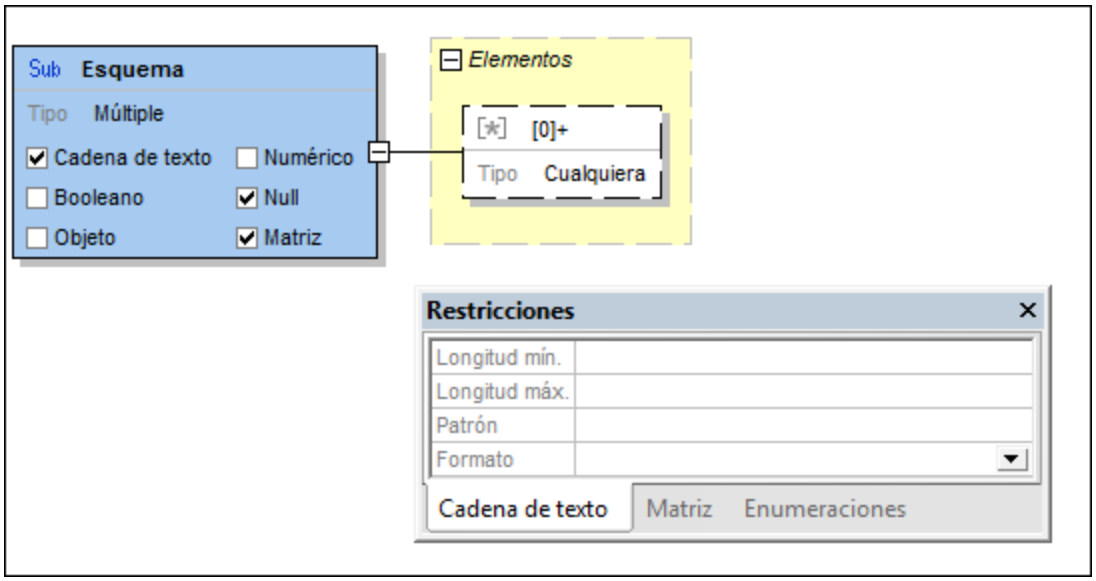
© 2016-2022 Altova GmbH

- Como se permiten objetos, en la vista se crea automáticamente un recuadro de propiedades. El recuadro de propiedades se define por defecto para permitir un número cualquiera de propiedades de cualquier tipo (por medio de un comodín de propiedad cuyo tipo es `cualquiera`). Las definiciones de propiedades se pueden definir según corresponda.
- Como se permiten matrices, en la vista se crea automáticamente un recuadro de elementos. El recuadro de elementos de matriz se define por defecto para permitir un número cualquiera de elementos de matriz de cualquier tipo (por medio de un comodín de elemento de matriz cuyo tipo es `cualquiera`). Las definiciones de los elementos se pueden modificar según corresponda.
- Como se permiten tipos de cadena de texto y numéricos (números y enteros), puede definir restricciones para estos tipos atómicos en el ayudante de entrada Restricciones.

Por tanto, todos estos tipos se definen de forma implícita con el selector de tipos `cualquiera`. Seleccione otro tipo si desea cambiar de tipo. También puede especificar como tipo un objeto o una matriz concretos: haga clic con el botón derecho en el objeto o en la matriz y seleccione el comando **Convertir en explícito**. Esto convierte el tipo en el tipo seleccionado y quita los demás tipos o los hace inactivos.

El selector de tipo múltiple

El selector de tipo `múltiple` se puede seleccionar siempre que se pueda seleccionar un tipo. Permite seleccionar uno o varios tipos JSON con solo marcar una casilla (*imagen siguiente*). Después se pueden especificar sus restricciones en el ayudante de entrada Restricciones.



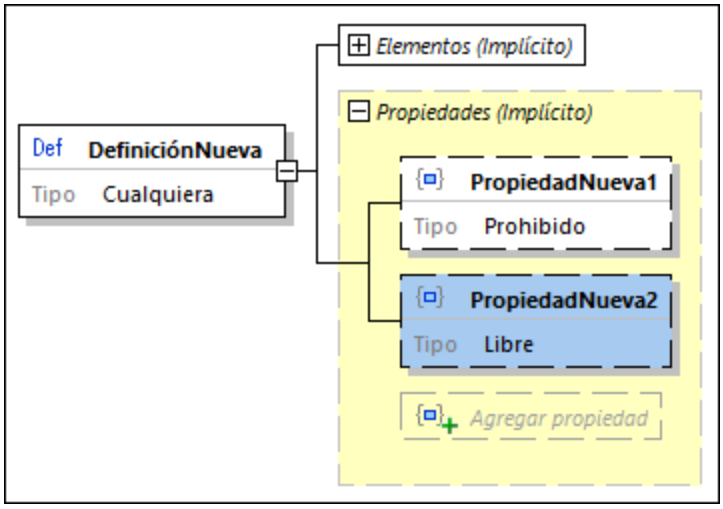
Por ejemplo, el subesquema de la imagen anterior permite tipos `string`, `null` y `array`. Puede definir restricciones para estos tipos en el ayudante de entrada Restricciones.

- Las restricciones de cadenas de texto se definen en el ayudante de entrada Restricciones.
- El tipo null no toma restricciones.
- Se crea automáticamente un recuadro de elementos de matriz. Aquí puede definir el número y los tipos de elementos de matriz.

En un documento de instancia los tipos seleccionados se permitirán en la posición correspondiente a la del subesquema.

Los selectores de tipo libre y prohibido

Los selectores de tipo `libre` y `prohibido` se pueden seleccionar en cualquier sitio en el que se pueda seleccionar un tipo y permiten indicar, respectivamente, que los objetos de un tipo cualquiera están permitidos o prohibidos.



En la imagen anterior vemos que la definición tiene dos propiedades. **`PropertyOne`** puede tener un valor de cualquier tipo, mientras que una propiedad con el nombre **`PropertyTwo`** no está permitida. En forma de código la imagen anterior tendría este aspecto:

```
"Definition": {
	"properties": {
		"PropertyOne": true,
		"PropertyTwo": false
	}
}
```

13.6.12 BSON (JSON binario) para MongoDB

La plataforma de datos MongoDB almacena los datos como representación binaria de los mismos pero mantiene la estructura JSON. Esa representación binaria se conoce como JSON binario o BSON. Las principales ventajas de usar BSON para MongoDB son:

- El formato binario BSON se analiza más rápidamente que el texto de los documentos JSON.
- JSON limita los tipos de datos, por lo que BSON cuenta con más tipo de datos (más concretamente, más tipos de datos numéricos).

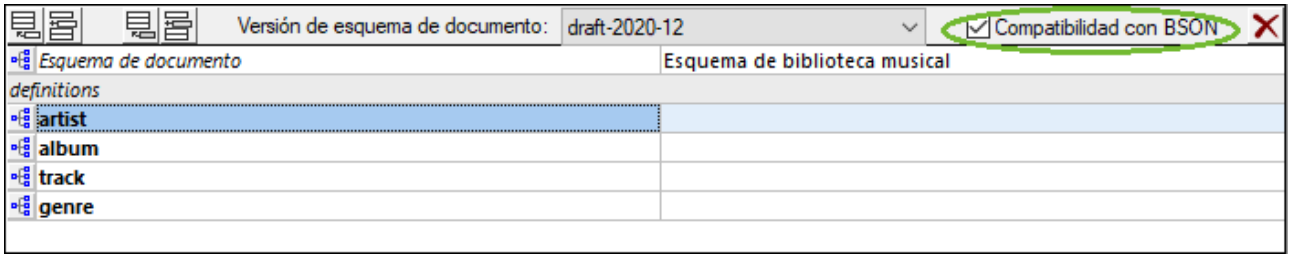
Para más información sobre MongoDB y BSON consulte esta página del sitio web de MongoDB.

© 2016-2022 Altova GmbH

Editar esquemas JSON para datos BSON

En MongoDB puede validar los datos de BD con un documento de esquema JSON mientras se agregan nuevos o se modifican los que ya existen. Sin embargo, como los tipos de datos nuevos de BSON no forman parte de las especificaciones oficiales de JSON Schema, sino que son un añadido, los esquemas JSON para BSON se editan en XMLSpy, en cuyo editor JSON se aplica una capa para BSON que se solapa con el esquema. Esta capa de edición permite añadir y modificar más opciones JSON.

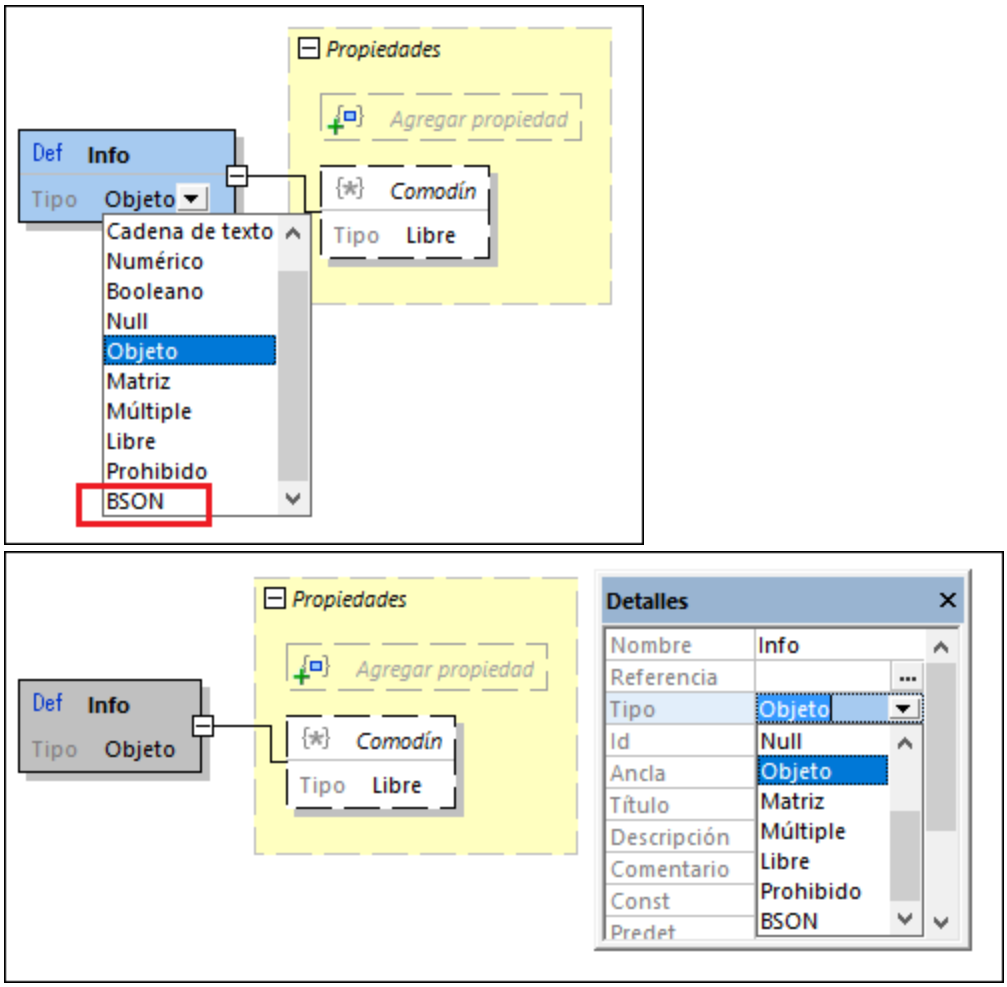
Esto quiere decir que puede editar documentos de esquema JSON de cualquier versión con o sin la capa de edición para BSON. Si no se usa la capa de edición para BSON al editar un documento, este se edita como documento JSON normal. Con BSON, el documento de esquema JSON puede definir también características específicas de BSON. Para cambiar las opciones de edición de BSON en la vista Esquema JSON *para el documento activo*, marque la casilla *Compatibilidad con BSON* de la parte derecha superior de la ventana principal (*dentro de un círculo verde en la imagen siguiente*). Recuerde que la compatibilidad BSON (i) se puede activar o desactivar independientemente de la versión del esquema que se haya elegido y (ii) afecta solamente al documento que esté activo en ese momento; es decir, se puede activar o desactivar para cada documento por separado.



Tipos BSON

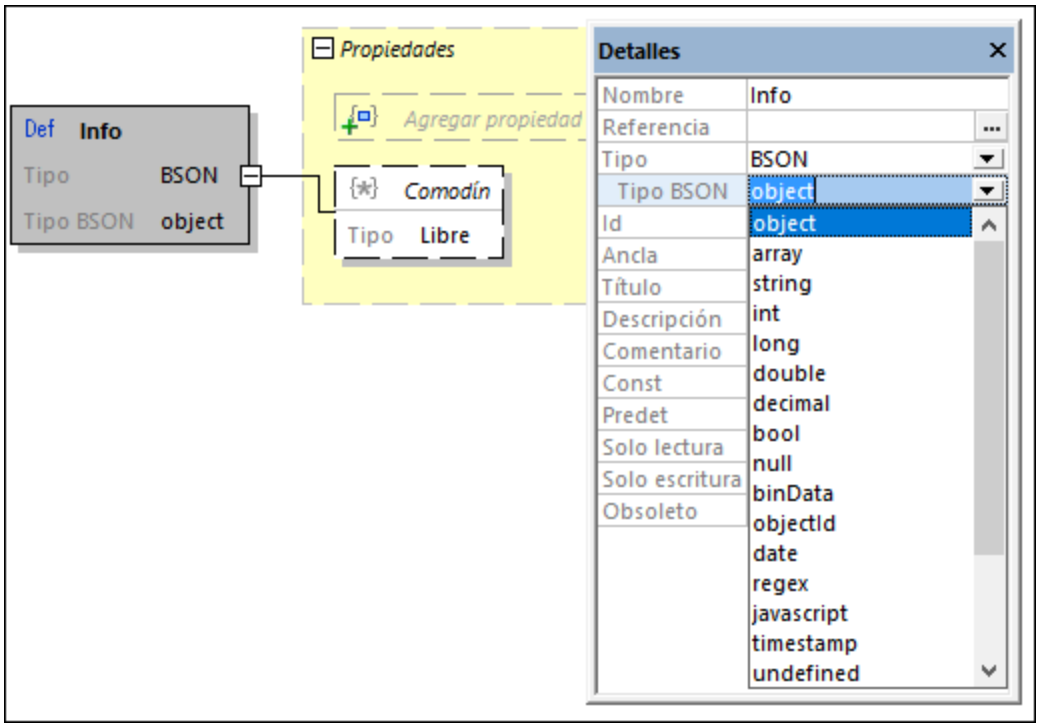
Una vez haya activado la capa de edición para BSON se habilitan también los tipos de datos BSON para los objetos, las propiedades y los elementos de matrices JSON. Para indicar que un componente es de tipo BSON debe seguir dos pasos:

1. Para indicar que el tipo de datos del componente es BSON (y no un tipo de datos JSON) seleccione BSON como el tipo JSON de base. Puede llevar a cabo este paso en el cuadro combinado de selección de tipos de datos del componente (haciendo doble clic en el valor del tipo, *imagen siguiente izquierda*) o en el ayudantes de entrada Detalles del componente (*imagen siguiente derecha*).



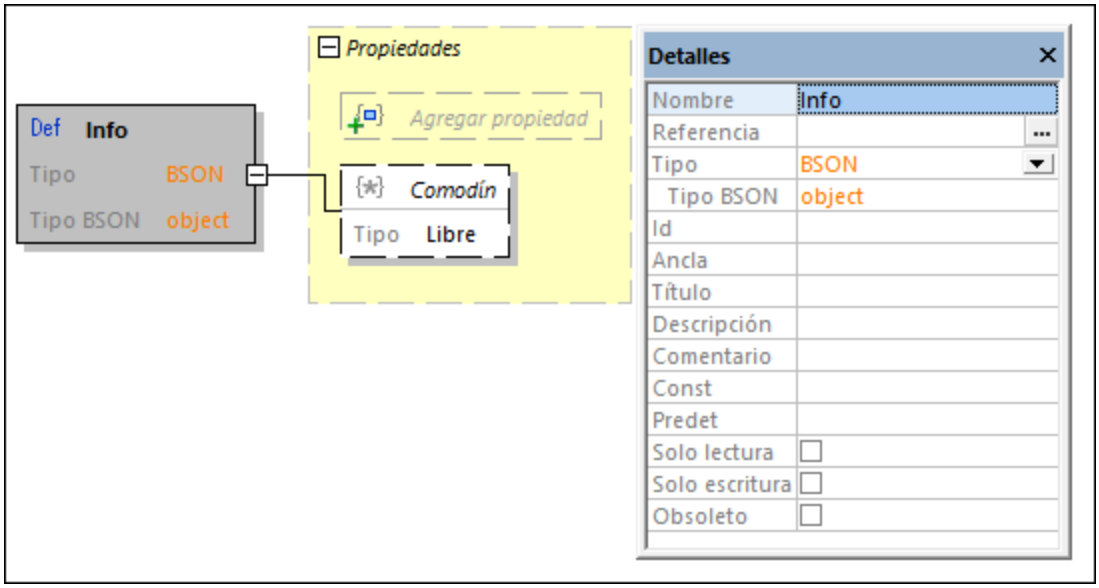
2. Una vez haya indicado que el tipo base del componente es *BSON* se habilitan los tipos BSON. Puede seleccionar el tipo BSON en el selector de tipo de datos del componente (*flecha izquierda en la imagen siguiente*) o en el ayudantes de entrada Detalles (*flecha derecha en la imagen siguiente*).

© 2016-2022 Altova GmbH



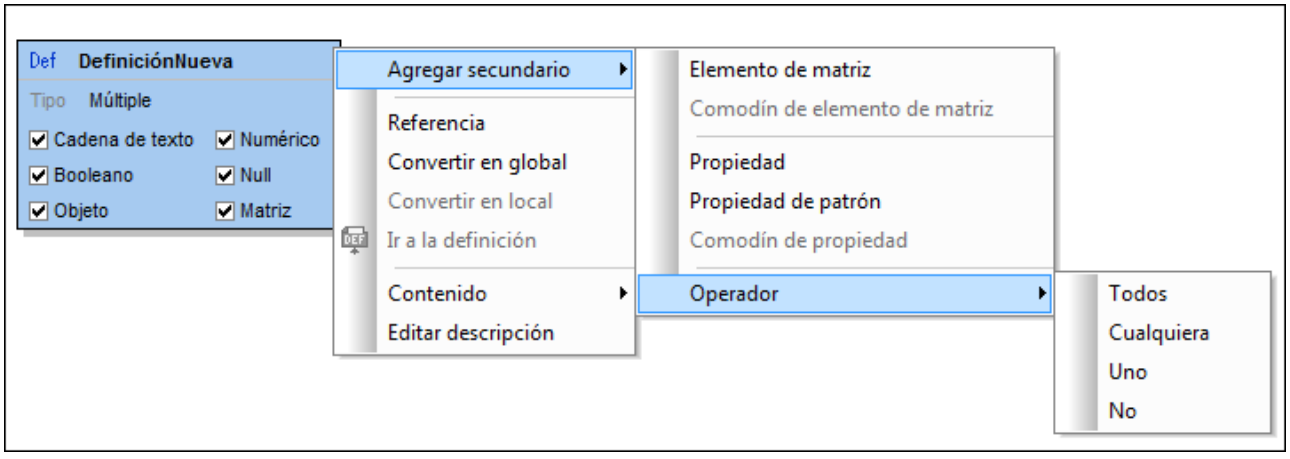
Deshabilitar la compatibilidad con BSON

Si asigna un tipo BSON a un componente de esquema JSON y desmarca la opción *Compatibilidad con BSON* para el documento en trabajando, aparece una caja de mensaje que le informa de que el documento contiene tipos BSON y le pregunta si los quiere eliminar, convertir o conservar. Si elige eliminar o convertirlos, se convertirán los tipos BSON que permitan esta acción y el resto se eliminará. Si elige conservar los tipos BSON, estos aparecerán en naranja en la vista Esquema porque se ha deshabilitado la compatibilidad con los tipos BSON (*imagen siguiente*).



13.6.13 Operadores

Existen cuadro operadores: (i) `todos`, (ii) `cualquiera`, (iii) `uno` y (iv) `no`. Los operadores sirven para especificar condiciones de validez y puede añadir un operador a cualquier definición. Para abrir el submenú de operadores, haga clic con el botón derecho en la definición donde desea agregar el operador y después seleccione **Agregar secundario | Operador** (*imagen siguiente*).



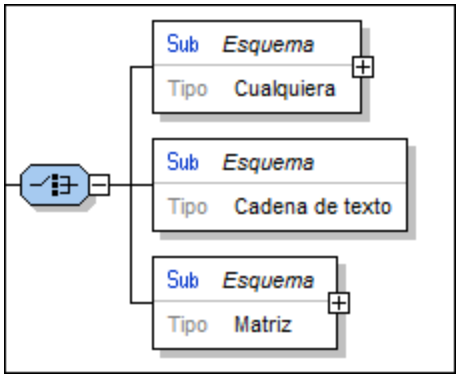
Estos operadores definen las condiciones necesarias para que la instancia sea válida:

Operador	Icono	Descripción
	Todos	Contiene uno o varios subesquemas (definiciones), añadidos como secundarios del operador. Una instancia es válida si es válida comparada con todos estos

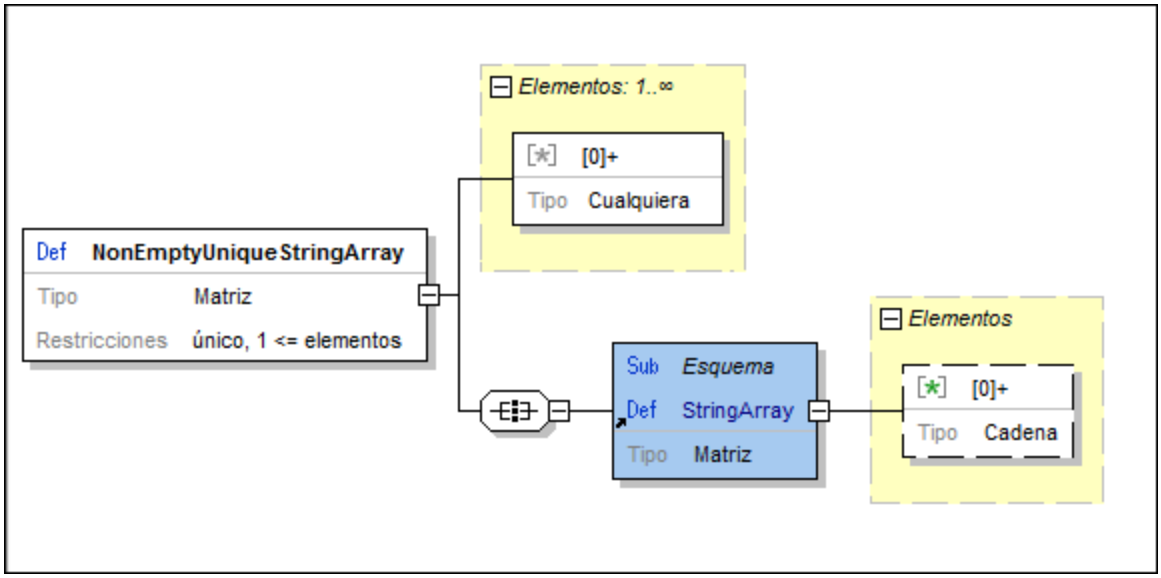
© 2016-2022 Altova GmbH

		subesquemas.
	Cualquiera	Contiene uno o varios subesquemas (definiciones), añadidos como secundarios del operador. Una instancia es válida si es válida comparada con uno de estos subesquemas como mínimo.
	Uno	Contiene uno o varios subesquemas (definiciones), añadidos como secundarios del operador. Una instancia es válida si es válida comparada con exactamente uno de estos subesquemas.
	No	Contiene exactamente un subesquema (definición), añadida como secundario del operador. Una instancia es válida si no es válida según la definición dada.

En el ejemplo de la imagen siguiente el operador *Uno* contiene tres subesquemas secundarios (definiciones). Para que la instancia sea válida debe tener una estructura de datos JSON (en este punto de la estructura del documento) que coincida con uno de las tres definiciones del subesquema.



Los operadores son muy prácticos a la hora de especificar reglas de herencia y restricciones. Por ejemplo, en la imagen siguiente se usó el operador *Todos* para definir una matriz que contiene cadenas de texto únicas no vacías.

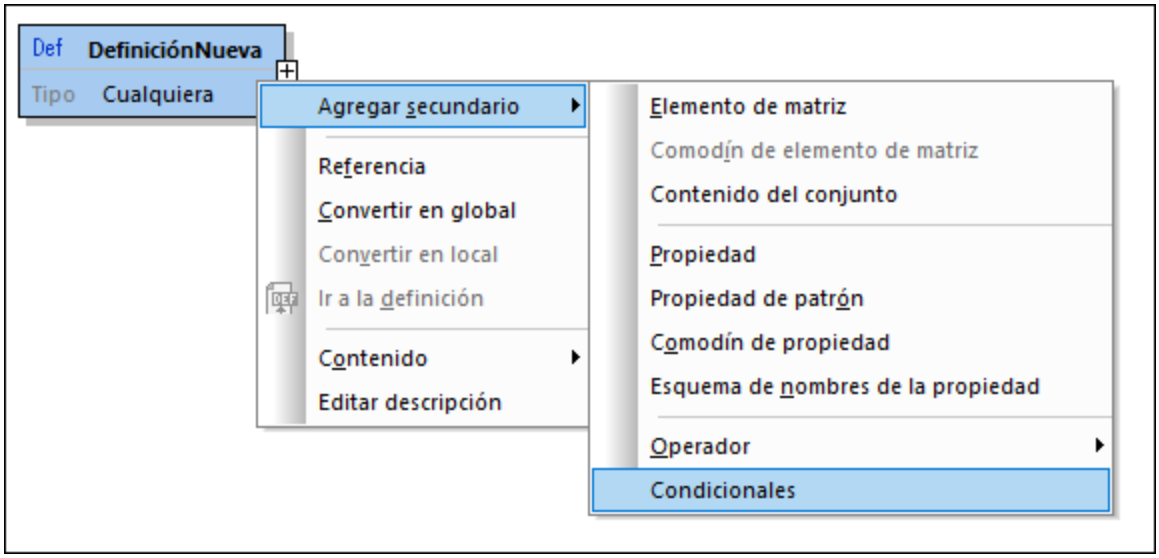


13.6.14 Condicionales

Los condicionales son una opción nueva de draft-07 715 y permiten indicar que las restricciones de validación deben ser distintas en función de ciertos aspectos del objeto en cuestión, como su tipo u otras restricciones propias de ese tipo.

Agregar un condicional

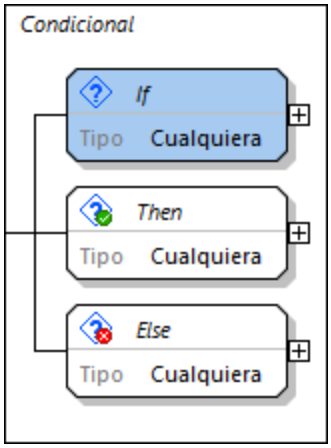
Puede agregar condicionales a cualquier definición desde el menú contextual de esa definición (*imagen siguiente*). Para acceder al submenú del condicional, haga clic con el botón derecho en la definición a la que quiere agregar el condicional y después seleccione **Agregar secundario | Condicionales**.



Configurar la validación de condicionales

El condicional se agrega como una caja con tres elementos: **`If-Then-Else`** (*imagen siguiente*).

© 2016-2022 Altova GmbH

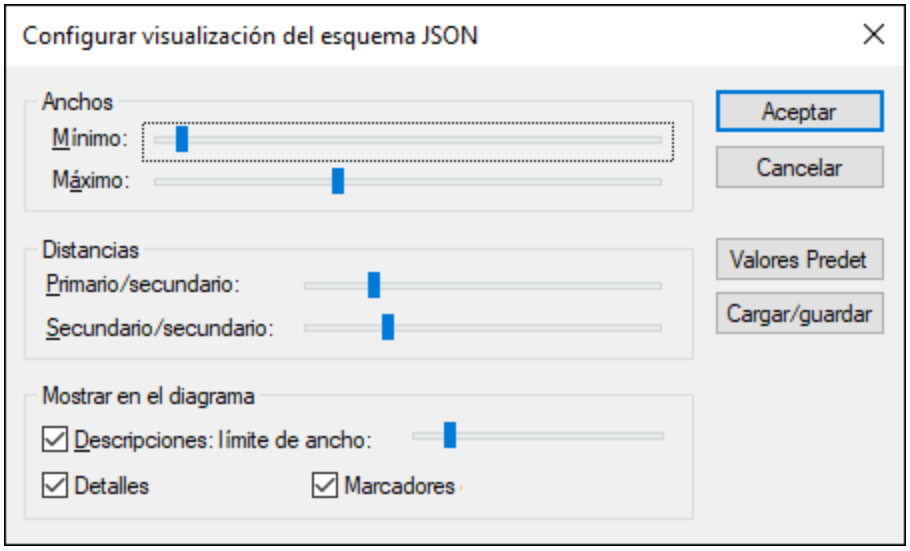


Para configurar la validación de condicionales:

1. Configure la condición de la caja *If*. Para ello seleccione un tipo en el ayudante de entrada Detalles y después una restricción basada en tipo en el ayudante de entrada Restricciones.
2. En la caja *Then* configure los requisitos de validación en caso de que se cumpla la condición indicada en la caja *If*.
3. En la caja *Else* configure los requisitos de validación en caso de que se cumpla la condición indicada en la caja *If*.

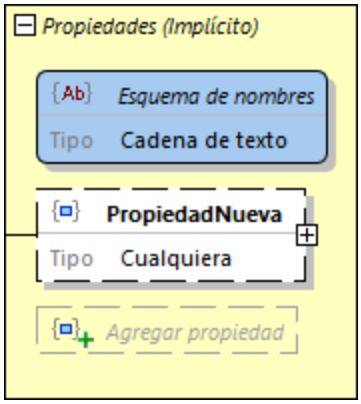
13.6.15 Configurar la vista Diseño

El cuadro de diálogo "Configurar visualización del esquema JSON" (*imagen siguiente*) sirve para configurar el aspecto del modo Vista de diseño y se abre con el comando de menú **Diseño de esquemas | Configurar opciones de vista**. Para abrir este cuadro de diálogo la ventana principal debe estar en el modo Vista de diseño.



Estos son los aspectos que se pueden configurar de la Vista de diseño:

- *Anchos:* dos controles deslizantes que determinan respectivamente el ancho mínimo y máximo de los recuadros de la vista Diseño. Es decir, determinan el ancho permitido de los recuadros.
- *Distancias entre primarios y secundarios:* control deslizante que determina la distancia entre cada nivel de la jerarquía.
- *Distancias entre secundarios:* control deslizante que determina la distancia vertical entre los recuadros.
- *Ancho de las descripciones:* control deslizante que determina el ancho de las líneas de las descripciones. Si el texto supera este límite, se inserta un ajuste automático de línea.
- *Visualización de detalles:* los detalles de las definiciones se pueden mostrar u ocultar en los recuadros de las definiciones. Para activar o desactivar esta opción puede usar el icono correspondiente de la barra de herramientas.
- *Visualización de marcadores de posición:* los marcadores de posición son elementos que todavía no se han definido; representan elementos potenciales. Esta opción indica si esos marcadores se deben mostrar u ocultar. Para activar o desactivar esta opción puede usar el icono correspondiente de la barra de herramientas. Por ejemplo, el elemento *Agregar propiedad* de la imagen siguiente es un marcador de posición.



 © 2016-2022 Altova GmbH

Nota: el comando de menú **Configurar opciones de vista** solo se activa en la Vista de diseño[714], que muestra la definición detallada de un objeto. No está disponible en la Cuadrícula de definiciones[714].

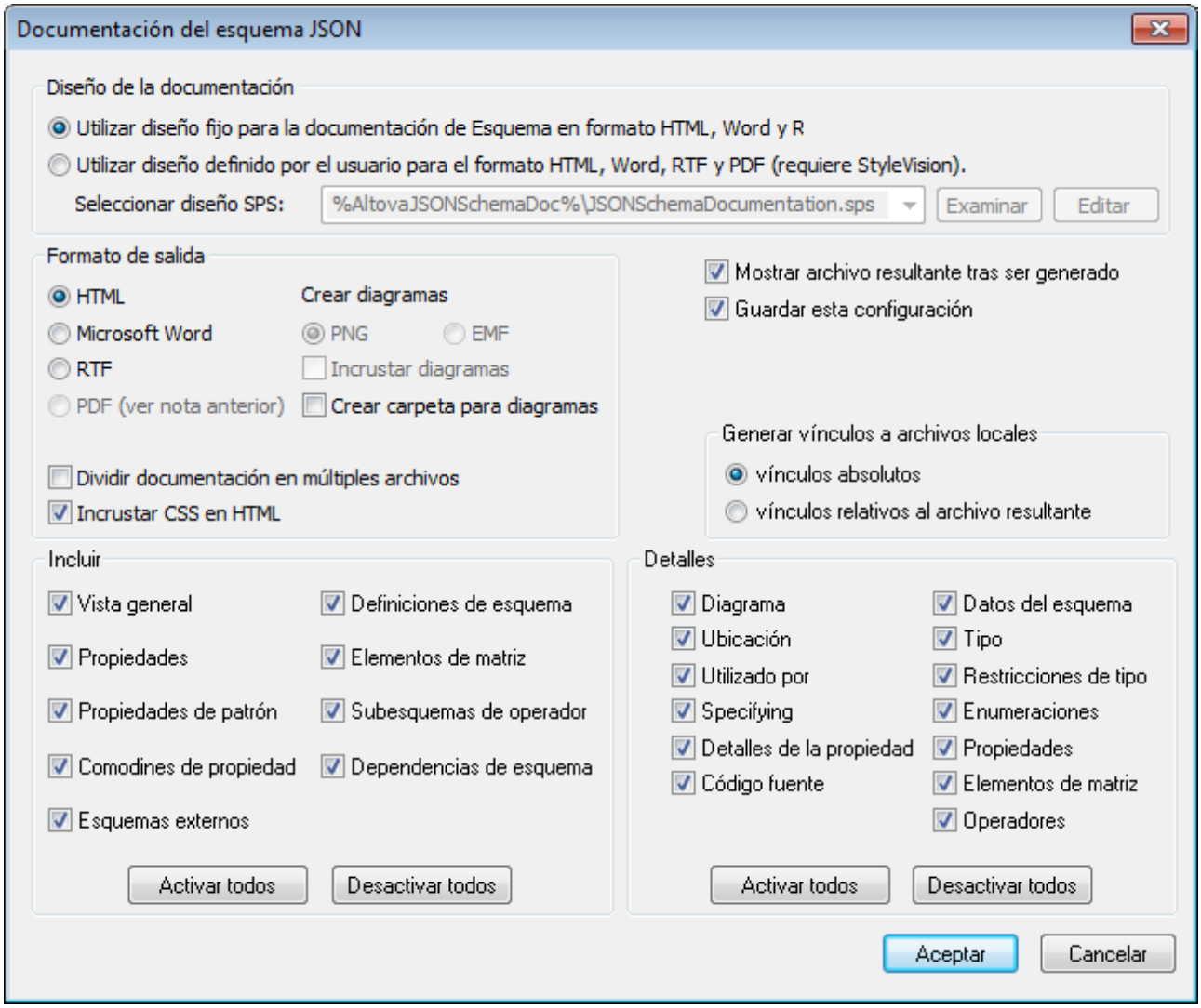
13.6.16 Generar documentación de esquemas JSON

Si el documento activo es un esquema JSON, puede generar documentación para este esquema con solo hacer clic en el comando **Diseño de esquemas | Generar documentación**. La documentación se puede generar en formato HTML, MS Word o RTF y además puede especificar qué componentes incluye. Los componentes JSON que estén relacionados entre sí se conectan mediante hipervínculos en la documentación, para que pueda navegar por el esquema con facilidad.

Nota: para poder generar documentación en formato MS Word es imprescindible tener instalado MS Word (versión 2000 o superior).

Pasos para generar la documentación de un esquema JSON

1. El documento activo debe ser el esquema JSON.
2. Cambie a la vista Esquema.
3. Seleccione el comando de menú **Diseño de esquemas | Generar documentación**. Esto abre el cuadro de diálogo "Documentación del esquema JSON" (*imagen siguiente*).
4. Seleccione el formato de salida: HTML, MS Word o RTF.
5. Seleccione qué componentes y detalles se incluyen en la documentación y configure el resto de opciones (*ver más abajo*).



6. Haga clic en **Aceptar** e introduzca el nombre con el que se debe guardar el archivo de la documentación.

Opciones de la documentación de esquemas JSON

- La plantilla de diseño puede ser el diseño integrado y fijo de XMLSpy o un diseño definido por el usuario y guardado en un archivo SPS. Para aprender a usar diseños definidos por el usuario consulte la sección Diseño definido por el usuario[1377].
- El formato de salida se especifica en el panel *Formato de salida*. Puede elegir entre generar la documentación en un solo archivo o dividirla en varios archivos. Si se decanta por generar varios archivos, cada archivo corresponderá a un componente del esquema. Puede elegir qué componentes se documentan en el panel *Incluir*.
- La casilla *Incrustar diagramas* se habilita cuando se elige el formato de salida MS Word o RTF. Si marca esta casilla, los diagramas se incrustan en el archivo de salida (en formato PNG o EMF). De lo contrario, los diagramas se crean en archivos PNG o EMF y aparecen en el archivo de salida como

© 2016-2022 Altova GmbH

enlaces de objeto. Si elige el formato de salida HTML, todos los diagramas se crean en archivos PNG fuera del documento.

- En el panel *Incluir* puede elegir qué componentes del esquema se documentan. Si marca la casilla *Vista general*, se documentarán todos los componentes y estos se ordenarán por tipos. Si no está marcada la casilla *Definiciones del esquema*, se deshabilitarán todos los componentes secundarios (es decir, todo menos *Esquemas externos*).
- El panel detalles incluye todos los detalles que se pueden documentar para los componentes. Si no está marcada la casilla *Definiciones*, entonces se deshabilitan todos los detalles.
- La casilla *Mostrar archivo resultante* se habilita con los tres formatos de salida. Si marca esta casilla, los archivos de salida se abren en la vista Explorador (si eligió el formato HTML), en MS Word (si eligió el formato MS Word) o en la aplicación predeterminada para archivos `.rtf` (si eligió el formato RTF).

13.7 Validar documentos JSON

XMLSpy incluye un motor de validación JSON que se puede invocar de dos maneras:

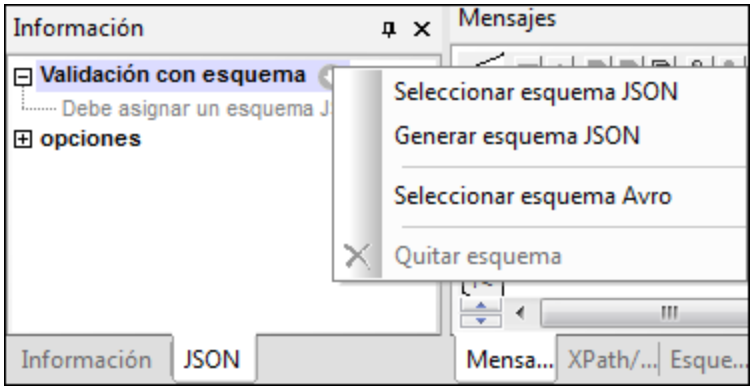
- *Si el documento activo es un esquema JSON:* valida el esquema JSON con la especificación JSON Schema adecuada (sin necesidad de tener asignado un esquema suplementario); la versión del esquema viene indicada por la palabra clave **`$schema`**[671]. La validación puede realizarse desde cualquier vista de edición (Texto[674], Cuadrícula[679] o Esquema JSON[714]).
- *Si el documento activo es una instancia JSON:* valida la instancia JSON con un esquema JSON. El esquema se asigna a la instancia JSON tal y como se explica más abajo. La validación de instancias JSON se puede realizar tanto en la vista Texto[674] como en la vista Cuadrícula[679].
- *Si el documento activo es una instancia JSON5:* valida la instancia JSON con un esquema JSON. El esquema se asigna a la instancia JSON5 tal y como se explica más abajo. La validación de instancias JSON5 se puede realizar tanto en la vista Texto[674] como en la vista Cuadrícula[679].

Validación de documentos Avro (solo en la edición Enterprise Edition)
Al ser documentos JSON, los documentos de datos Avro y esquemas Avro se pueden validar tanto en la vista Texto[674] como en la vista Cuadrícula[679]:

- *Si el documento activo es una instancia de datos Avro en formato JSON:* valida la instancia Avro con un esquema Avro. El esquema se asigna a la instancia tal y como se explica más abajo.
- *Si el documento activo es un esquema Avro:* valida el esquema Avro con la especificación Avro schema (no hace falta asignarle ningún esquema). La validación puede realizarse tanto en la en la vista Texto[674] como en la vista Cuadrícula[679].

Asignar un esquema JSON o Avro a una instancia JSON

Para poder validar una instancia JSON con un esquema JSON o Avro, el esquema debe estar asignado al documento de instancia activo. Esta asignación se introduce en la ventana Información (*imagen siguiente*) del documento de instancia activo o en el cuadro de diálogo Propiedades del proyecto[1328] (en el campo *Validar con*). Tenga en cuenta que los documentos de instancia JSON5 se validan con esquemas JSON.



En la pestaña *JSON* de la ventana Información haga clic en el icono en forma de flecha situado junto al campo **`Validación con esquema`** y seleccione **Seleccionar esquema JSON** o **Seleccionar esquema Avro** en el menú contextual (*imagen anterior*).. Busque el esquema y después haga clic en **Aceptar**. El esquema se asignará al documento de instancia activo y su nombre de archivo aparecerá en la ventana Información. Si el documento de instancia JSON está vacío, gracias a la asignación de esquema JSON o Avro se rellenará automáticamente la instancia JSON con datos de muestra generados a partir del esquema.

© 2016-2022 Altova GmbH

Para eliminar la asignación, seleccione el comando **Quitar esquema** del mismo menú (*imagen anterior*).

Para aprender a generar un esquema JSON a partir de una instancia JSON, consulte el apartado Generar esquemas JSON a partir de instancias JSON[762].

Validar documentos de instancia y esquema

Para validar el documento JSON activo (instancia o esquema) o el esquema Avro activo, seleccione el comando **XML | Validar XML (F8)** o haga clic en el icono **Validar (F8)**  de la barra de herramientas. Si se trata de un documento de instancia, debe tener asignado un esquema. Los resultados de la validación aparecen en la ventana Mensajes[125].
Los errores también se señalan en el margen de numeración de líneas. Si la aplicación dispone de una solución para un error, aparece un icono en forma de bombilla en la línea que genera el error. Si pasa el cursor por encima de este icono, aparecen todas las soluciones para el error en un menú contextual. Seleccione una solución para corregir el error inmediatamente.

Nota: los indicadores de errores de validación y las soluciones rápidas que se describen más arriba solo se actualizan si se ejecuta el comando **XML | Validar (F8)**, pero no se actualizan en segundo plano. Por tanto, una vez que corrija un error debe ejecutar el comando **Validar (F8)** de nuevo para asegurarse de que ese error se ha corregido.

Para ir al documento de esquema desde el documento de instancia, haga doble clic en el esquema en la ventana Información o seleccione el comando **DTD/Schema | Ir al esquema**. Para ir a la definición de esquema de una palabra clave o de un objeto JSON directamente, seleccione el elemento en el documento de instancia y haga clic en el comando **DTD/Schema | Ir a la definición**.

También puede validar una carpeta de proyecto[1060] que contenga archivos JSON con solo ejecutar el comando **Validar**.

Validar al modificar

El modo *Validar al editar*, está activado por defecto, permite validar un documento y comprobar si su formato es el correcto mientras lo modifica en la vista Cuadrícula JSON. Para validar un documento JSON (además de comprobar su formato) debe asignar un esquema JSON a ese documento. Si hay errores, el texto que los contiene aparece en rojo y el lugar del error se marca con un signo rojo de exclamación. Para más información consulte Validar documentos JSON[754].

El modo *Validar al editar* se puede activar y desactivar (i) con el comando de menú **XML | Validar al editar**[1344], (ii) el botón **Validar al editar** de la barra de herramientas o (iii) con la opción *Al editarlos* de las las opciones de validación del cuadro de diálogo "Opciones"[1584].

13.8 Insertar fragmentos JSON

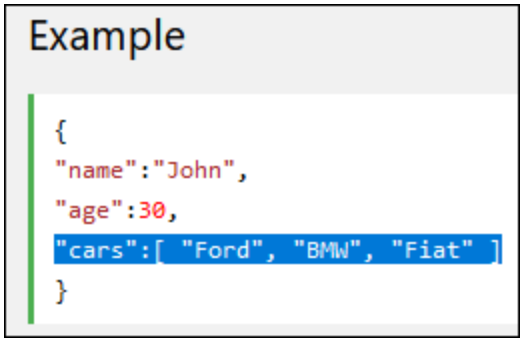
Puede insertar rápidamente fragmentos JSON de otras aplicaciones y sitios web. Hay dos maneras de insertar estos fragmentos:

- Con la función de arrastrar y colocar de la vista Texto o la vista Cuadrícula. Si arrastra y coloca en la vista Cuadrícula, la información inteligente de la capa de arrastre 689 le ayudará a saber dónde colocar el fragmento.
- Con la función de copiar y pegar de la vista Texto o la vista Cuadrícula.

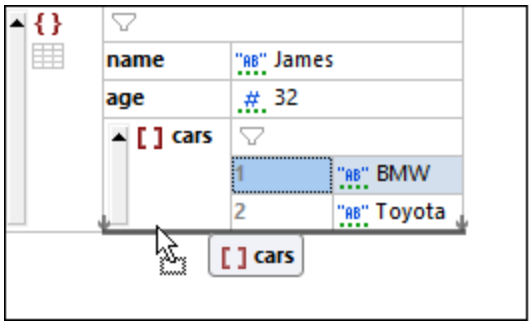
Ejemplo

En este ejemplo verá cómo añadir rápidamente un fragmento y cómo saber cuál es la ubicación correcta en un documento JSON.

1. Seleccione el fragmento resaltado que ve a continuación (que proviene del mismo tutorial JSON de w3schools). Se trata de una matriz llamada **cars**.

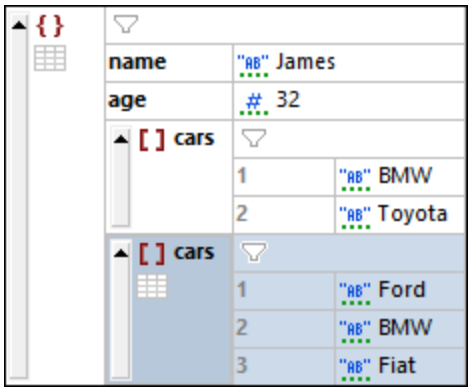


2. La imagen siguiente muestra la vista Cuadrícula de un documento JSON que contiene una matriz **cars** parecida. Al arrastrar el fragmento desde el sitio web hasta la matriz **cars** que ya existe aparece una capa de arrastre que contiene información del fragmento JSON que se colocará bajo la matriz como una matriz nueva llamada **cars**.



3. Al soltar el botón del ratón el fragmento se coloca exactamente donde quiere (*imagen siguiente*).

© 2016-2022 Altova GmbH


{ }
name
"AB" James
age
32
[ ] cars
1
"AB" BMW
2
"AB" Toyota
[ ] cars
1
"AB" Ford
2
"AB" BMW
3
"AB" Fiat

13.9 Transformaciones JSON con XSLT/XQuery

Las expresiones **XPath/XQuery 3.1** se pueden utilizar con asignaciones, conjuntos u objetos JSON. El resultado es que se pueden transformar documentos JSON con documentos **XSLT 3.0, XQuery 3.1** y **XQuery Update 3.0** usando XMLSpy.

Esto nos permite:

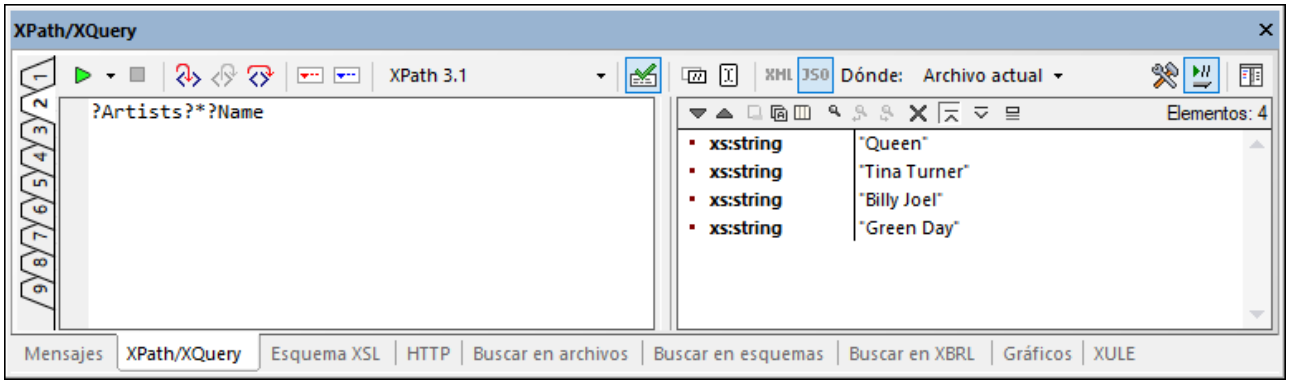
- Un documento JSON activo se puede consultar con expresiones XPath/XQuery 3.1 758 desde la ventana de resultados XPath/XQuery 127
- Un documento JSON activo se puede transformar con un archivo XSLT o XQuery 759
- Un archivo XSLT o XQuery activo se puede ejecutar en un archivo de origen JSON 759

Estas funciones se describen con más detalle a continuación. Para más información sobre cómo construir expresiones XQuert para documentos JSON consulte la sección Expresiones XQuery para JSON 760.

Nota: puede familiarizarse con las transformaciones JSON usando los archivos JSON, XSLT y XQuery que encontrará en la carpeta *Examples*, en el proyecto **Examples**, que encontrará en la carpeta de la aplicación: `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Examples.spp`.

Consultar un documento JSON desde la ventana de resultados XPath/XQuery

Se pueden consultar documentos JSON con expresiones XPath/XQuery 3.1 en la ventana ventana de resultados XPath/XQuery 127 (*imagen siguiente*). Seleccione el icono **Xpath 3.1** o el icono **XQuery 3.1** y asegúrese de que la ventana se encuentra en modo de evaluación JSON (*véase el siguiente apartado*).



El siguiente apartado explica la evaluación de documentos JSON en modo de evaluación JSON. (Para más información sobre la ventana XPath/XQuery y su uso, consulte Ventana de resultados: XPath/XQuery 127.)

Modo de evaluación JSON
El modo de evaluación JSON consiste en los siguientes puntos:

- La ventana XPath/XQuery puede estar en modo de evaluación XML o en modo de evaluación JSON. El botón de modo de evaluación muestra qué modo está **activo** en ese momento. En la imagen anterior la ventana está en modo de evaluación JSON.
- Observe que en la imagen anterior los botones de modo de evaluación XML/JSON aparecen en gris, lo que indica que están **deshabilitados**. Cuando los botones están deshabilitados no se pueden activar

© 2016-2022 Altova GmbH

ni desactivar. Por el contrario, si los botones están habilitados (no aparecen en gris), entonces se puede cambiar el modo de evaluación de la ventana.

- El estado *habilitado/deshabilitado* de los botones de modo de evaluación XML/JSON depende del alcance de la evaluación (el valor del campo Dónde; *véase imagen anterior*). En lo que respecta al estado *habilitado/deshabilitado* de los botones de modo de evaluación, los valores de alcance de la evaluación se dividen en dos grupos: (i) archivo único (*archivo actual*) y (ii) múltiples archivos (*archivos abiertos, proyecto, carpeta*).
- Si se selecciona un único archivo para el alcance de la evaluación (*imagen anterior*), entonces el modo de la ventana (JSON o XML) se determina en base a la extensión del archivo[1587]. Si el archivo es conforme a JSON[1587] se **activa** el modo de evaluación JSON; si el archivo no es conforme a JSON[1587] se activa el modo de evaluación XML. Como conocemos el tipo de archivo, se activa el modo de evaluación correcto y se **deshabilitan** ambos botones para que no sea posible cambiar el modo de evaluación.
- Si se selecciona la opción de archivos múltiples (a*rchivos abiertos, proyecto, carpeta*), entonces los dos botones de evaluación quedan **habilitados** y es posible activar un modo u otro (JSON o XML). El modo de evaluación predeterminado para un alcance de evaluación de archivos múltiples es XML.
- En modo de evaluación XML, se procesarán los archivos conforme a XML[1587] y se ignorarán los archivos JSON.
- En modo de evaluación JSON, se procesarán los archivos conforme a JSON[1587] y se ignorarán los archivos XML.
- Las expresiones JSON también pueden consultarse en modo depuración[582].

Transformar un documento JSON con XSLT/XQuery

Para transformar un documentos JSON activo[119] con XSLT 3.0, XQuery 3.1 o XQuery Update 3.0 haga lo siguiente:

- *Transformación XSLT 3.0:* haga clic en **XSL/XQuery | Transformación XSL**, busque el archivo XSLT 3.0 y haga clic en **Aceptar**.
- *Transformación XQuery 3.1 or XQuery Update 3.0:* haga clic en **XSL/XQuery | Ejecutar XQuery Update**, busque el archivo XQuery 3.1 or XQuery Update y haga clic en **Aceptar**.

Esto generará el archivo transformado, que se puede visualizar directamente en XMLSpy.

Nota: el depurador XSLT/XQuery[531] se puede iniciar desde un documento JSON, pero los puntos de interrupción y de seguimiento solamente se pueden establecer en el documento XSLT o XQuery.

Archivos JSON como origen para un documento XSLT/XQuery

Para ejecutar un documento activo XSLT o XQuery[119] en un archivo de origen JSON, haga lo siguiente:

- *Documento Active XSLT 3.0:* haga clic en **XSL/XQuery | Transformación XSL**, busque el archivo JSON y haga clic en **Aceptar**.
- *Documento Active XQuery 3.1 or XQuery Update 3.0:* haga clic en **XSL/XQuery | Ejecutar XQuery Update**, busque el archivo JSON y haga clic en **Aceptar**.

Esto generará el documento transformado, que se puede visualizar directamente en XMLSpy.

Nota: el depurador XSLT/XQuery[531] se puede iniciar desde un documento XSLT o XQuery y se puede asignar un documento JSON como entrada para la sesión de depuración. Sin embargo, los puntos de interrupción y de seguimiento solamente se pueden establecer en el documento XSLT o XQuery.

13.10 Expresiones XQuery para JSON

Expresiones XQuery 3.1 para JSON

Las estructuras de datos JSON suelen usar objetos y matrices y usan el operador de búsqueda XQuery 3.1 **?**, que se usa para ubicar nodos dentro de objetos (que básicamente son asignaciones desde la perspectiva de XQuery) y matrices JSON. Esta forma de ubicar nodos es distinta a la que usa expresiones de ruta de acceso para ubicar nodos en documentos XML. Es este caso, el operador barra **/** se usa para conectar pasos en una expresión de ruta de acceso (por ejemplo: **items/***). En las expresiones XQuery para JSON, el operador barra no se usa para ubicar nodos.

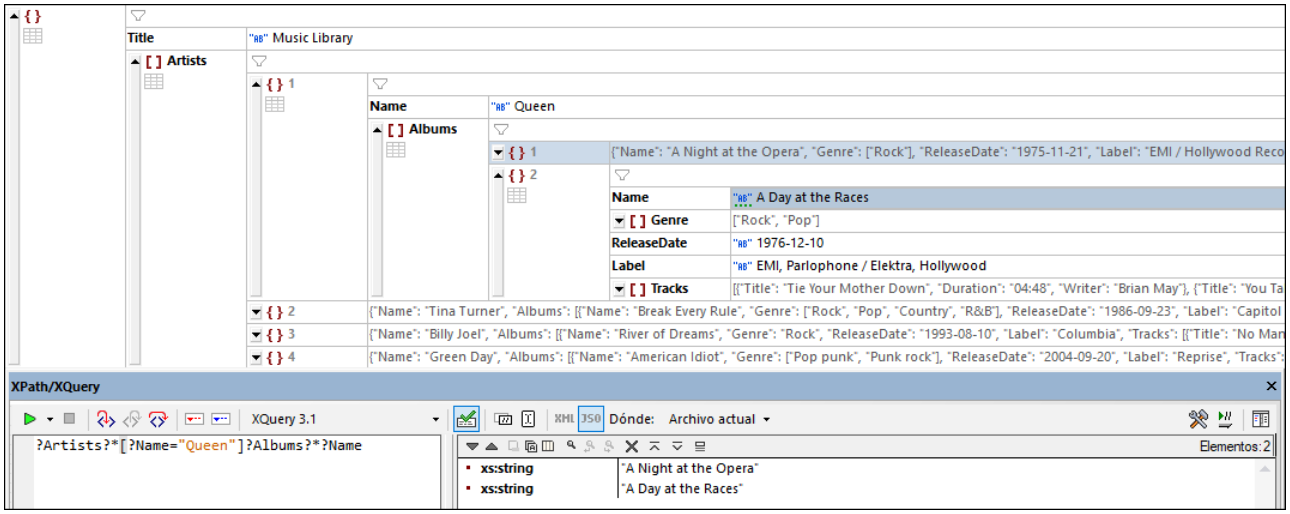
Ejemplos de expresiones XQuery para JSON

?items?*
Significa: buscar el nodo secundario **items** y después buscar todos sus nodos secundarios. Observe que se espera que **items** sea un nodo secundario del nodo de contexto.

?Artists?1?Albums?2?Name
Significa: buscar el nodo secundario **Artists** y después buscar su primer nodo secundario. Dentro de ese nodo, buscar el nodo secundario **Albums** y después buscar el segundo nodo secundario de este. Devolver el nodo **Name** de ese segundo nodo secundario.

?Tracks?*[contains(?Writer, 'Brian')]
Significa: buscar el nodo secundario **Tracks** y después buscar todos sus secundarios. Mientras se buscan los secundarios, buscar los nodos secundarios de cada secundario **Writer** y seleccionar solamente los que contengan la cadena de texto **'Brian'**. Observe que en esta expresión hay tres operadores de búsqueda. Cada uno se usa en un paso nuevo en el que se busca en un conjunto de nodos.

?Artists?*[?Name="Queen"]?Albums?*?Name
Significa: dentro del objeto raíz, buscar el nodo secundario **Artists** y después buscar todos sus secundarios cuyo nodo **Name** tiene el valor **"Queen"**. Dentro de estos nodos, buscar todos los nodos secundarios **Album** y sus secundarios. Dentro de esos secundarios, buscar (y devolver) los respectivos nodos **Name**.En la imagen siguiente esta expresión se muestra en la ventana XPath/XQuery 573 junto con la representación de la vista Cuadrícula JSON 679 del documento JSON de destino.



 © 2016-2022 Altova GmbH

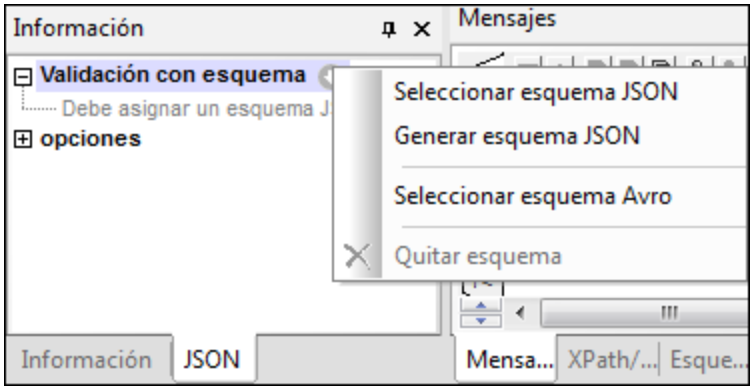
13.11 Generar esquemas JSON a partir de instancias JSON

XMLSpy puede generar esquemas JSON a partir de documentos de instancia JSON y de instancias JSON5. Gracias a esta característica ya no es necesario crear el esquema a mano y desde cero. Tras generar el esquema, podrá modificarlo o ampliarlo según sus requisitos.

Generar el esquema

Siga estos pasos para generar un esquema a partir de una instancia JSON:

1. El documento activo debe ser el documento de instancia JSON.
2. En la pestaña *JSON* de la ventana Información, haga clic en el icono en forma de flecha situado junto a *Validación con esquema* y, en el menú contextual, haga clic en el comando **Generar esquema JSON** (*imagen siguiente, comandos relacionados con Avro solamente disponibles en la edición Enterprise Edition*). Si lo prefiere, también puede seleccionar el comando de menú **DTD/Schema | Generar DTD o esquema**.

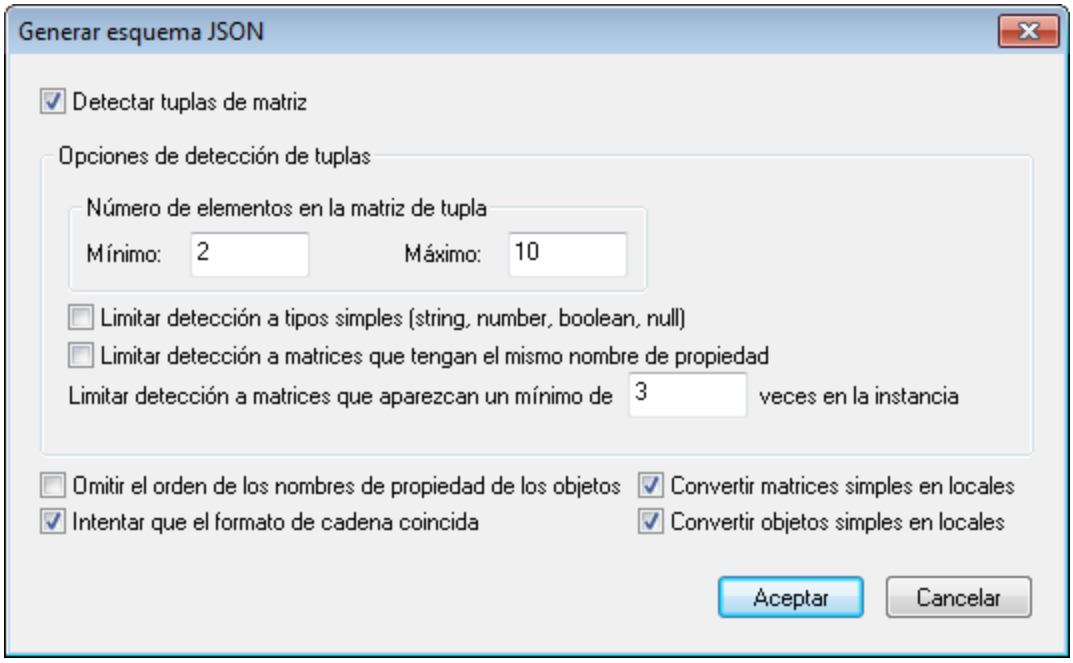


3. Ahora aparece el cuadro de diálogo "Generar esquema JSON" (*imagen siguiente*). Aquí puede indicar cómo se lleva a cabo la generación. Haga clic en **Aceptar**.
4. La aplicación solicita una ruta de acceso y nombre de archivo para el esquema JSON que se generará. Cuando haga clic en **Guardar** el esquema se generará y será el documento activo de la vista. El archivo de esquema generado se asignará al archivo de instancia, sobrescribiendo cualquier asignación realizada con anterioridad. Para cambiar la asignación, haga clic en el comando **Seleccionar esquema JSON** 754 del menú contextual (*imagen anterior*).

Configuración para generar el esquema JSON

Puede configurar las opciones de generación de esquemas JSON en el cuadro de diálogo "Generar esquema JSON" (*imagen siguiente*). Consulte la sección previa 762 para más información sobre cómo acceder a este cuadro de diálogo.

© 2016-2022 Altova GmbH



Detectar tuplas de matriz

Una tupla de matriz es la secuencia de ítems de una matriz. Por ejemplo, esta matriz tiene una tupla de tres ítems: `[ 1, 2, "abc" ]`. Para la validación de matrices el esquema puede especificar si debe tenerse en cuenta o no el orden y el tipo de datos de los ítems de las matrices (tuplas). Si marca la casilla *Detectar tuplas de matriz*, entonces se tendrá en cuenta el orden y el tipo de datos de los ítems. Dependiendo de lo que se detecte, se creará una definición en el esquema. Esta definición puede configurarse con estas opciones:

- *Número de elementos de la tupla:* puede especificar un máximo y un mínimo de elementos de tupla. Si una tupla de la instancia tiene un número de elementos comprendido entre este máximo y mínimo, la matriz se detecta y se define.
- *Limitar detección a tipos simples:* solamente se tendrán en cuenta las tuplas que tengan ítems de tipo simple (tipos atómicos `string`, `number`, `integer`, `boolean` y `null`).
- *Limitar detección a matrices que tengan el mismo nombre:* solamente se tendrán en cuenta las matrices que se definan como valores de propiedades que tengan el mismo nombre. Por ejemplo, en este fragmento de datos JSON, las matrices marcadas en rojo son valores de propiedades llamadas `a1` (marcadas con fondo azul): `{"object1": [ { "a1": [ 1, 2, "abc" ] }, { "a1": [ 3, 4, "def" ] }, { "a1": [ 5, 6, "ghi" ] } ] }`.
- *Número mínimo de matrices:* puede especificar el número mínimo de matrices necesarias para la detección de matrices.

Otras opciones

- *Omitir el orden de los nombres de propiedad de los objetos:* si no marca esta opción, se comprueba el orden de las propiedades de los objetos y se recrea de la forma lo más exacta posible. De lo contrario, no se comprueba el orden.
- *Intentar que el formato de cadena coincida:* el esquema puede especificar que los tipos de datos string tengan un formato concreto. Si marca esta opción, XMLSpy intentará detectar el formato de cadena y en la medida de lo posible añadirá una definición de formato a las cadenas.

- *Convertir matrices simples en locales:* una matriz simple es aquella cuyos ítems tienen el mismo tipo de datos simple. Si marca esta opción, todas las matrices simples se definirán localmente en el esquema, en lugar de usar definiciones globales a las que se hace referencia localmente.
- *Convertir objetos simples en locales* un objeto simple es aquél cuyos valores de propiedades tienen el mismo tipo de datos simple. Si marca esta opción, todos los objetos simples se definirán localmente en el esquema, en lugar de usar definiciones globales a las que se hace referencia localmente.

Nota: después de generar el esquema JSON puede convertir en globales las definiciones locales de los objetos y matrices. Para más información consulte el apartado Definiciones globales y locales 721.

 © 2016-2022 Altova GmbH

13.12 Generar instancias JSON a partir de esquemas JSON

Si el archivo activo de la vista Texto, Cuadrícula o Esquema JSON es un esquema JSON, haga clic en el comando **DTD/Schema | Generar archivo XML/JSON de muestra** 1365 para generar un documento de instancia JSON. No olvide que este comando genera un documento JSON, pero no un documento JSON5.

13.13 Convertir datos JSON en XML y viceversa

XMLSpy ofrece estas funciones de conversión, que pueden invocarse desde la vista Texto y Cuadrícula:

- Convertir instancia XML en JSON:[1476] si el documento activo es un documento de instancia XML, haga clic en el comando **Conversión | Convertir instancia XML en JSON y viceversa**[1476] para crear un documento de instancia JSON o JSON5 a partir de la instancia XML.
- Convertir instancia JSON en XML[1476]: si el documento activo es un documento de instancia JSON, haga clic en el comando **Conversión | Convertir instancia XML en JSON y viceversa**[1476] para crear un documento de instancia XML a partir de la instancia JSON.
- Convertir esquema XML en esquema JSON[1480]: si el documento activo es un esquema XML, haga clic en el comando **Conversión | Convertir esquema XML en esquema JSON y viceversa**[1480] para crear un esquema JSON a partir del esquema XML.
- Convertir esquema JSON en esquema XML[1480]: si el documento activo es un esquema JSON, haga clic en el comando **Conversión | Convertir esquema XML en esquema JSON y viceversa**[1480] para crear un esquema XML a partir del esquema JSON.

© 2016-2022 Altova GmbH

14 Avro y Avro Schema

Apache Avro™ es un sistema para serializar datos en un formato binario compacto. Una estructura de datos Avro se define en un esquema Avro, que está escrito en formato JSON. A la hora de la implementación, un documento Avro se suele serializar como archivo binario que contiene no sólo las estructuras de datos Avro, sino también el esquema Avro que se usó para definir dichas estructuras. El binario Avro, por tanto, contiene tanto los datos como la definición de la estructura de datos (el esquema Avro). Sin embargo, los datos Avro también se pueden serializar como JSON. En este caso los datos Avro (incluidos en un archivo JSON) hacen referencia a un esquema Avro externo.

XMLSpy es compatible con Apache Avro™ 1.8.1.

XMLSpy ofrece todas estas funciones para trabajar con Avro:

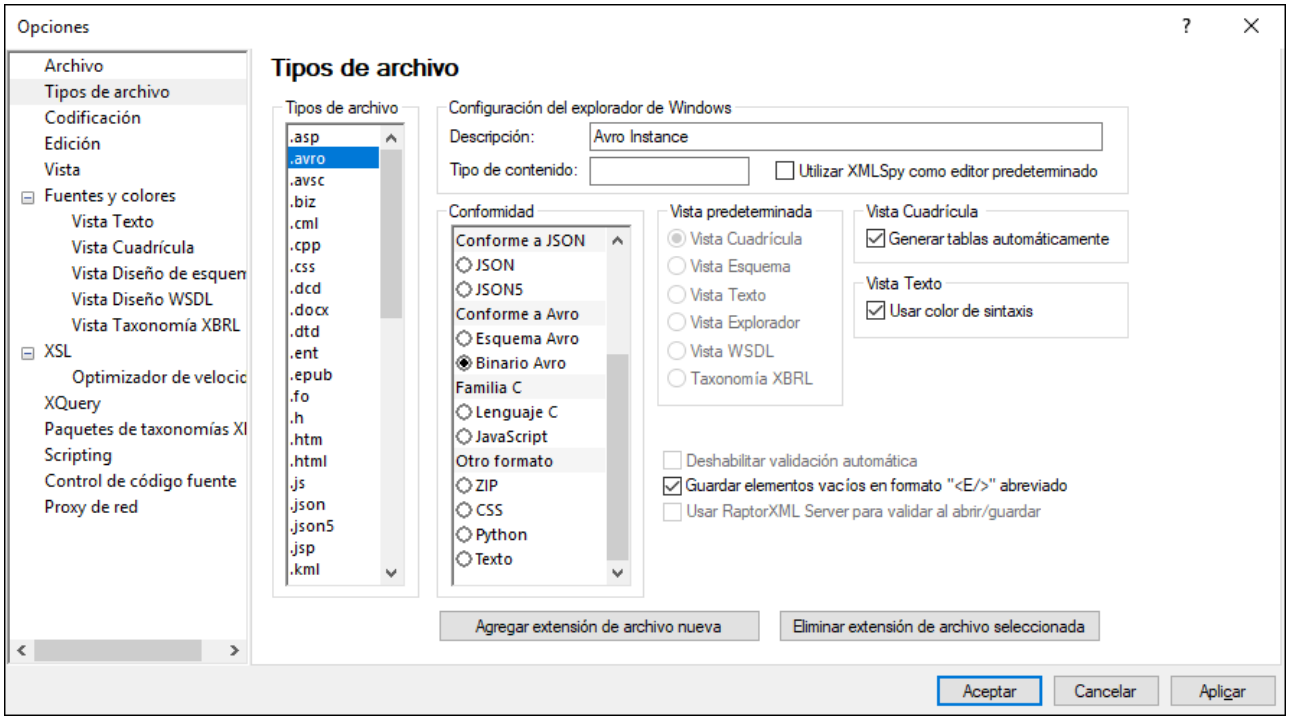
- Edición de datos Avro (como documentos JSON **.json**) en las vistas Texto[674] y Cuadrícula[679]. Ambas vistas incluyen funciones de edición inteligente. El documento de datos puede tener asignado un esquema Avro y se puede validar en estas vistas.
- Edición de esquemas Avro (como documentos Avro Schema **.avsc**) en las vistas Texto[674] y Cuadrícula[679]. Los esquemas Avro se pueden validar con la especificación Avro schema y las vistas ofrecen funciones de edición inteligentes.
- Consulta de instancias binarias Avro (archivos **.avro**) en la vista Avro[950], que presenta los bloques de datos Avro en formato de tabla.

El software servidor de Altova RaptorXML también incluye funciones para trabajar con Avro:

- Validación de datos Avro (serializados como JSON, archivos `.json`) con un esquema Avro.
- Validación de datos Avro (serializados como binarios, archivos `.avro`).
- Validación de esquemas Avro (por lo general archivos `.avsc`) con la especificación Avro schema.
- Extracción de esquemas Avro de binarios Avro.

Abrir documentos Avro y crear documentos Avro nuevos

En la sección[1587] *Tipos de archivo*[1587] del cuadro de diálogo "Opciones"[1587] (*imagen siguiente*) puede configurar en qué vista predeterminada se abren los documentos de tipo Avro (formato de datos JSON, esquemas Avro y binarios Avro). Esta configuración se puede modificar en todo momento.



Tipo de documento	Extensión de archivo	Conformidad	Vistas disponibles
Datos Avro en formato JSON	`.json`	Conforme a JSON | JSON	Texto y Cuadrícula
Esquema Avro	`.avsc`	Conforme a Avro | Avro Schema	Texto y Cuadrícula
Datos Avro en archivo binario	`.avro`	Conforme a Avro | Binarios Avro	Cuadrícula (Avro)

Debe tener en cuenta estas características:

- Todos los documentos (tanto actuales como nuevos) de un tipo seleccionado se abrirán en la vista predeterminada que seleccione en la sección *Tipos de archivo* del cuadro de diálogo "Opciones".
- Los archivos binarios Avro solamente se pueden ver en la vista Avro[773], que es una vista de solo lectura. Cuando un tipo de archivo se define como conforme a Avro, la única vista disponible será la vista Avro[773].
- Si desea que XMLSpy pueda leer archivos de cierta extensión de archivo como si fuera un tipo de documento Avro de los descritos más arriba, basta con añadir esta nueva extensión de archivo a la lista de tipos de archivos y asignarle la conformidad correspondiente.
- Para crear un documento nuevo haga clic en Archivo **| Nuevo** y seleccione el tipo de documento correspondiente. Huelga decir que los binarios Avro no se pueden crear de esta manera (porque son binarios). En XMLSpy solamente se pueden leer binarios (en la vista Avro[773]).

© 2016-2022 Altova GmbH

14.1 Esquemas Avro

Un esquema Avro especifica la estructura de un bloque de datos Avro. Al igual que los esquemas JSON[671], los esquemas Avro especifican cómo se organizan los datos JSON en un documento JSON. Especifican qué campos de datos deben esperarse y cómo se representan los valores. Aquí encontrará información sobre los esquemas Avro y su especificación.

Debe tener en cuenta estos aspectos de los esquemas Avro:

- Un esquema Avro se crea en formato JSON.
- Un esquema Avro puede ser una cadena JSON, un objeto JSON o una matriz JSON.
- Un esquema Avro puede contener cuatro atributos: **name**, **namespace**, **type** y **fields**.
- Hay ocho tipos de datos primitivos: **null**, **boolean**, **int**, **long**, **float**, **double**, **bytes** y **string**.
- Hay seis tipos de datos complejos: **records**, **enums**, **arrays**, **maps**, **unions** y **fixed**.
- Los tipos primitivos no tienen atributos. Cada tipo complejo tiene su propio conjunto de atributos.

Para más información consulte la especificación Avro Schema.

Ejemplos

A continuación damos algunos ejemplos sencillos de esquemas Avro, acompañados de sus correspondientes ejemplos de instancia JSON. Recuerde que el esquema define una estructura determinada. En algunos casos, cuando la estructura definida se usa con instancias varias veces, puede que el resultado obtenido no sea código JSON válido. Por ejemplo, un esquema puede definir la estructura de un objeto JSON. Si el objeto JSON se usa con instancias varias veces, cada objeto puede ser válido según el esquema Avro, pero el documento entero no sería código JSON válido porque no existe un objeto contenedor. Para que el código JSON sea válido, sería necesario volver a escribir el esquema Avro para validar una matriz de objetos JSON. Esto se entiende mejor con los ejemplos nº4 y 5.

- *Ejemplo nº1: esquema Avro como cadena JSON*

  Este esquema es una sola cadena de texto que especifica que el documento de instancia debe contener un valor que es del tipo de datos Avro primitivo `"int"`

  Ejemplo de código JSON válido: `2016`
  Ejemplo de código JSON no válido: `"2016"`

- *Ejemplo nº2: esquema Avro como objeto JSON*

  Este esquema especifica exactamente lo mismo que el anterior, pero es un objeto JSON. El documento de instancia debe contener un elemento que es un valor del tipo de datos Avro primitivo:

```
{
   "type": "int"
}
```

  Ejemplo de código JSON válido: `2016`
  Ejemplo de código JSON no válido: `"2016"`

- *Ejemplo nº3: esquema Avro como objeto JSON: matriz de enteros*

Este esquema es un objeto JSON que especifica una matriz de enteros:

```
{
   "type": "array",
   "items": "int"
}
```

Ejemplo de código JSON válido: `[2016, 2017]`
Ejemplo de código JSON válido: `[2016]`
Ejemplo de código JSON válido: `[2016]`
Ejemplo de código JSON no válido: `2016, 2017`

- *Ejemplo nº4: esquema Avro como objeto JSON: registro*

Este esquema es un objeto JSON que especifica un solo registro:

```
{
   "type": "record",
   "name": "ages",
   "fields" : [
   {"name": "name", "type": "string"},
   {"name": "age", "type": "int"}
   ]
}
```

Ejemplo de código JSON válido: `{"name":"John", "age":35}`

- *Ejemplo nº5: esquema Avro como objeto JSON: varios registros*

Este esquema es un objeto JSON que especifica una matriz de registros y cada uno de ellos debe ser un objeto JSON:

```
{
   "type": "array",
   "items": {
   "type": "record",
   "name": "ages",
   "fields" : [
   {"name": "name", "type": "string"},
   {"name": "age", "type": "int"}
   ]
    }
}
```

Ejemplo de código JSON válido: `[{"name":"Mary", "age":34}, {"name":"John", "age":35}]`

Tipos de archivo de esquema Avro

Para poder usar las funciones de edición y validación de esquemas Avro de XMLSpy, la aplicación debe ser capaz de reconocer que un archivo es un esquema Avro. Para ello es necesario definir la extensión del archivo como extensión de archivo de esquema Avro y esto se hace en el cuadro de diálogo "Opciones" (**Herramientas | Opciones | Tipos de archivo**[1587]). La configuración predeterminada de XMLSpy define una extensión de archivo[767] como extensión de archivo de esquema Avro: la extensión `.avsc`. Si desea crear otras extensiones de archivo que especifiquen documentos de esquema Avro[767], añádalas a la lista de extensiones de esquemas Avro en el cuadro de diálogo "Opciones"[1587].

© 2016-2022 Altova GmbH

Crear y editar esquemas Avro

En XMLSpy puede crear archivos nuevos 1258 como esquemas Avro con solo elegir la extensión de archivo de esquema Avro como extensión del archivo. Mientras teclea, XMLSpy ofrece ayuda contextual, como la entrada automática de pares de paréntesis, corchetes, llaves y comillas, color de sintaxis y finalización automática de palabras clave. Además, ofrece tres prácticos ayudantes de entrada: las ventanas Propiedades JSON, Valores JSON y Entidades JSON. Las entradas que aparecen en estas ventanas no son contextuales. Haga doble clic en una entrada para introducirla en el documento (en la posición actual del cursor). Después puede validar el archivo con la especificación Avro schema specification con el comando de menú **XML | Validar el documento XML (F8)**.

14.2 Datos Avro en formato JSON

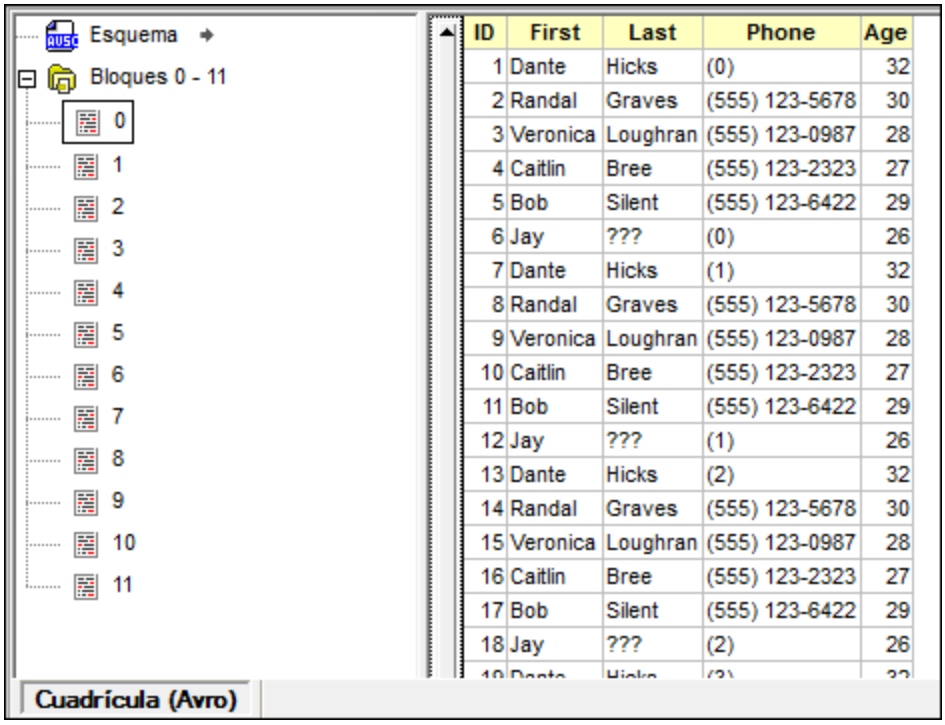
Los datos Avro se pueden serializar en formato binario o formato JSON. Estas son las características generales de las funciones de XMLSpy compatibles con este formato Avro:

- Los datos Avro que están en formato JSON se suelen guardar con la extensión de archivo `.json`. Puede configurar XMLSpy para que también reconozca otras extensiones de archivo como conformes a JSON 767.
- Los archivos Avro JSON se pueden abrir y editar en las vistas Texto 674 y Cuadrícula 679.
- Puede asignar un archivo de esquema Avro al archivo Avro JSON 754 y el archivo de datos se podrá validar con dicho esquema Avro 754.
- Las funciones de edición inteligentes para documentos JSON están disponibles en las vistas Texto 674 y Cuadrícula 679. Además, si el documento de datos Avro en formato JSON tiene asignado un esquema Avro, también estará disponible la función de finalización automática de palabras clave definidas por el esquema 674.

© 2016-2022 Altova GmbH

14.3 Vista Avro: vista cuadrícula de binarios Avro

Los datos Avro se pueden serializar en formato binario o en formato JSON. El formato binario contiene tanto las estructuras de datos Avro como su esquema y se suele generar mediante procesos automatizados de procesamiento de datos. Un binario Avro se puede abrir en la vista Avro de XMLSpy, una vista en forma de cuadrícula que muestra las estructuras de datos en forma de tabla (*imagen siguiente*). Es decir, la vista Avro es un práctico y sencillo visor de binarios Avro.



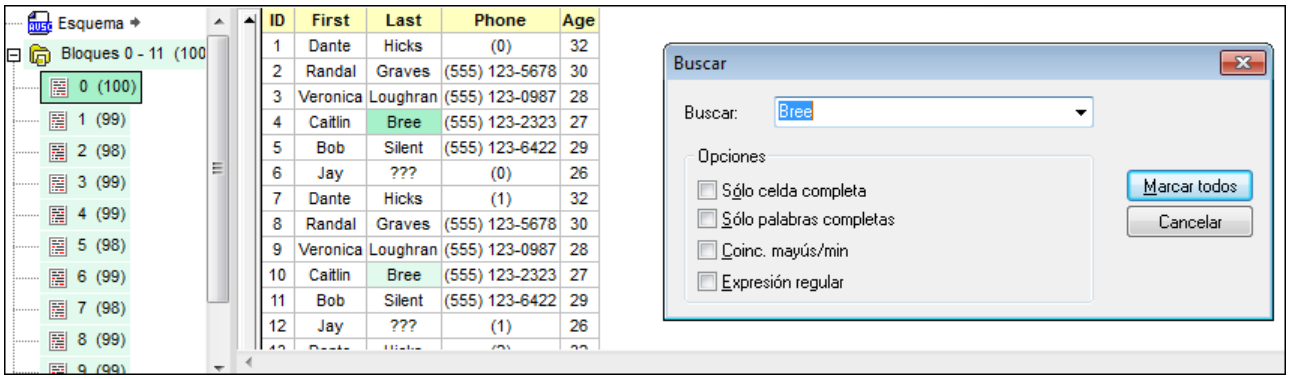
ID	First	Last	Phone	Age
1	Dante	Hicks	(0)	32
2	Randal	Graves	(555) 123-5678	30
3	Veronica	Loughran	(555) 123-0987	28
4	Caitlin	Bree	(555) 123-2323	27
5	Bob	Silent	(555) 123-6422	29
6	Jay	???	(0)	26
7	Dante	Hicks	(1)	32
8	Randal	Graves	(555) 123-5678	30
9	Veronica	Loughran	(555) 123-0987	28
10	Caitlin	Bree	(555) 123-2323	27
11	Bob	Silent	(555) 123-6422	29
12	Jay	???	(1)	26
13	Dante	Hicks	(2)	32
14	Randal	Graves	(555) 123-5678	30
15	Veronica	Loughran	(555) 123-0987	28
16	Caitlin	Bree	(555) 123-2323	27
17	Bob	Silent	(555) 123-6422	29
18	Jay	???	(2)	26

Debe tener en cuenta estos aspectos de la vista Avro:

- XMLSpy debe ser capaz de reconocer el binario Avro. Para ello es necesario configurar la extensión del archivo binario como archivo conforme con Avro[767] en la sección[1587] *Tipos de archivo*[1587] del cuadro de diálogo "Opciones"[1587]. La extensión de archivo `.avro` está configurada por defecto como extensión conforme con Avro. Si quiere, puede añadir más tipos de archivo como conformes con Avro[767] para poder abrirlos en la vista Avro.
- La vista Avro está formada por dos paneles: (i) el panel *Bloques* sirve para navegar por el binario y (ii) el panel *Datos* muestra la estructura de datos que seleccione en el panel *Bloques*.
- El panel *Bloques* organiza los bloques de datos por grupos de `1000`. Cada grupo se puede expandir y contraer. Los bloques de datos se enumeran según su número de índice.
- Para ver un bloque de datos particular, basta con buscarlo en el panel *Bloques* y hacer doble clic.
- El panel *Bloques* también contiene una entrada llamada `Schema`. Haga clic en el botón de flecha situado junto a esta entrada para extraer el esquema Avro del binario Avro y abrirlo en una pestaña nueva de la vista Texto. Desde esta pestaña podrá guardar el esquema Avro en disco.

Búsquedas de texto

Para buscar una cadena de texto en la vista Avro seleccione el comando de menú **Edición | Buscar (Ctrl+F)**. Esto abre un cuadro de diálogo donde puede introducir el término de búsqueda como cadena de texto o como expresión regular. Seleccione los criterios de búsqueda correspondientes (*ver descripción*[1289]) y haga clic en **Marcar todos**.



- Los resultados de la búsqueda se resaltan en los dos paneles de la vista Avro. El resultado seleccionado se marca en verde oscuro, el resto en verde claro.
- En el panel *Bloques* se indica el número de resultados encontrados en cada bloque.
- Para navegar por los resultados basta con ir hasta un bloque, seleccionar uno de sus campos y después usar **F3** (**Edición | Buscar siguiente**) y **Mayús+F3** (**Edición | Buscar anterior**).
- Recuerde que la vista Avro es una vista de solo lectura donde no se pueden editar datos.

© 2016-2022 Altova GmbH

15 WSDL y SOAP

En esta sección describimos las funciones WSDL y SOAP de XMLSpy.

Sitio web de Altova: Editor WSDL, Cliente y depurador SOAP

WSDL

Un documento WSDL es un documento XML que describe un servicio web. XMLSpy es compatible con WSDL 1.1 y WSDL 2.0. Puede crear y editar documentos WSDL 1.1 y WSDL 2.0 en la vista WSDL de XMLSpy, que ofrece automáticamente el entorno de edición correspondiente a la versión WSDL que se esté editando.

En la vista WSDL de XMLSpy puede construir documentos WSDL usando bloques gráficos, lo cual simplifica considerablemente la creación de este tipo de documentos. La vista WSDL 251 se describe en la sección Vistas de edición 141. El *Tutorial de WSDL* 776 ofrece una descripción práctica sobre cómo crear un documento WSDL. También puede ver y editar documentos WSDL en las vistas Texto 145 y Cuadrícula 169. En estas vistas los documentos WSDL se editan como si fuesen documentos XML 285 normales y corrientes.

En la vista WSDL puede crear firmas XML 411 para los archivos WSDL (tanto como firma *envuelta* e incrustada en el documento como en un archivo externo). Para más información consulte la sección Firmas XML 411.

SOAP

SOAP es una especificación de intercambio de mensajes XML que se usa para enviar mensajes de una aplicación a otra. En XMLSpy no sólo puede crear y editar documentos SOAP en la vista Texto 145 y Cuadrícula 169 (con las mismas funciones de edición inteligente de XMLSpy que sirven para trabajar con documentos XML 285). También puede generar un archivo de solicitud SOAP a partir de un archivo WSDL. El *Tutorial de WSDL* 776 describe cómo se genera una solicitud SOAP a partir de un archivo WSDL. XMLSpy puede enviar y recibir solicitudes SOAP (con los comandos del menú SOAP 1504). Asimismo, puede depurar solicitudes SOAP con ayuda del depurador SOAP 788 de XMLSpy, que se explica en un apartado de esta sección.

15.1 Tutorial de WSDL

Este tutorial se divide en dos partes:

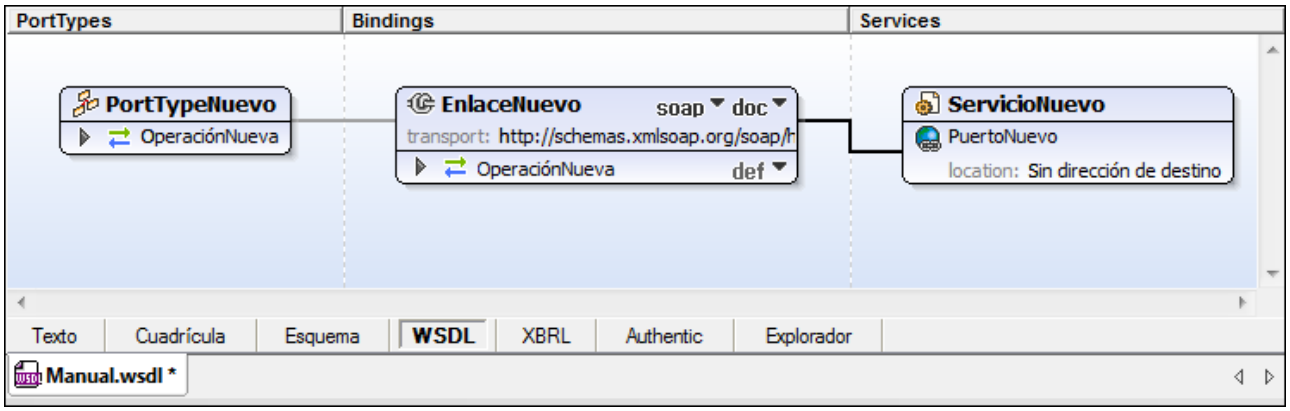
- En la primera parte explicamos cómo se crea un documento WSDL 1.1 en la vista WSDL gráfica de XMLSpy. En esta primera parte aprenderá (i) a crear un documento WSDL básico con la opción de menú **Archivo | Nuevo,** (ii) a crear un portType, (iii) a crear un enlace, (iv) a crear un servicio y un puerto, y (v) a validar y guardar el documento.
- En la segunda parte del tutorial explicamos cómo se establece una conexión con un servicio web, cómo se guarda el archivo WSDL localmente y cómo se envía una solicitud SOAP al servicio web.

Todas estas tareas se pueden llevar a cabo en la vista WSDL 251 gráfica sin necesidad de usar la vista Texto. En la vista WSDL puede manipular los componentes WSDL mediante operaciones de arrastrar y colocar. Y los valores de las propiedades se introducen en los ayudantes de entrada.

Véase también: encontrará más información sobre cómo trabajar con documentos WSDL en las secciones Vista WSDL 251 y Comandos de menú | Menú WSDL 1492.

15.1.1 Crear un documento nuevo

Para crear un documento WSDL nuevo, seleccione el comando **Archivo | Nuevo**. Aparece el cuadro de diálogo "Crear documento nuevo". Seleccione la opción `WSDL` (WSDL Web Service Description v 1.1) y haga clic en **Aceptar**. Como resultado se crea un documento nuevo base (*ver imagen siguiente*) que se abre en la vista WSDL gráfica (en adelante *vista WSDL*).



Asignar un espacio de nombres de destino

Cambie a la vista Texto. La etiqueta inicial del elemento `wsdl:definitions` tiene este aspecto:

© 2016-2022 Altova GmbH

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
3       xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
4       xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
5       xmlns:xs="http://www.w3.org/2001/XMLSchema"
6       xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
7       xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
8       xmlns:tns="http://new.webservice.namespace"
9       targetNamespace="http://new.webservice.namespace">
10
```

Cambie el atributo `targetNamespace` por `http://mywebservice.namespace` o cualquier otro texto (este tutorial explica cómo crear un documento WSDL, pero no ofrece ningún servicio así que puede usar el valor que quiera para este atributo). También es necesario cambiar el valor de espacio de nombres del atributo `tns` por `http:://mywebservice.namespace` (o por el espacio de nombres que usó para el atributo `targetNamespace`).

Nota: en el documento base inicial los elementos WSDL están en el espacio de nombres de destino, mientras que las referencias a los elementos WSDL se hacen por medio del prefijo `tns`. Por ejemplo, `<wsdl:binding name="EnlaceNuevo" type="tns:PortTypeNuevo">`. Para que el prefijo `tns` coincida con el espacio de nombres de destino, su valor de espacio de nombres debe ser idéntico al espacio de nombres de destino (`targetNamespace`).

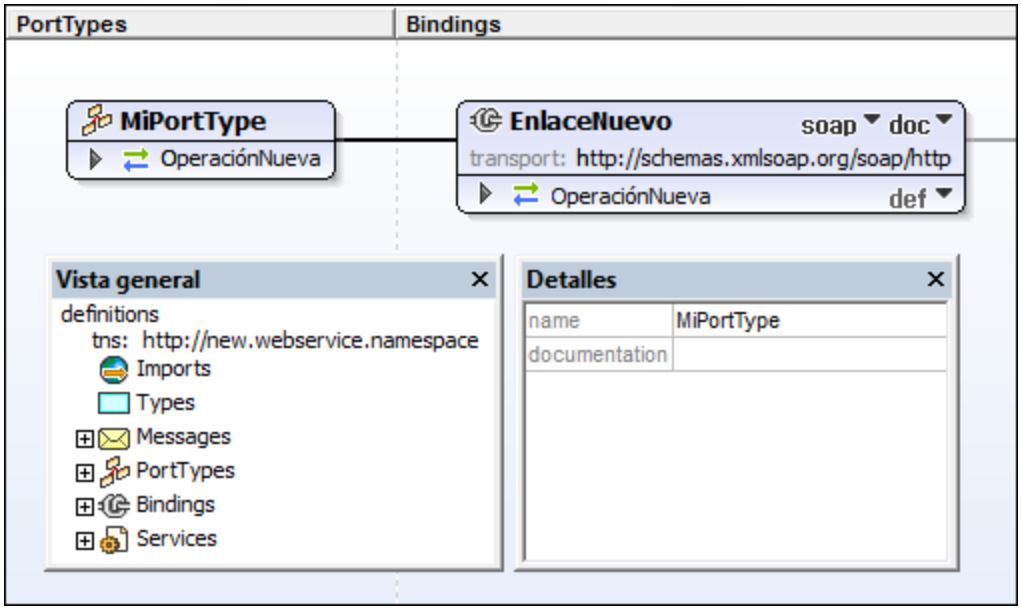
15.1.2 Crear un portType

La creación de un portType implica todos estos pasos:

- Nombrar el portType
- Insertar una operación
- Agregar mensajes de entrada y salida
- Agregar parámetros a los mensajes

Nombrar el portType

Cambie el nombre del portType `PortTypeNuevo` por `MiPortType`. Para ello haga doble clic en la barra de título del recuadro `PortTypeNuevo` del diseño, cambie el nombre y pulse la tecla **Entrar**. Observe que el nombre del portType también cambia en los ayudantes de entrada Vista general y Detalles (*ver imagen siguiente*).



Insertar una operación

En el caso del portType `MiPortType` ya existe una operación (`OperaciónNueva`), de modo que trabajaremos con esta operación. Para empezar cambie el nombre de la operación `OperaciónNueva` por, por ejemplo, `EchoString` (haga doble clic en el nombre de la operación, edítelo y pulse **Entrar**). Para insertar operaciones nuevas en un portType, haga clic con el botón derecho en el recuadro del portType, seleccione el comando **Anexar operación** y haga clic en el tipo de operación que desea insertar.

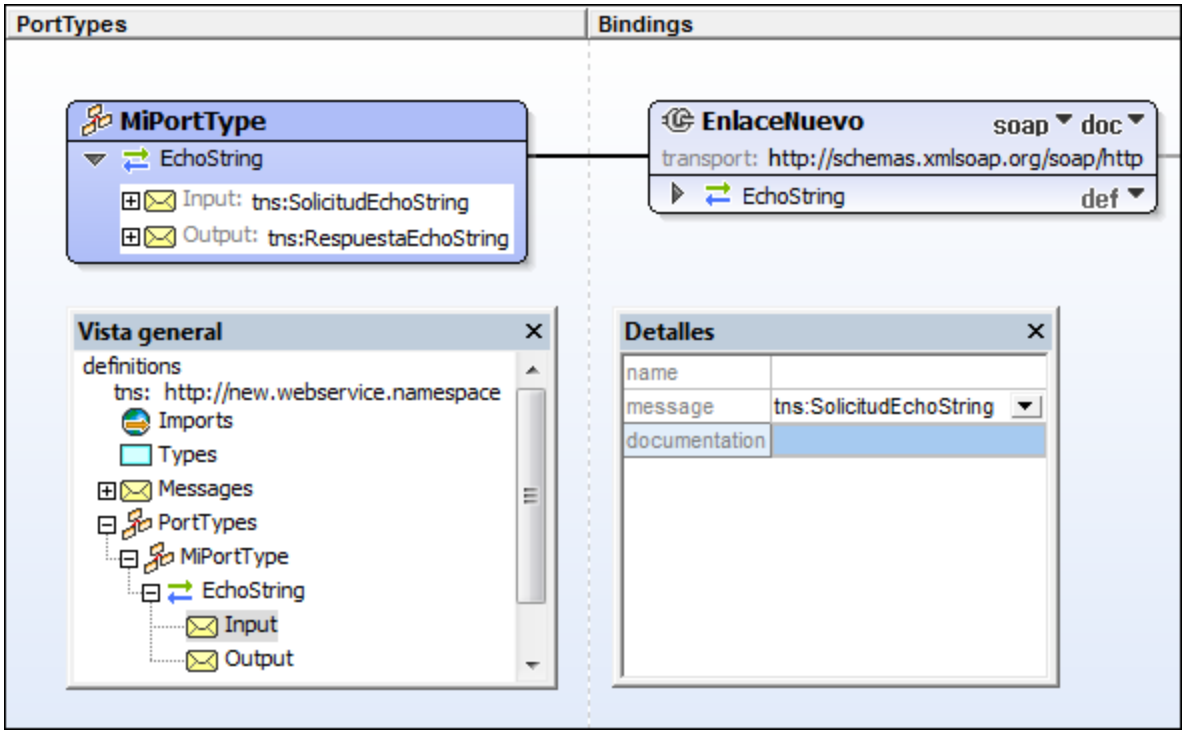
Agregar mensajes de entrada y salida

Cuando se anexa una operación a un portType, puede elegir qué tipo de operación se anexa. Hay cinco tipos de operaciones:

- Solicitud-respuesta
- Petición-respuesta
- Unidireccional
- Notificación
- Operación vacía

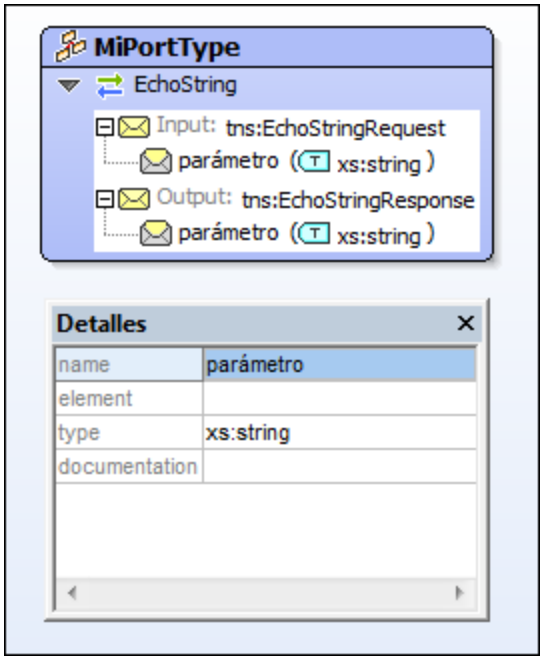
A cada tipo de operación se añaden automáticamente mensajes de entrada y salida. Cuando se selecciona **Operación vacía** y se hace clic con el botón derecho en la operación, puede seleccionar el tipo de mensaje que se inserta. Para eliminar un mensaje, haga clic con el botón derecho en el mensaje y seleccione **Eliminar elemento de entrada, salida o error**. En el ejemplo de la operación que nos ocupa, `EchoString`, cambie el nombre de los mensajes de entrada y salida por `SolicitudEchoString` y `RespuestaEchoString` respectivamente.

© 2016-2022 Altova GmbH



Agregar parámetros a mensajes

Cada mensaje de entrada o salida se crea con una parte de mensaje (o parámetro) predeterminada de tipo `xs:string` (*ver imagen siguiente*). Para agregar otro parámetro, haga clic con el botón derecho en el mensaje o en uno de sus parámetros y seleccione **Agregar parte de mensaje (parámetro)**.



Para editar un parámetro tiene tres opciones: (i) puede hacer doble clic en el texto y editarlo, (ii) hacer clic con el botón derecho en el parámetro y seleccionar **Editar** o (iii) usar el ayudante de entrada Detalles (*ver imagen anterior*).

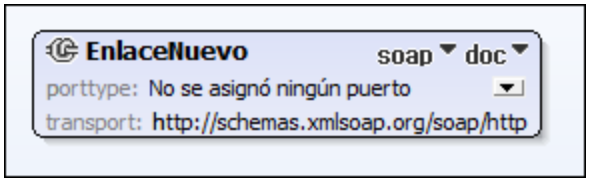
15.1.3 Crear un enlace

Un enlace (*binding*) es un protocolo y una especificación de formato de datos para un portType. Por tanto, para crear un enlace es necesario:

- Asociar el enlace a un portType.
- Definir el protocolo y la especificación de formato de datos del enlace.
- Asociar el enlace a un puerto.

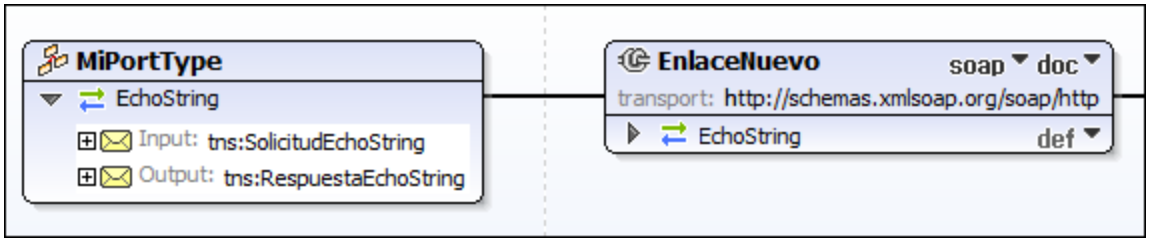
Asociar un enlace a un portType

Para crear un enlace nuevo, haga clic con el botón derecho en cualquier parte del diseño que esté vacía y seleccione **Insertar enlace**. Cuando se crea un enlace nuevo, el enlace no tiene asociado ningún portType (*ver imagen siguiente*). (Cuando se crea un documento WSDL nuevo, el enlace predeterminado que se crea está asociado al portType predeterminado y la asociación se ilustra por medio de una línea que conecta los dos recuadros.)



 © 2016-2022 Altova GmbH

Para asociar un portType a un enlace, haga clic en la punta de flecha que aparece en el recuadro del enlace, junto a la entrada `porttype` (*ver imagen anterior*). Al hacer clic en la punta de flecha aparece una lista de los portType definidos en el documento. Seleccione el portType al que desea asociar el enlace. Cuando un portType está asociado a un enlace, la asociación se ilustra por medio de una línea que conecta el recuadro del portType seleccionado con el del enlace (*ver imagen siguiente*).



Seleccionar el protocolo y el formato de datos

Para seleccionar el protocolo del enlace, haga clic en la primera punta de flecha de la barra de título del recuadro del enlace (junto a la entrada `soap/http`) y seleccione uno de los cuatro protocolos posibles: `SOAP`, `SOAP 1.2`, `HTTP-GET` y `HTTP-POST` (*ver imagen siguiente*). Si elige el protocolo SOAP 1.1 o 1.2, puede seleccionar `document` o `rpc` como formato de datos (en la lista que aparece al hacer clic en la segunda punta de flecha de la barra de título, a la derecha de la entrada `soap`).



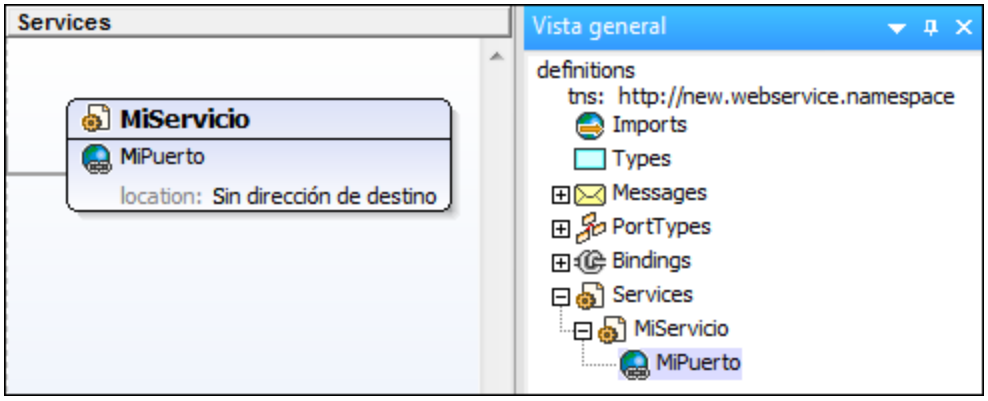
El atributo `soapAction` de cada operación del enlace se puede definir en el diseño (*ver imagen anterior*) o en el ayudante de entrada Detalles cuando la operación está seleccionada.

Asociar el enlace a un puerto

Para asociar el enlace a un puerto, antes debe definir el puerto. Para aprender a crear un puerto dentro de un servicio y asociar un puerto a un enlace, consulte el apartado Crear un servicio y puertos 781.

15.1.4 Crear un servicio y puertos

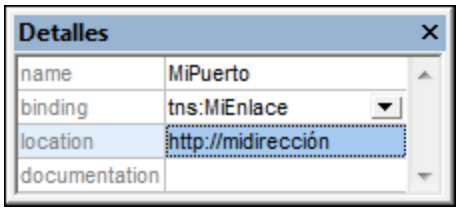
Para agregar un servicio nuevo, haga clic con el botón derecho en cualquier parte del diseño que esté vacía y seleccione el comando **Insertar servicio** del menú contextual. Si creó un documento WSDL nuevo, el diseño ya debe incluir un servicio. Para cambiar el nombre de servicio, haga doble clic en el nombre del servicio, edítelo y pulse **Entrar**. Observe que el nombre del servicio también cambia en el ayudante de entrada Vista general (*ver imagen siguiente*).



En el ayudante de entrada Vista General, haga doble clic en la entrada `PuertoNuevo`, cambie el nombre por `MiPuerto` y pulse **Entrar**. Observe que el nombre del puerto también cambia en el recuadro `MiServicio` del diseño (*ver imagen anterior*). Para agregar más puertos, haga clic con el botón derecho en el servicio o puerto y seleccione **Insertar puerto** en el menú contextual.

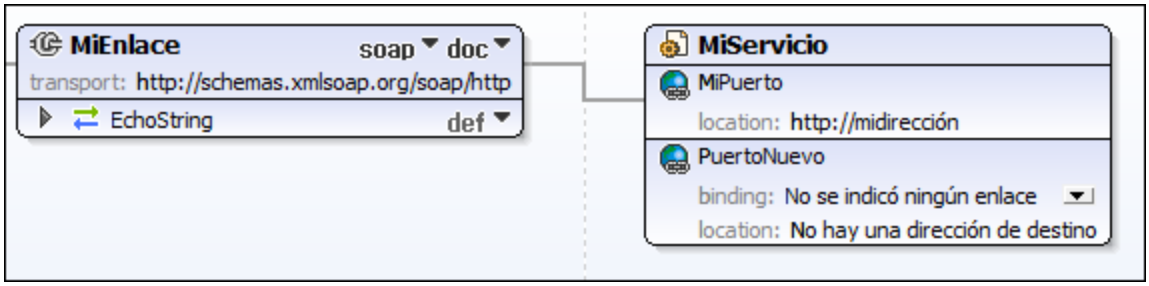
Introducir la dirección de un puerto

La dirección de un puerto se puede introducir de dos maneras distintas: (i) directamente en el diseño, como valor del atributo `Location` (*ver imagen anterior*) o (ii) en el ayudante de entrada Detalles (haciendo doble clic en el campo `Location` y escribiendo la dirección, como muestra la imagen siguiente).



Asociar un enlace a un puerto

Un puerto es el extremo que combina un enlace con una dirección de red. Una vez definida la dirección de un puerto, lo siguiente es asociar un enlace al puerto. Para asociar un enlace a un puerto, haga clic en la punta de flecha situada junto al atributo `binding` del recuadro del puerto y seleccione una opción de la lista de enlaces definidos en el documento.

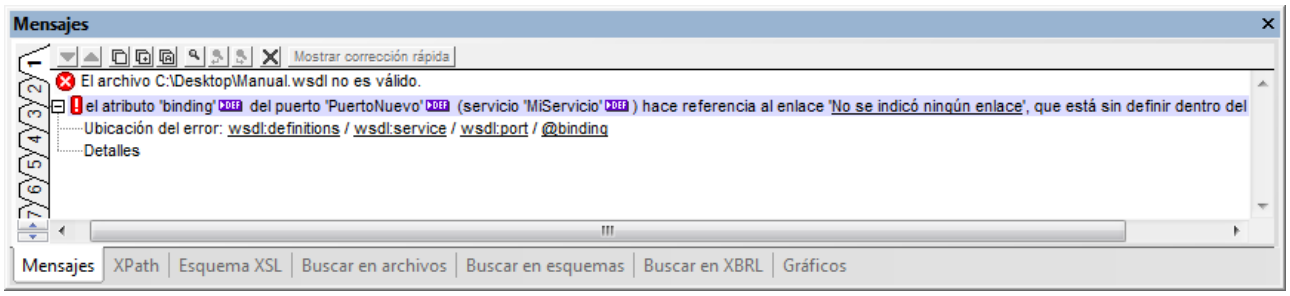


© 2016-2022 Altova GmbH

Nota: si un enlace ya está asociado a un puerto y desea asociarlo a otro puerto, antes debe eliminar la referencia de enlace (haciendo clic con el botón derecho en el puerto) y después insertar la referencia de enlace nueva.

15.1.5 Validar el documento WSDL

Cuando termine de editar su documento WSDL puede validarlo con el comando **XML | Validar el documento XML (F8)**. Los resultados de la validación aparecen en la ventana Mensajes (*ver imagen siguiente*).



Los mensajes de error incluyen información detallada sobre el error detectado para que pueda localizarlo y corregirlo rápidamente.

15.1.6 Conectarse a un servicio web y abrir archivos

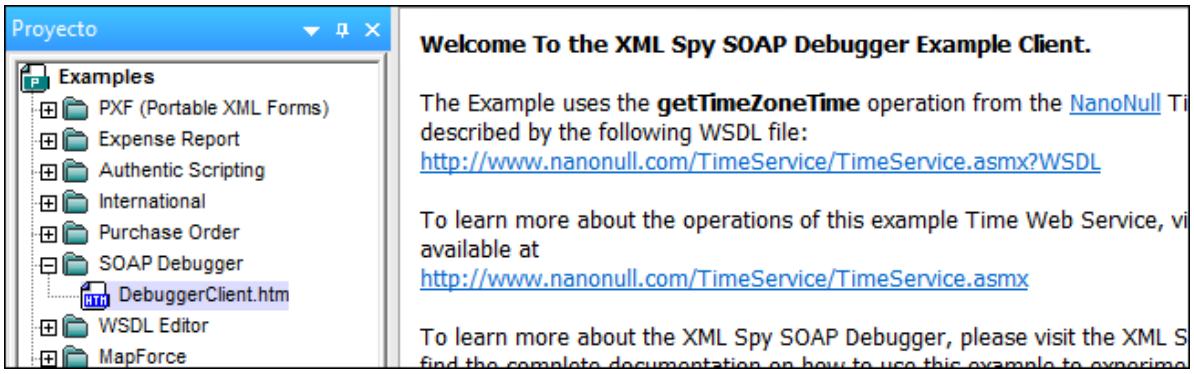
En esta sección aprenderá:

- A acceder a un servicio web con XMLSpy
- A abrir un archivo web WSDL con XMLSpy
- A abrir con XMLSpy un esquema XML asociado al documento WSDL

Acceder a un servicio web

A los servicios web se suele acceder desde una página HTML, como la página **`DebuggerClient.htm`** de la carpeta *Examples*, así como desde el proyecto de XMLSpy 1060 **`Examples/Soap Debugger`** (abra **`Examples.spp`** para trabajar con el proyecto). Para acceder al servicio web que aparece en esta página siga estos pasos:

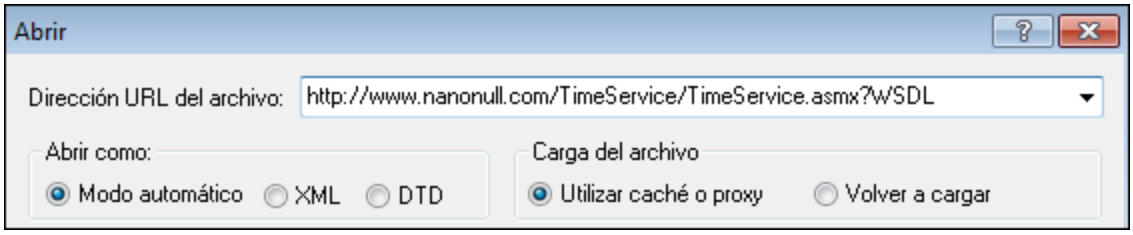
1. Si no está visible ya, active la ventana Proyecto (con la opción de menú **Ventanas | Ventana de proyecto**).
2. Haga clic en el icono + situado junto a la carpeta SOAP Debugger y después haga doble clic en el archivo **`DebuggerClient.htm`**. El archivo se abre en la ventana principal.



Abrir un archivo WSDL con XMLSpy

Para abrir un archivo WSDL en la web con XMLSpy siga estos pasos:

1. Seleccione la opción de menú Archivo | Abrir y, en el cuadro de diálogo Abrir, haga clic en el botón Cambiar a URL. Escriba o copie la dirección `http://www.nanonull.com/TimeService/TimeService.asmx?WSDL` en el campo Dirección URL del archivo del cuadro de diálogo.



2. Haga clic en **Abrir** para cargar el archivo WSDL. El archivo WSDL se carga en la vista Texto.
3. Seleccione la opción de menú **Archivo | Guardar como...** y guarde el archivo como `CompanyLast.xml`.
4. Haga clic en la pestaña WSDL de la ventana principal para ver el archivo en el editor WSDL gráfico.

Ver el archivo de esquema asociado al archivo WSDL activo

Con el archivo `timeservice.wsdl` abierto en la vista WSDL, seleccione la opción de menú **WSDL | Tipos | Editar esquemas en la vista Esquema**. Como resultado se abre el archivo de esquema que define todos los tipos de datos utilizados en el archivo `timeservice.wsdl`. A continuación puede cambiar el esquema y guardar los cambios realizados. Estos cambios surtirán efecto cuando se vuelva a analizar el archivo WSDL.

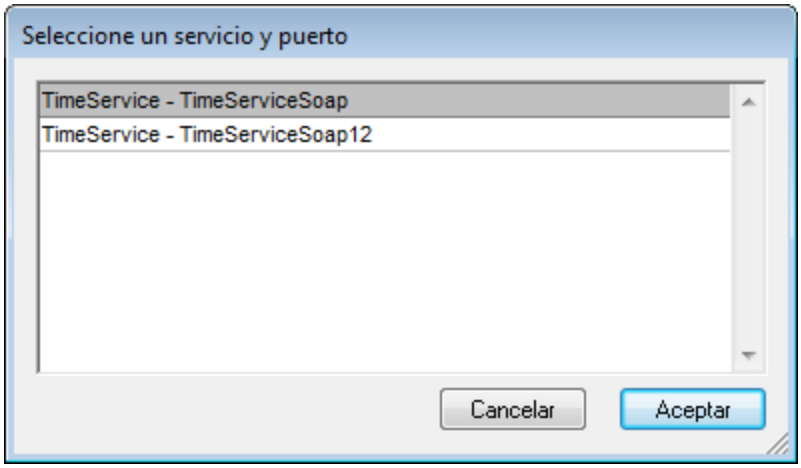
Nota: recomendamos que acceda a la vista WSDL usando el nombre de archivo (por ejemplo, `timeservice.wsdl`) en lugar de hacerlo con el método de consulta `?wsdl`. El motivo es que el método de consulta puede devolver un archivo WSDL que no sea idéntico al archivo WSDL original o uno que no funcione correctamente.

© 2016-2022 Altova GmbH

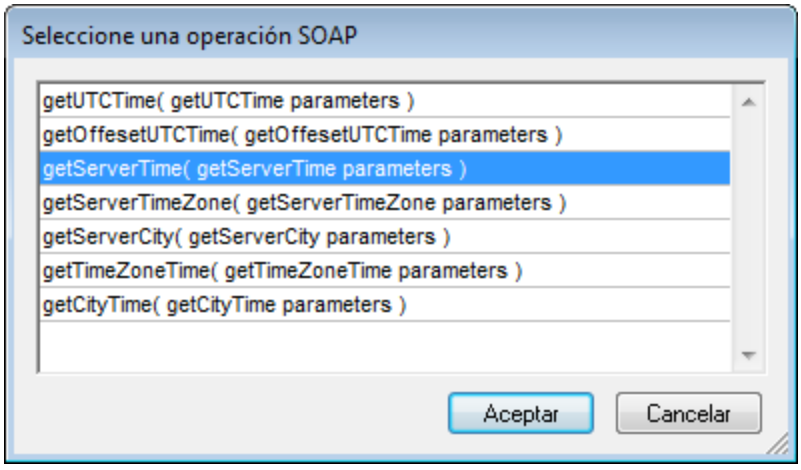
15.1.7 Enviar una solicitud SOAP desde el archivo WSDL

Para enviar una solicitud SOAP desde el archivo `timeservice.wsdl` siga estos pasos:

1. Abra el archivo `timeservice.wsdl` en la vista WSDL.
2. Seleccione la opción de menú **SOAP | Crear solicitud SOAP** nueva.
3. Busque el archivo `timeservice.wsdl` y haga clic en **Aceptar**.
4. Si entre los servicios definidos en el documento existen varios puertos que hacen referencia a un enlace SOAP 1.1 o 1.2, entonces aparece un mensaje (*imagen siguiente*) pidiendo que seleccione el servicio y puerto que desea utilizar. Elija una opción y haga clic en **Aceptar**.



5. A continuación aparece otro cuadro de diálogo (*imagen siguiente*) donde debe seleccionar una operación SOAP, por ejemplo, `getServerTime`. Haga clic en **Aceptar**.



   Como resultado se crea un documento de solicitud SOAP que contiene la operación `getServerTime`. Si quiere puede guardar este documento.
6. Con el documento de solicitud como documento activo seleccione la opción de menú **SOAP | Enviar solicitud al servidor**. Como resultado aparece en la ventana principal un documento de respuesta SOAP. Este documento de respuesta incluye el elemento `getServerTimeResult`, que muestra la hora actual del servidor del servicio de hora Nanonull.com.

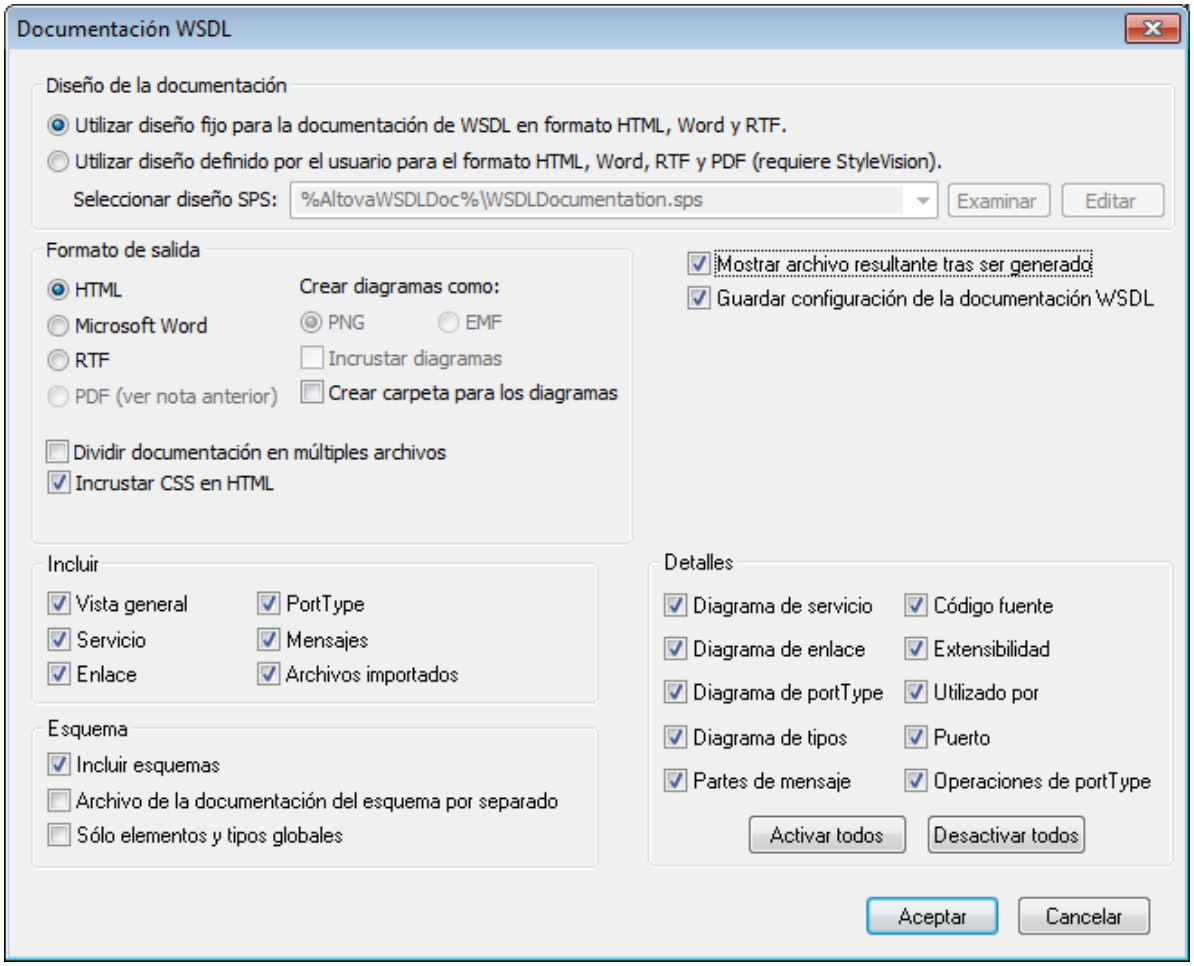
15.1.8 Crear documentación WSDL

La opción de menú **WSDL | Generar documentación** sirve para producir documentación detallada del documento WSDL. Puede generar la documentación en formato HTML, MS Word o RTF e indicar qué componentes desea incluir en la documentación. Los elementos WSDL relacionados están conectados mediante hipervínculos en la documentación, para que pueda navegar por ella fácilmente.

Nota: Para generar documentación en formato MS Word, es necesario tener instalado MS Word (versión 2000 o posterior).

Siga estos pasos para generar documentación para el archivo WSDL:

1. Abra el archivo `timeservice.wsdl` en la vista **WSDL**.
2. Seleccione la opción de menú **WSDL | Generar documentación**. Aparece el cuadro de diálogo Documentación WSDL (*imagen siguiente*).
3. Seleccione el formato de salida: HTML, MS Word o RTF.
4. Seleccione los componentes WSDL que desea incluir en la documentación y configure el resto de opciones (vea el siguiente apartado *Opciones de la documentación WSDL*).



6. Haga clic en **Aceptar** y escriba el nombre de archivo de la documentación WSDL en el cuadro de diálogo Guardar como.

© 2016-2022 Altova GmbH

Opciones de la documentación WSDL

Estas son las opciones de la documentación WSDL:

- El formato de salida se indica en el grupo de opciones *Formato de salida:* HTML, Microsoft Word o RTF. La documentación se puede generar en un solo archivo o en varios archivos. Si elige generar la documentación en varios archivos, se genera un archivo por cada componente. Para indicar qué componentes desea incluir en la documentación, utilice las casillas del grupo de opciones *Incluir*.
- La opción *Incrustar diagramas* está habilitada si se seleccionan los formatos de salida MS Word o RTF. Si marca esta casilla, los diagramas se incrustan en el archivo de documentación, en formato PNG o EMF. Si no marca esta casilla, los diagramas se guardan como archivos PNG o EMF y el archivo de documentación incluye vínculos de objeto a estos archivos. Si el formato de salida elegido es HTML, los diagramas se crean como archivos PNG externos.
- En el grupo de opciones *Incluir* puede seleccionar los componentes que desea incluir en la documentación. Si marca la opción *Vista general,* se enumeran todos los componentes, ordenados por tipo, al principio del archivo de documentación. Si marca la casilla *Archivos importados*, se incluyen también los componentes de los archivos importados.
- En el grupo de opciones *Esquema* puede seleccionar si se incluyen o no los esquemas que contiene el documento WSDL. Si decide incluir los esquemas en la documentación, también puede especificar: (i) si la documentación del esquema se genera en un archivo diferente o en el archivo de documentación principal y (ii) si se incluye todo el esquema o sólo los elementos, tipos complejos y tipos simples globales.
- El grupo de opciones *Detalles* enumera los detalles que se pueden incluir en la documentación de cada componente. Seleccione los detalles que desea incluir en la documentación.
- La opción *Mostrar archivo resultante tras ser generado* está habilitada para las tres opciones de formato de salida. Si marca esta casilla, los archivos resultantes se abren en la vista Explorador (formato de salida HTML), en MS Word (formato de salida Word) y en la aplicación predeterminada para archivos `.rtf` (formato de salida RTF).

15.1.9 Conversión en WSDL 2.0

En XMLSpy es muy fácil pasar documentos WSDL 1.1 a formato WSDL 2.0. Para ello siga estos pasos:

1. Abra el archivo `TimeService.wsdl` en la vista WSDL. (Este archivo está en la carpeta `WSDL Editor` del archivo de proyecto `Examples` de XMLSpy.)
2. Haga clic en el comando **WSDL | Convertir en WSDL 2.0**.
3. Aparece el cuadro de diálogo "Guardar como". Escriba el nombre que desea dar al archivo WSDL 2.0, por ejemplo `TimeService20.wsdl`, y haga clic en **Aceptar**.
4. A continuación se genera un archivo nuevo, se valida automáticamente y se abre en la vista WSDL.

15.2 SOAP

En esta sección aprenderá:

- A validar mensajes SOAP con archivos WSDL. Los mensajes SOAP se pueden revisar para ver si son válidos con respecto a la especificación SOAP y con respecto a los esquemas XML a los que se hagan referencia en la definición WSDL correspondiente
- A enviar y recibir solicitudes SOAP con el depurador SOAP
- A establecer puntos de interrupción para enviar y recibir solicitudes SOAP
- A editar solicitudes SOAP incorrectas antes de enviarlas al servicio web

Cuando se desarrolla una aplicación que accede a datos desde un servicio web y después procesa los datos, pueden pasar cosas inesperadas. Este apartado explica cómo ver y editar los datos que se están transmitiendo. Esto se hace con el depurador SOAP, que funciona como un **servidor proxy** entre su cliente y el servicio web. El depurador es mucho más que una simple herramienta de seguimiento que almacena solicitudes y respuestas de cada llamada.

Con el depurador SOAP puede establecer puntos de interrupción por cada mensaje de solicitud y respuesta e incluso definir puntos de interrupción condicionales mediante expresiones XPath.

El depurador SOAP puede:

- recorrer paso a paso las solicitudes y respuestas SOAP
- modificar solicitudes y respuestas SOAP
- reenviar al cliente o servidor solicitudes SOAP modificadas

Sitio web de Altova: Depurador SOAP

15.2.1 Validación SOAP

Puede revisar los mensajes SOAP para ver si son válidos con respecto a la especificación SOAP y con respecto a los esquemas a los que se haga referencia en la definición WSDL correspondiente.

© 2016-2022 Altova GmbH

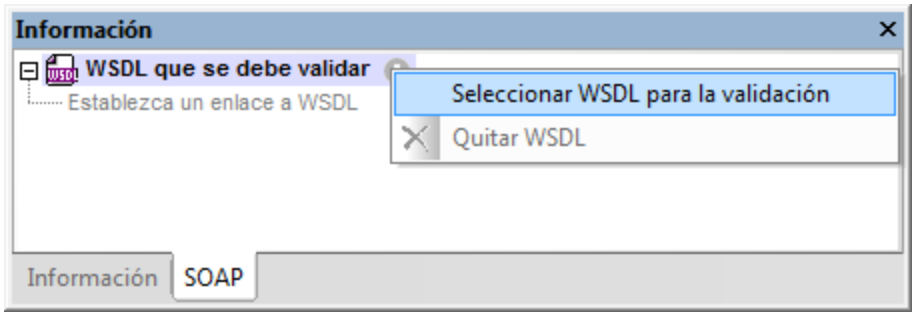
Validación con respecto a las reglas SOAP

Para validar un mensaje SOAP, abra el archivo de mensaje SOAP (*imagen siguiente*) y pulse **F8** (o el comando de menú **XML | Validar**). Puesto que no se asoció ningún archivo WSDL con el archivo de mensaje SOAP, el mensaje SOAP se valida con respecto a las reglas para mensajes SOAP. El archivo es válido si es válido con respecto a estas reglas (*ver imagen siguiente*).



Validación con respecto a las reglas SOAP y al archivo WSDL asociado

Para validar un mensaje SOAP con respecto al archivo WSDL asociado, antes debe asociar un archivo WSDL al archivo SOAP. Esto se hace en la pestaña SOAP de la ventana Información (*ver imagen siguiente*). Haga clic en el icono situado a la derecha de la entrada *WSDL que se debe validar* y seleccione el comando **Seleccionar WSDL para la validación**. Aparece un cuadro de diálogo. Busque el archivo WSDL y haga clic en **Aceptar**. El archivo WSDL aparece en la ventana Información y el archivo de mensaje SOAP se asocia al WSDL.



Al pulsar **F8** (o el comando de menú **XML | Validar**) se valida el mensaje SOAP, no sólo con respecto a las reglas SOAP, sino también con respecto a las reglas del archivo WSDL asociado.



El archivo es válido si es válido con respecto a ambos conjuntos de reglas (*ver imagen anterior*).

Nota: La pestaña SOAP de la ventana Información aparece sólo si la solicitud SOAP se creó con la función de creación de solicitud SOAP de XMLSpy a partir de un archivo WSDL (**SOAP | Crear solicitud SOAP nueva**). Si la pestaña SOAP no aparece en la ventana Información (porque la solicitud SOAP no se creó con XMLSpy), guarde el archivo de solicitud SOAP y aparecerá la pestaña.

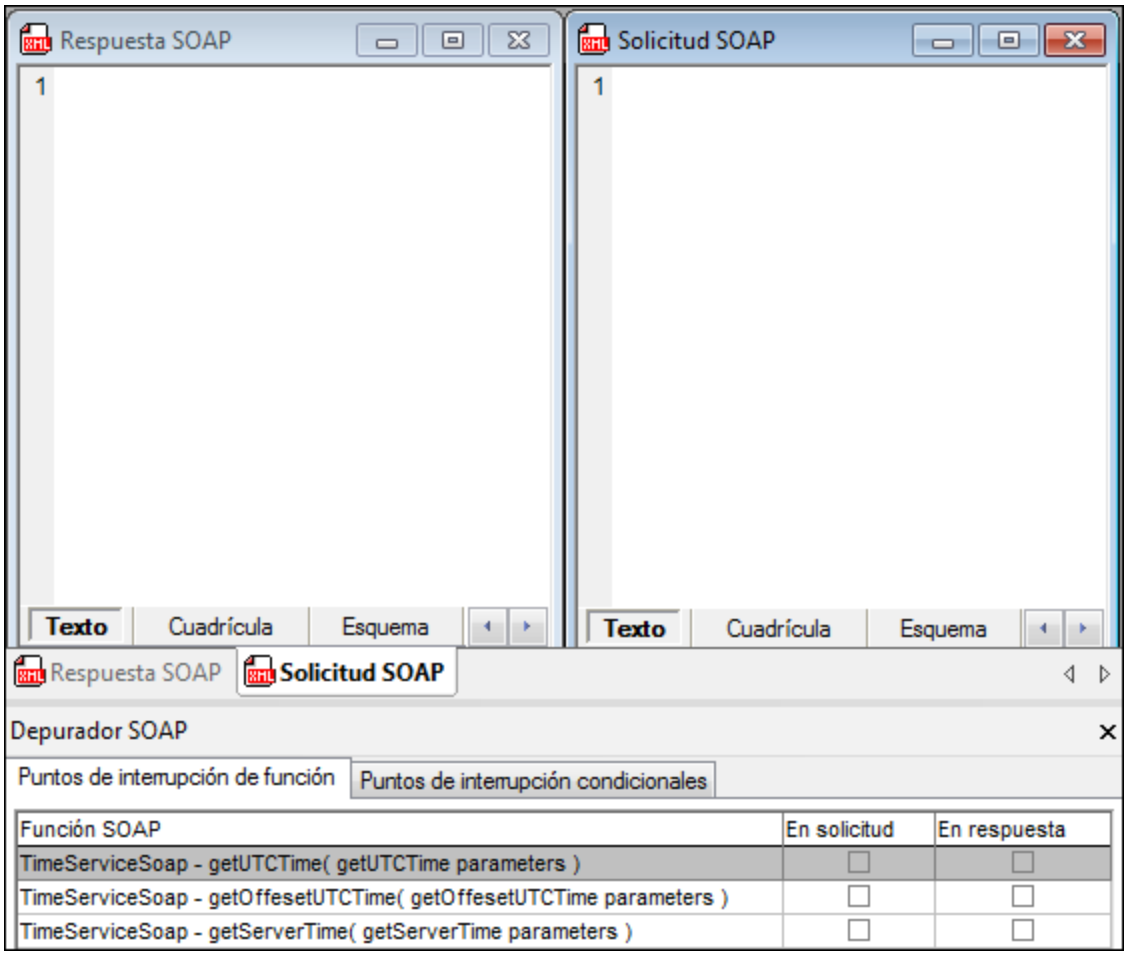
15.2.2 Depurador SOAP

El depurador SOAP (*imagen siguiente*) sirve para ver y analizar solicitudes y respuestas SOAP. Funciona como **servidor proxy** entre el cliente y el servicio web.

En el depurador SOAP puede:

- recorrer las solicitudes y respuestas SOAP paso a paso
- modificar las solicitudes y respuestas SOAP
- reenviar las solicitudes modificadas al cliente o servidor
- establecer puntos de interrupción en cada solicitud y mensaje de respuesta, incluidos puntos de interrupción condicionales con expresiones XPath

© 2016-2022 Altova GmbH



Así funciona el depurador SOAP:

- Antes de iniciar una sesión del depurador SOAP debería configurar las opciones del depurador SOAP[792]. Estas opciones incluyen la dirección IP del equipo, el tiempo de espera, etc.
- Para abrir el depurador SOAP (iniciar una sesión)[793], seleccione el comando **SOAP | Sesión de depurador SOAP**. Será necesario especificar (i) la ubicación del archivo WSDL que contiene la información SOAP y (i) los datos de los puertos de origen y destino.
- En la ventana Puntos de interrupción del depurador SOAP defina los puntos de interrupción necesarios[798].
- Ahora puede abrir el archivo que realiza la solicitud SOAP y ejecutar el depurador SOAP[798].
- Después puede analizar los resultados[800] y, si encuentra errores, corregirlos.
- Para cerrar el depurador SOAP seleccione el comando **SOAP | Sesión de depurador SOAP**.

En los subapartados siguientes explicamos cómo usar el depurador SOAP.

Como ejemplo utilizamos el archivo `DebuggerClient.htm`, guardado en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples`. Para este archivo de ejemplo la ventana del explorador hace las funciones de aplicación cliente que envía y recibe mensajes SOAP. El servicio Nanonull Time Service es el servidor del servicio web y está ubicado en `http://www.nanonull.com/TimeService/TimeService.asmx?WSDL`.

15.2.2.1 Proceso de la comunicación SOAP

Una vez iniciado el servidor proxy (el depurador SOAP), el proceso de la comunicación SOAP es este:

El servidor proxy escucha constantemente a un puerto o socket para detectar solicitudes de cliente entrantes

- La aplicación cliente envía una solicitud al servidor proxy
- Las solicitudes de cliente se pueden modificar si/cuando se desencadenan puntos de interrupción
- Los datos de solicitud del servidor proxy se reenvían al servidor del servicio web

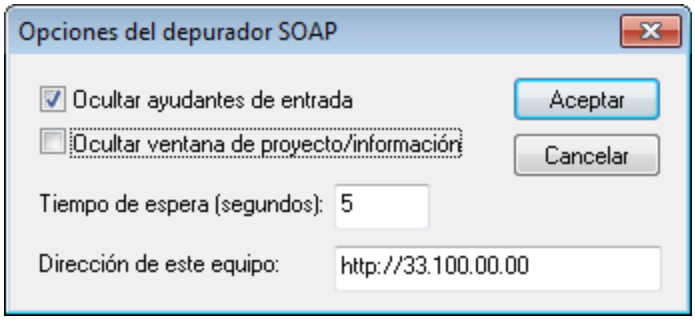
El servidor del servicio web responde a la solicitud del proxy y devuelve los datos de respuesta al servidor proxy

- Las respuestas del servidor se pueden modificar si/cuando se desencadenan puntos de interrupción
- Los datos de respuesta del servidor proxy se reenvían a la aplicación cliente
- La aplicación cliente recibe los datos de respuesta del servidor proxy

Configuración de los puertos
El depurador SOAP usa el puerto 8080 para supervisar las solicitudes de clientes. El puerto sólo se puede alterar cuando se inicia una sesión de depuración SOAP nueva. Si este puerto estuviera deshabilitado por servidores de seguridad o filtros de publicidad, deberá deshabilitar estos programas o seleccionar una dirección de puerto diferente.

15.2.2.2 Opciones del depurador SOAP

El depurador SOAP se puede configurar en el cuadro de diálogo "Opciones del depurador SOAP" (*imagen siguiente*). En este cuadro de diálogo puede especificar la dirección IP del equipo y otras opciones de depuración (descritas más abajo). Para abrir este cuadro de diálogo haga clic en el comando de menú **SOAP | Opciones del depurador SOAP**.



- *Dirección de este equipo:* se trata de la dirección del servidor proxy desde el que se ejecuta el depurador. El depurador del servidor proxy recibe solicitudes de equipos de la red y los envía al servicio web. Como el depurador se ejecuta dentro de XMLSpy, el equipo donde esté instalado XMLSpy también hace de servidor proxy. La dirección IP del equipo se detecta automáticamente y aparece en este campo del cuadro de diálogo. Es decir, solamente deberá introducir la dirección en este campo si XMLSpy no puede detectarla automáticamente. Para averiguar la dirección IP de su equipo abra una ventana de DOS, escriba el comando `ipconfig /all` y pulse la tecla **Entrar**.

© 2016-2022 Altova GmbH

- *Tiempo de espera:* este valor es el tiempo que permanece el depurador en un punto de interrupción. El valor predeterminado es 5 segundos.
- *Ocultar ayudantes de entrada y ventana de proyecto/información:* utilice estas opciones para obtener más espacio en pantalla para la ventana del depurador SOAP.

15.2.2.3 Iniciar una sesión de depuración

Puede iniciar una sesión de depurador SOAP siempre que quiera, independientemente del tipo de archivo activo. Es decir, para iniciar el depurador SOAP no es necesario que el archivo activo sea un archivo SOAP ni un archivo de punto de entrada para la solicitud SOAP[796]. Al iniciar la sesión de depuración SOAP la aplicación solicita:

1. la ubicación del archivo WSDL que contiene la información SOAP y
2. las opciones de configuración de la conexión.

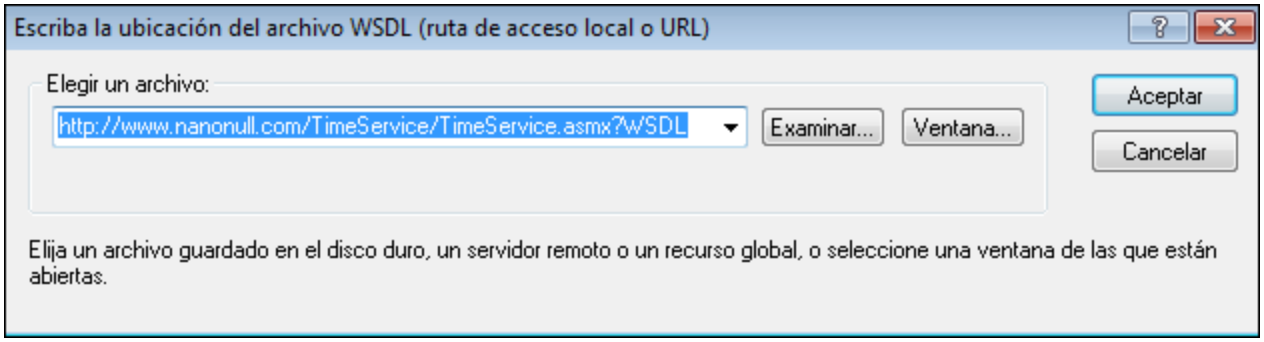
A continuación describimos el proceso entero.

Ubicación del archivo WSDL

Cuando se inicia la sesión de depuración, la aplicación solicita la URL del archivo WSDL que contiene la información SOAP. Nuestro archivo de ejemplo[796], `DebuggerClient.html`, utiliza esta URL de archivo:

```
http://www.nanonull.com/TimeService/TimeService.asmx?WSDL
```

Inicie el depurador SOAP seleccionando el comando **SOAP | Sesión de depurador SOAP**. Aparece este cuadro de diálogo:



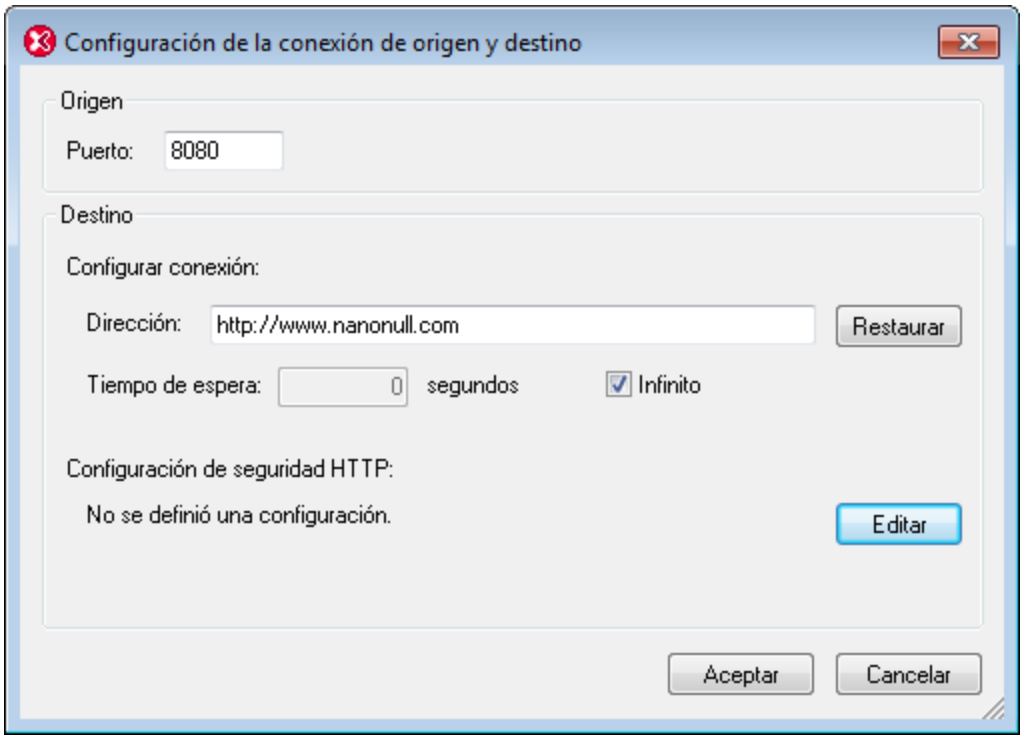
Escriba la URL del archivo WSDL y haga clic en **Aceptar**. Aparece el cuadro de diálogo de configuración de la conexión.

Configuración de la conexión

El cuadro de diálogo "Configuración de la conexión de origen y destino" (*imagen siguiente*) ofrece varias opciones:

- *Puerto de origen:* el puerto del servidor proxy (que puede ser su equipo) que se utilizará para la comunicación. El puerto predeterminado es `8080`. El puerto de origen se puede cambiar cada vez que se inicie el depurador SOAP.

- *Puerto y dirección de destino:* estas opciones vienen dadas por el archivo WSDL seleccionado en el paso anterior y se introducen automáticamente. El puerto predeterminado es el número 80. Puede definir un tiempo de espera para la conexión o marcar la casilla Infinito si prefiere no definir un tiempo de espera. Para definir la configuración de seguridad HTTP haga clic en el botón **Editar** y elija la configuración correspondiente. Para más información consulte el apartado Configuración de la solicitud SOAP[1507].



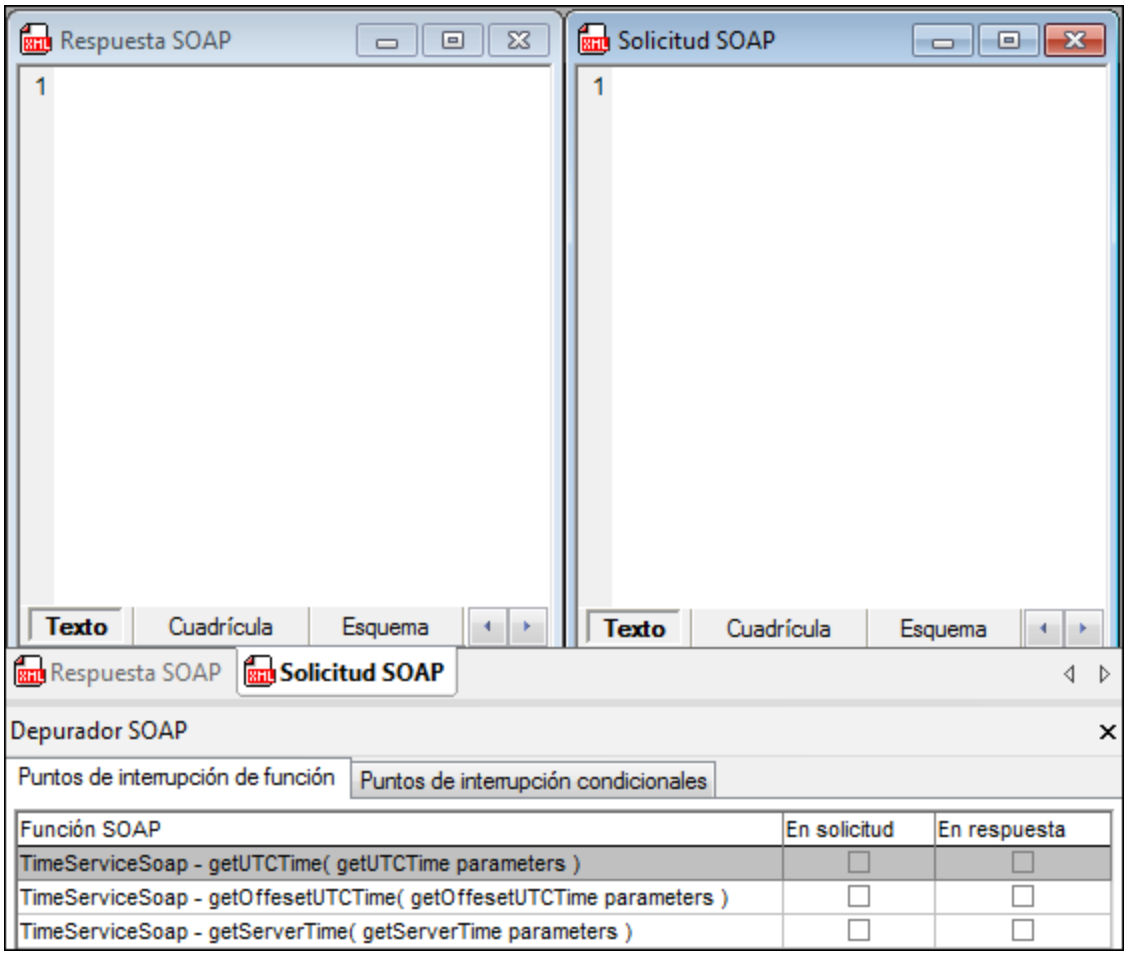
Cuando termine de configurar los puertos haga clic en **Aceptar** para iniciar la sesión de depurador SOAP. El depurador SOAP se inicia pero el servidor proxy está inactivo (observe que el icono del servidor proxy de la barra de herramientas está atenuado). Para iniciar el servidor proxy (es decir, la depuración), haga clic en el icono **Ejecutar** de la barra de herramientas o haga clic en el comando **SOAP | Ejecutar**. Para más información consulte el subapartado Depuración[798].

Disposición del depurador SOAP

El depurador SOAP tiene tres ventanas (*imagen siguiente*):

- la ventana Solicitud SOAP
- la ventana Respuesta SOAP y
- la ventana de puntos de interrupción. Las dos primeras se abren por defecto en la parte superior de la pantalla de XMLSpy y la ventana de puntos de interrupción se abre debajo de ellas. La imagen siguiente muestra la disposición predeterminada de las ventanas del depurador SOAP.

© 2016-2022 Altova GmbH



Si necesita más espacio para las ventanas del depurador SOAP, oculte las ventanas laterales de XMLSpy (es decir, las ventanas de los ayudantes de entrada, la ventana Proyecto y la ventana Información). Esto puede configurarse en el cuadro de diálogo "Opciones del depurador SOAP" 792 (**SOAP | Opciones de depurador SOAP**).

Certificados de confianza

Los productos de Altova utilizan Internet Explorer (IE) para acceder y gestionar certificados de confianza de servidores web seguros. Instalar el certificado de un servidor web en IE permite a IE acceder al servidor web sin emitir una advertencia ni cancelar el proceso. A continuación aparecen los pasos básicos del proceso de instalación de un certificado de un servidor web seguro. Dependiendo de la versión del explorador, el proceso de instalación puede ser más complicado.

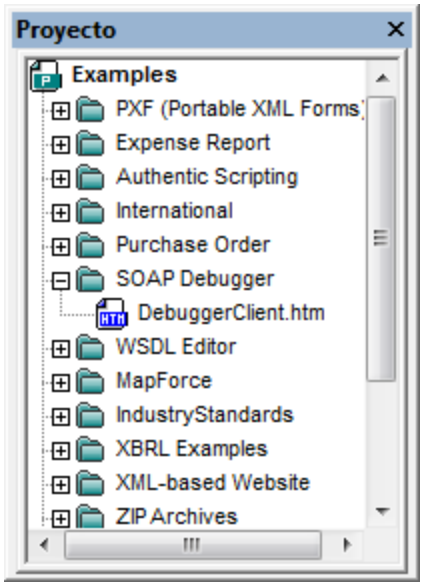
1. Abra el sitio web seguro en Internet Explorer 8 (o superior)
2. Seleccione **Archivo | Propiedades** y haga clic en el botón **Certificados**.
3. Haga clic en **Instalar certificado** e inicie el asistente de importación de certificados. (Este asistente también se puede abrir desde **Herramientas | Opciones de Internet | Contenido | Certificados | Importar**.)
4. El certificado debería estar en el almacén "Entidades de certificación raíz de confianza", que se puede buscar manualmente.
5. Siga los pasos restantes del asistente y cierre los cuadros de diálogo Certificados y Propiedades haciendo clic en **Aceptar** en cada uno de ellos. Quizás sea necesario reiniciar Internet Explorer.

15.2.2.4 Punto de entrada de la solicitud SOAP

El archivo HTML `DebuggerClient.htm` del proyecto Examples contiene un script que ilustra el uso del depurador SOAP. Este archivo permite enviar una solicitud SOAP a un servicio web y después muestra la respuesta del servicio web.

Para abrir este archivo en XMLSpy:

1. Seleccione el comando de menú **Proyecto | Abrir proyecto**.
2. Examine la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples` y seleccione el archivo `Examples.spp`. El proyecto Examples se carga en la ventana Proyecto (*imagen siguiente*).



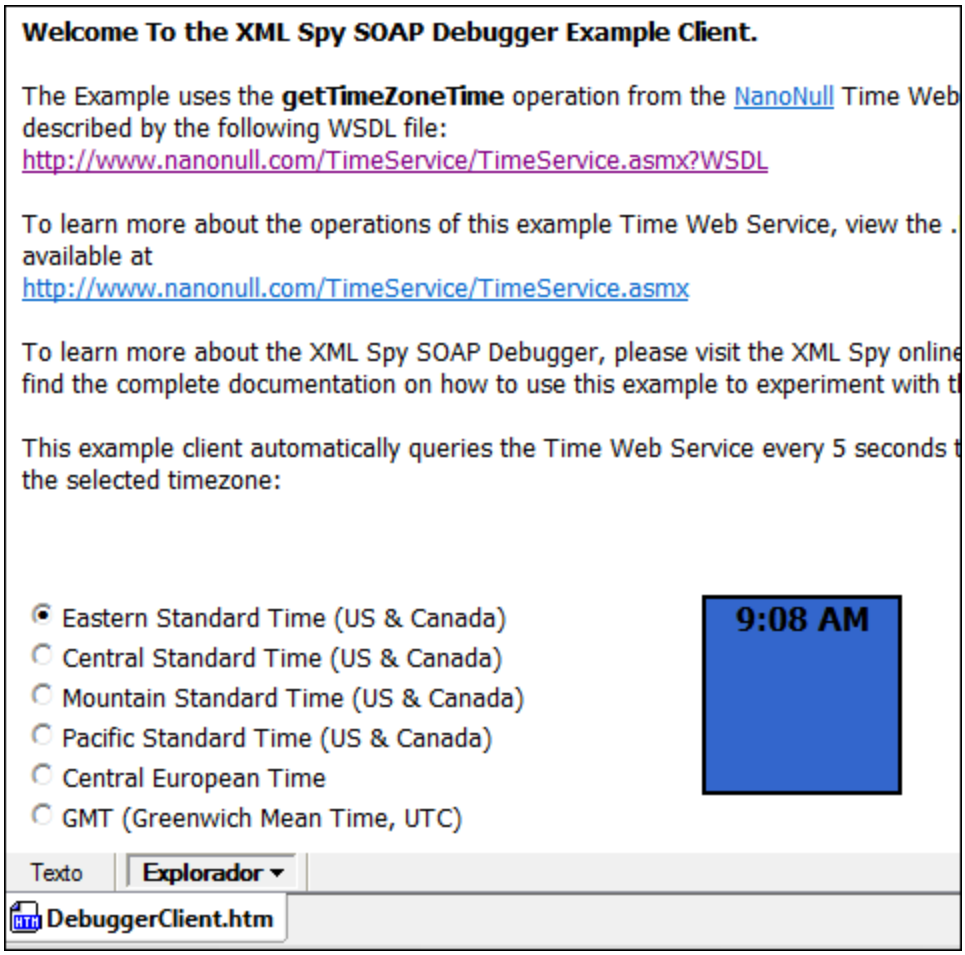
3. Haga clic en el icono + de la carpeta *SOAP Debugger* para ver su contenido. Haga doble clic en `DebuggerClient.htm` para abrir el archivo en XMLSpy.

Nota: otra opción es abrir este archivo con el comando **Archivo | Abrir**. El archivo está en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples`.

El archivo de ejemplo

El archivo `DebuggerClient.htm` tiene este aspecto en la vista Explorador de XMLSpy. Cuando se selecciona uno de los botones de opción, se envía una solicitud SOAP al servicio web Nanonull Time. La respuesta del servicio web aparece en el recuadro coloreado situado a la derecha de los botones de opción.

© 2016-2022 Altova GmbH



Observe que la respuesta a una solicitud de *Eastern Standard Time* aparece en el recuadro en color azul (`9:08`) (*imagen anterior*). Ahora seleccione el botón de opción GMT. En el recuadro no aparece el valor de GMT. En su lugar aparece un mensaje de error y el recuadro se vuelve de color rojo (*imagen siguiente*).



Ahora puede utilizar el depurador SOAP para analizar los mensajes SOAP y localizar el error. En los siguientes subapartados vemos (i) cómo se establecen puntos de interrupción 798, (ii) cómo se ejecuta el depurador SOAP 798 y (iii) cómo se analizan los resultados del depurador para localizar los errores 800.

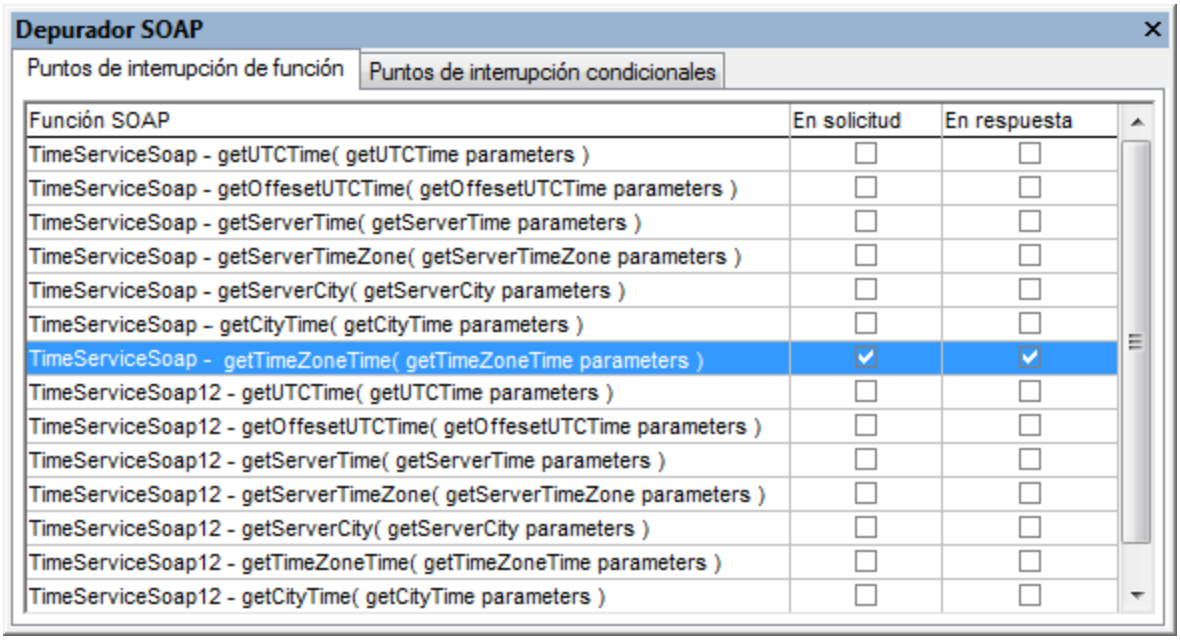
15.2.2.5 Establecer puntos de interrupción

Antes de iniciar la depuración debe establecer puntos de interrupción. Cuando empiece la depuración, el depurador SOAP mostrará las solicitudes y respuestas de los puntos de interrupción que encuentre.

El depurador SOAP enumera los puntos de interrupción (en las pestañas Puntos de interrupción de función y Puntos de interrupción condicionales) en función de la información obtenida del archivo WSDL que seleccionó al iniciar del depurador SOAP[793]. Estos puntos de interrupción están relacionados con las solicitudes SOAP que puede generar el archivo WSDL. Por cada solicitud SOAP se puede seleccionar un punto de interrupción en solicitud y en respuesta (marcando sus casillas en la columna correspondiente, *imagen siguiente*).

En nuestro ejemplo usamos estos archivos:

- `DebuggerClient.htm` como punto de entrada de la solicitud SOAP[796] y
- el archivo WSDL `http://www.nanonull.com/TimeService/TimeService.asmx?WSDL` que seleccionamos cuando iniciamos el depurador SOAP[793].



El servicio web solicitado por `DebuggerClient.htm` usa el método `getTimeZoneTime` para buscar la hora del uso horario seleccionado. En el depurador SOAP, las solicitudes SOAP que se pueden generar a partir del archivo WSDL seleccionado aparecen como puntos de interrupción. Establecemos puntos de interrupción en el método `getTimeZoneTime`, tanto en solicitud como en respuesta (*imagen anterior*). Así podemos analizar si hay errores tanto en las solicitudes SOAP como en las respuestas del servicio web.

Para más información consulte el subapartado Más información sobre puntos de interrupción[802].

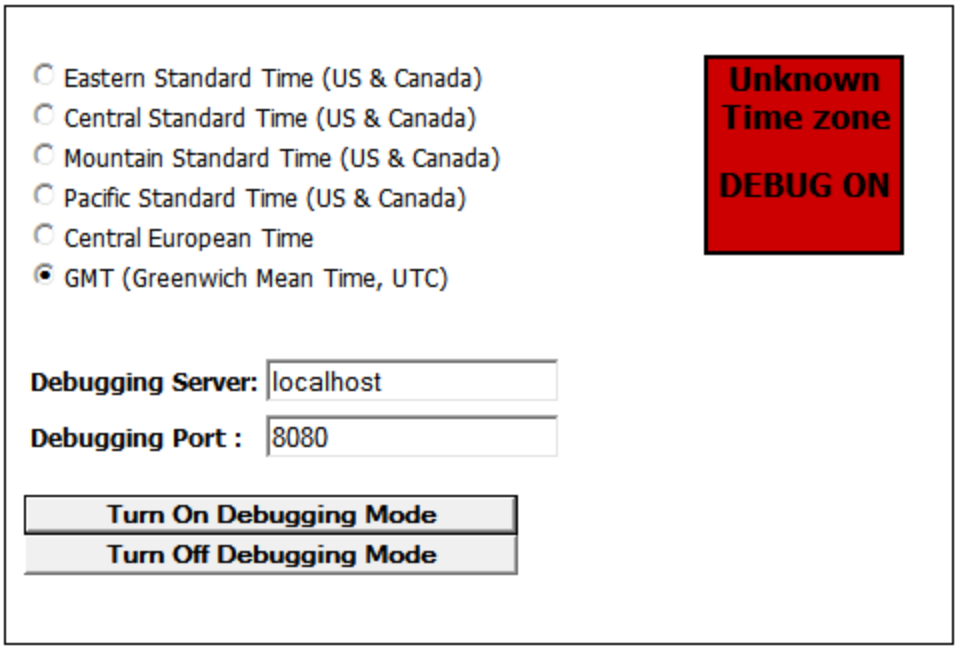
15.2.2.6 Depuración

En nuestro ejemplo utilizamos

© 2016-2022 Altova GmbH

- `DebuggerClient.htm` como punto de entrada de la solicitud SOAP 796 y
- el archivo WSDL `http://www.nanonull.com/TimeService/TimeService.asmx?WSDL` que seleccionamos cuando iniciamos el depurador SOAP 793.

Después de establecer los puntos de interrupción 798, haga clic en el icono Ejecutar  (o use el comando **SOAP | Ejecutar**). Después haga clic en la pestaña *DebuggerClient.htm* para cambiar al archivo de punto de entrada SOAP. Compruebe que la opción *GMT* está seleccionada y haga clic en el botón **Turn On Debugging Mode** (*imagen siguiente*). En el recuadro aparece el mensaje *Debug On* y se envía la solicitud SOAP al depurador SOAP. Los resultados del depurador aparecen en las ventanas Solicitud SOAP y Respuesta SOAP y se describen en el subapartado siguiente Analizar resultados y corregir errores 800.



Controles del depurador SOAP

La barra de herramientas del depurador SOAP (*imagen siguiente*) contiene iconos con los que puede controlar el depurador.



Estos iconos/comandos son, de izquierda a derecha:

- *Ejecutar:* inicia la depuración.
- *Paso a paso:* recorre paso a paso el proceso de solicitud-respuesta, deteniéndose en los puntos de interrupción.
- *Interrumpir en la siguiente solicitud:* el depurador se detiene en la siguiente solicitud SOAP.
- *Interrumpir en la siguiente respuesta:* el depurador se detiene en la siguiente respuesta del servicio web.

- *Detener el servidor proxy:* la depuración se detiene. No olvide que esto no es lo mismo que detener la sesión de depuración SOAP. Para terminar/iniciar la sesión de depuración SOAP, seleccione el comando **SOAP | Sesión de depurador SOAP**.

15.2.2.7 Analizar resultados y corregir errores

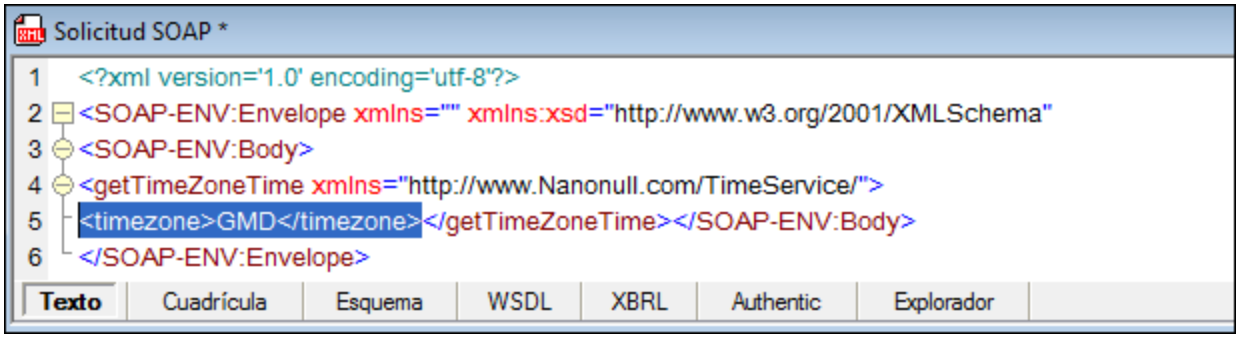
Los resultados del depurador SOAP aparecen en dos ventanas: Solicitud SOAP y Respuesta SOAP. Los puntos de interrupción, que se establecen en las ventanas, están situadas por defecto en la parte inferior de la ventana del depurador SOAP. De acuerdo con los puntos de interrupción establecidos, el depurador SOAP mostrará los resultados en la ventana de resultados correspondiente: en la ventana Solicitud SOAP o en la ventana Respuesta SOAP.

En nuestro ejemplo usamos estos archivos:

- `DebuggerClient.htm` como punto de entrada de la solicitud SOAP [796] y
- el archivo WSDL `http://www.nanonull.com/TimeService/TimeService.asmx?WSDL` que seleccionamos cuando iniciamos el depurador SOAP [793].

Detectar el error

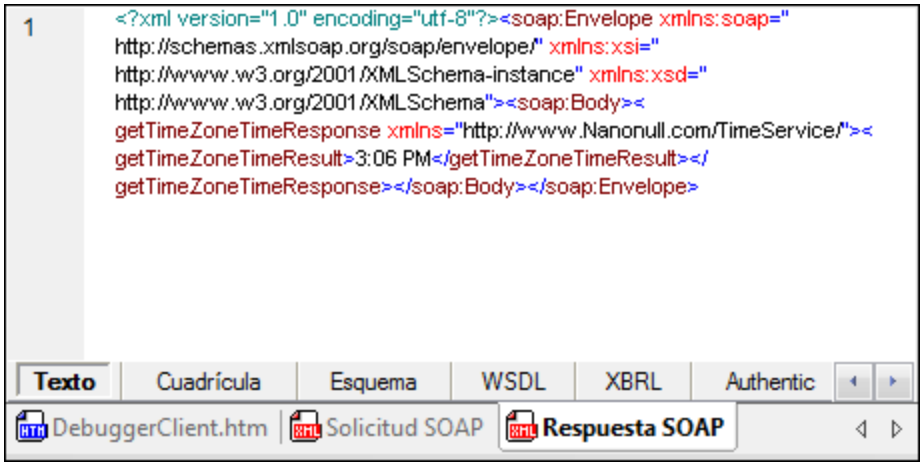
La depuración se inició tal y como describe el subapartado Depuración [798]. La solicitud SOAP para la opción GMT aparece en la ventana Solicitud SOAP del depurador, en la vista Texto. Veamos si la solicitud contiene algún error.



Si miramos el elemento `timezone`, observamos que el valor es GMD. Este valor no es correcto, así que hay que cambiarlo por `GMT`. Para ello hacemos doble clic en el elemento `timezone` y escriba `GMT`.

Ahora hacemos clic en el icono **Ejecutar** de la barra de herramientas (o en el comando **SOAP | Ejecutar**) para enviar la solicitud corregida al servicio web. Unos segundos después aparece la respuesta del servicio web a la solicitud SOAP en la ventana Respuesta SOAP. Seleccione el comando **Vista | Ajuste automático de línea** para ver toda la solicitud SOAP (*imagen siguiente*).

© 2016-2022 Altova GmbH



Ahora haga clic en la pestaña `DebuggerClient.htm` y en el icono **Ejecutar** de la barra de herramientas. El mensaje de error desaparece y en el recuadro aparece la hora correcta (*imagen siguiente*).



Ya puede cerrar la sesión del depurador SOAP (haciendo clic en el comando **SOAP | Sesión de depurador SOAP**).

Corregir el error

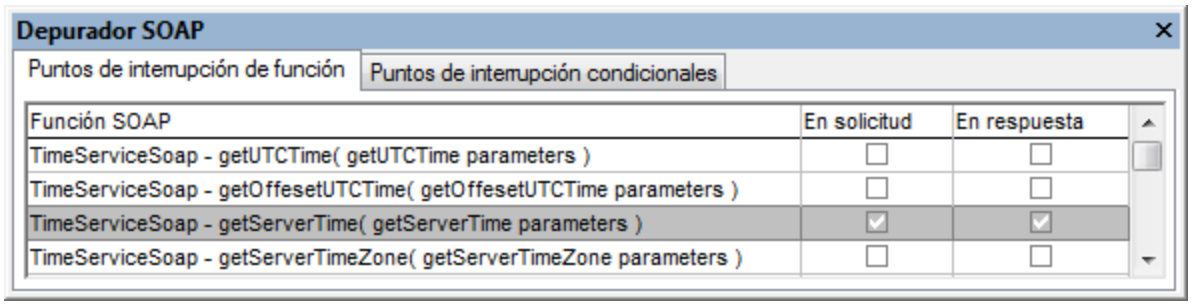
Ahora ya sabemos que en la solicitud SOAP se genera el valor no válido `GMD` en lugar de `GMT`. Si abrimos el archivo de entrada de la solicitud SOAP y buscamos `GMD` (con **Ctrl+F** o **Edición | Buscar**), encontramos la errata (*imagen siguiente*).

```
function changeZones(){
    if (timezone[0].checked)
        msCurrentTimeZone='EST';
    else if (timezone[1].checked)
        msCurrentTimeZone='CST';
    else if(timezone[2].checked)
        msCurrentTimeZone='MST';
    else if(timezone[3].checked)
        msCurrentTimeZone='PST';
    else if(timezone[4].checked)
        msCurrentTimeZone='CET';
    else if(timezone[5].checked)
        msCurrentTimeZone='GMD';
}
```

Si corregimos este error y seleccionamos el botón de opción GMT, el error *Unkown Timezone* ya no aparece. En su lugar aparece la hora GMT. En su lugar aparece la hora GMT.

15.2.2.8 Más información sobre puntos de interrupción

La ventana Depurador SOAP incluye dos pestañas donde puede establecer y eliminar puntos de interrupción (*imagen siguiente*).



Pestaña Puntos de interrupción de función

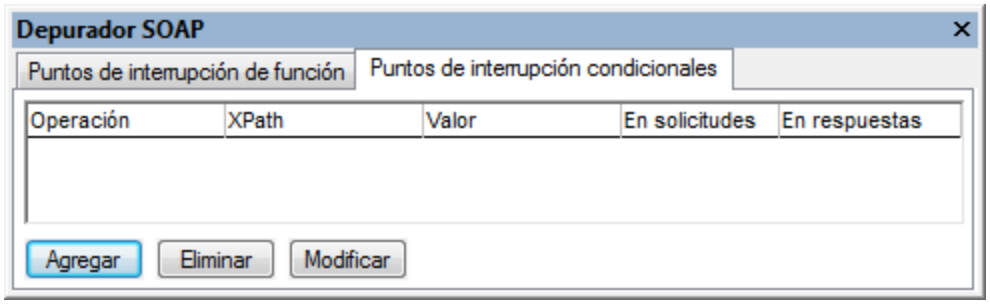
En esta pestaña puede establecer puntos de interrupción en las solicitudes y respuestas a métodos SOAP. El depurador resalta la función que desencadenó el punto de interrupción. Los paquetes de datos recibidos/enviados por el cliente se analizan y se casan con las funciones correspondientes del archivo WSDL. Si se establece un punto de interrupción para un método concreto, el depurador se detiene en este método. En ese momento se habilitan los botones de la barra de herramientas.

Los datos aparecen en las ventanas Solicitud SOAP y Respuesta SOAP. Los documentos SOAP que aparecen en estas ventanas ahora se pueden modificar. Los datos se envían en cuanto haga clic en uno de los iconos de la barra de herramientas (excepto si es el icono Detener servidor proxy).

© 2016-2022 Altova GmbH

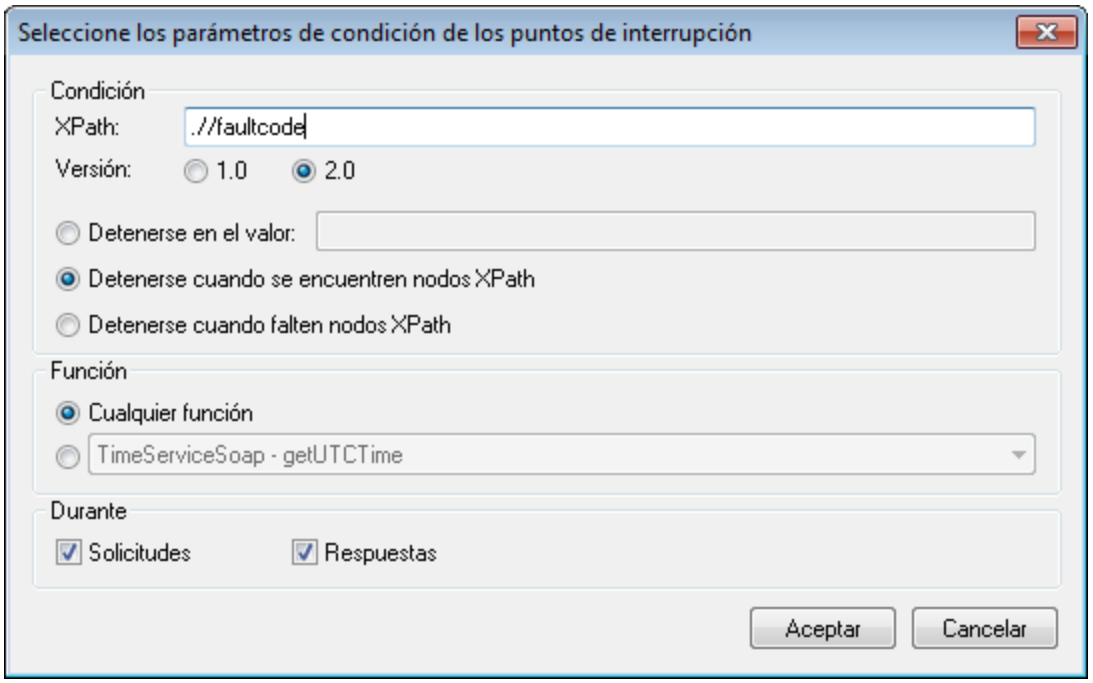
Pestaña Puntos de interrupción condicionales

En esta pestaña (*imagen siguiente*) puede usar expresiones XPath para definir puntos de interrupción. Si una solicitud SOAP causa un error, la respuesta SOAP debe contener un elemento `faultcode` Por tanto, creamos un punto de interrupción que se desencadena cada vez que aparezca un elemento `faultcode`.



Para añadir un punto de interrupción condicional:

1. Seleccione la pestaña Puntos de interrupción condicionales y haga clic en el botón **Agregar**. Aparece este cuadro de diálogo.

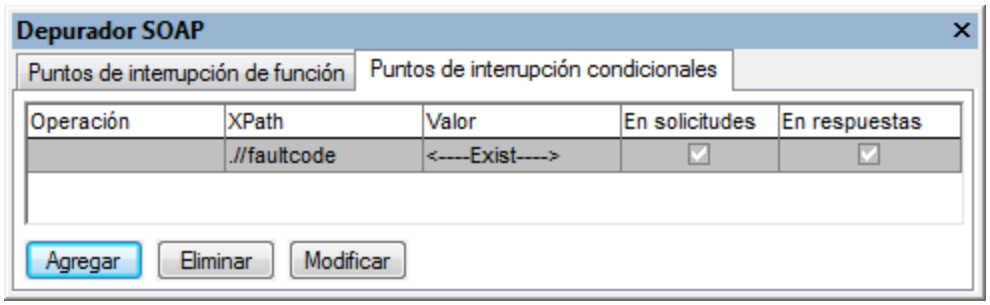


2. Escriba la expresión XPath (p. Ej. `.//faultcode`) en el campo XPath.
3. Seleccione la versión de XPath que desea usar (1.0/2.0) y el botón de opción *Detenerse cuando se encuentren nodos XPath*.
4. Haga clic en **Aceptar** para confirmar la configuración. El depurador SOAP se detendrá cada vez que aparezca el elemento `.//faultcode` en una solicitud o respuesta SOAP.

Estas son las opciones disponibles en el cuadro de diálogo anterior:

- *XPath:* escriba aquí la expresión/el nodo XPath. Para poder seleccionar uno de los botones de opción es necesario introducir aquí una expresión XPath.
- *Versión:* la versión de XPath que desea usar para la expresión.
- *Botones de interrupción:* el depurador se detiene cuando encuentra la opción seleccionada. Las opciones son: (i) detenerse cuando el nodo XPath coincida con el valor introducido en este campo, (ii) detenerse cuando el nodo XPath exista en la solicitud o respuesta SOAP y (iii) detenerse cuando el nodo XPath no exista en la solicitud ni en la respuesta SOAP.
- *Solicitudes/Respuestas:* estas casillas permite definir si las opciones del cuadro de diálogo se aplican a respuestas o solicitudes SOAP.
- *Función:* decida si la condición definida se busca en todos los métodos/funciones (*Cualquier función*) o en una función/un método concreto.

Para la condición definida en la imagen anterior se crea este punto de interrupción condicional en la pestaña *Puntos de interrupción condicionales*:



Estas son las columnas de la pestaña *Puntos de interrupción condicionales*:

- La columna *Operación* contiene el método/la función que se debe buscar. Si seleccionó la opción *Cualquier función* en el cuadro de diálogo anterior, esta columna estará vacía. Si definió un método/una función en el cuadro de diálogo, el método/la función aparece en esta columna.
- La columna *XPath* contiene la expresión XPath definida.
- La columna *Value* contiene el valor XPath con el que se comparan los nodos devueltos. Si seleccionó la opción *Detenerse en el valor* en el cuadro de diálogo, el valor indicado aparece en esta columna. Si seleccionó *Detenerse cuando se encuentren nodos XPath*, en la columna aparece la cadena `<--Exist-->`. Si seleccionó la opción *Detenerse cuando falten nodos XPath*, aparece la cadena `<--Missing-->`.
- Las casillas *En solicitudes* y *En respuestas* indican dónde se debe buscar la condición. Estas casillas se pueden modificar en esta pestaña directamente.

Para editar un punto de interrupción condicional haga doble clic en el su fila o haga clic en el botón **Modificar** (*imagen anterior*). Para eliminar un punto de interrupción condicional, selecciónelo y haga clic en el botón **Eliminar**.

© 2016-2022 Altova GmbH

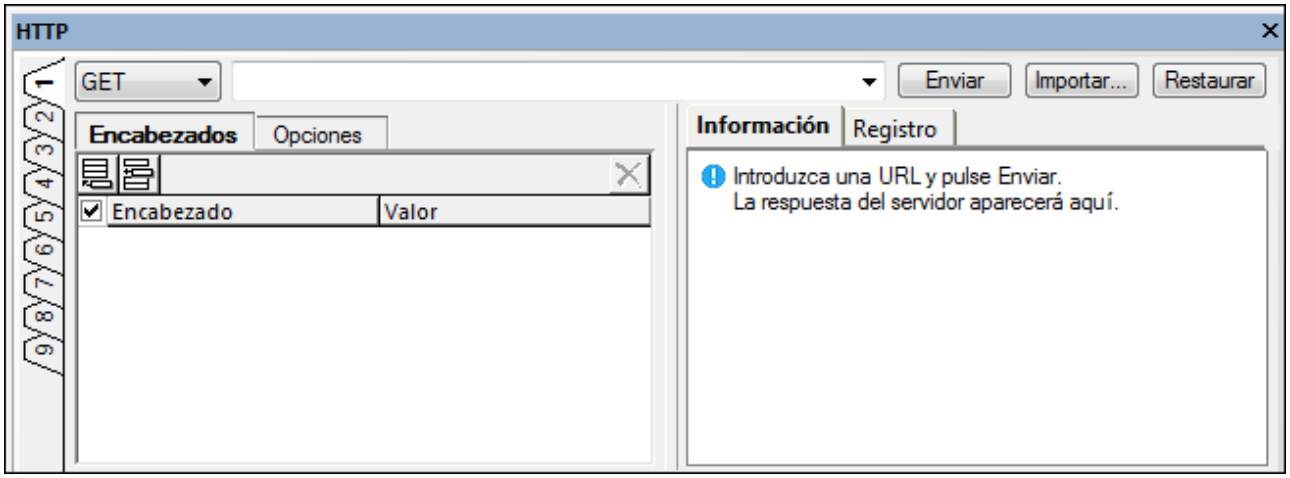
16 HTTP

HTTP (Protocolo de transferencia de hipertexto) es el protocolo (o conjunto de reglas) que define cómo se transmiten archivos (texto, imágenes, audio, vídeo y otros archivos multimedia) por Internet. Cada servidor web ejecuta un programa (denominado *demonio*) que espera continuamente solicitudes HTTP y que se ocupa de ellas a medida que llegan. Por ejemplo, cuando visitamos la página de inicio de un sitio web, el explorador envía un comando HTTP al servidor web del sitio web que solicita la descarga de la página de inicio. El demonio HTTP del servidor recibe la solicitud y envía la página solicitada. Hay una propiedad muy importante de HTTP que debe tenerse en cuenta: HTTP es un sistema que no tiene estado, es decir, cada comando HTTP se lleva a cabo de forma independiente, sin referencia alguna a comandas anteriores ni posteriores.

En XMLSpy puede probar comandos HTTP en la ventana HTTP de resultados 129 (*imagen siguiente*). En esta ventana puede crear y enviar una solicitud HTTP a un servidor web y recibir y consultar la respuesta.

Partes de la ventana HTTP

La ventana HTTP cuenta con nueve pestañas (*imagen siguiente*). En cada pestaña puede almacenar una solicitud y después pasar de una pestaña a otra. Tras crear una solicitud en la ventana, podrá enviarla con solo hacer clic en el botón **Enviar**. La respuesta aparece directamente en la ventana.



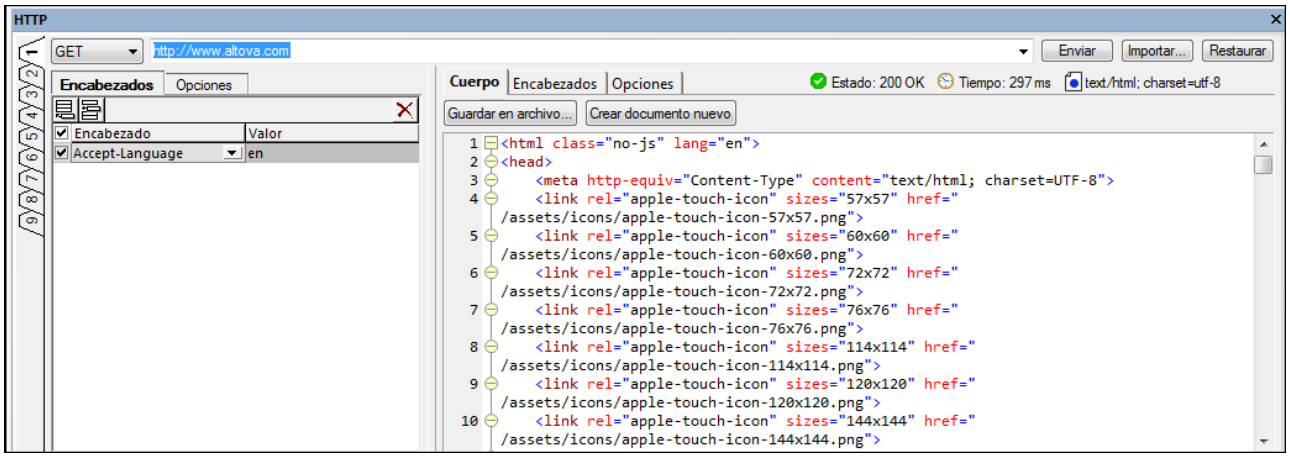
La ventana se divide en varios paneles:

- En el panel superior hay: (i) un cuadro combinado donde puede seleccionar el método HTTP que desea utilizar, (ii) un campo de entada para la URL del servidor web y (iii) botones relacionados con la ejecución de solicitudes HTTP (**Enviar**, **Importar** y **Restaurar**).
- En el panel izquierdo puede crear la solicitud 806.
- En el panel derecho aparece la respuesta 814.

El funcionamiento de la ventana HTTP se describe en los apartados de esta sección.

16.1 Enviar la solicitud

Desde la ventana HTTP puede enviar solicitudes HTTP (*imagen siguiente*). La solicitud se define en el **panel izquierdo de la ventana**, que cuenta con 9 pestañas distintas (una para cada solicitud). Cada solicitud se compone (i) del método HTTP[807] de la solicitud y la URL de destino (que se define en la parte superior de la ventana); (ii) de los encabezados HTTP de la solicitud (en la pestaña *Encabezados*); (iii) de las opciones de configuración de la conexión (en la pestaña *Opciones*) y (iv), en el caso de los métodos POST Y PUT[807], del cuerpo del mensaje HTTP (en la pestaña *Cuerpo*). Con el botón **Restaurar** (en la esquina superior derecha) puede revertir la solicitud a su estado vacío.



Para enviar una solicitud HTTP:

1. En el cuadro combinado de la esquina superior izquierda (*imagen anterior*) seleccione un método HTTP (**`GET`**, **`POST`**, **`PUT`**, **`DELETE`**, **`HEAD`** o **`OPTIONS`**).
2. Introduzca la URL de la página web de destino (p. ej. **`http://www.altova.com`**). También puede introducir **`altova.com`** simplemente porque la aplicación la completará con la parte **`http://`** de la URL.
3. En la pestaña *Encabezados* puede especificar los valores de encabezado HTTP (*imagen anterior*). Puede seleccionar o introducir un encabezado e introducir su valor (aquí puede consultar una lista completa de encabezados HTTP 1.1). Con los botones **Insertar**, **Anexar** y **Eliminar** de la barra de herramientas de la pestaña puede agregar o eliminar encabezados. En lugar de eliminar un encabezado, también puede desactivarlo con solo desactivar su casilla *Activar*, situada a la izquierda del nombre del encabezado. Así, si más adelante necesita utilizar ese encabezado, no tendrá que volver a introducirlo, bastará con volver a activarlo. Consulte también el apartado Aceptar la respuesta[816]. (Si define un valor para un encabezado que se añadiría automáticamente a la hora del envío, entonces el valor que introduzca manualmente se usará en lugar del valor que se añadiría automáticamente).
4. Si está enviando una solicitud **`POST`** o **`PUT`**, aparecerá una pestaña llamada *Cuerpo*. Consulte el apartado El cuerpo de las solicitudes POST y PUT[807] para aprender a crear el cuerpo de una solicitud **`POST`** o **`PUT`**.
5. En la pestaña *Opciones* puede especificar tiempos de espera y opciones de seguridad. Esta pestaña se describe más detalle en el apartado Configuración de la solicitud HTTP[809].
6. Haga clic en el botón **Enviar** (situado en la parte superior derecha de la ventana) para enviar la solicitud.
7. Si desea borrar toda la información y dejar la pestaña en blanco otra vez, haga clic en el botón **Restaurar**. Cuando se hace clic en ese botón: (i) el método seleccionado es el primer método de la

© 2016-2022 Altova GmbH

lista desplegable del cuadro combinado (es decir, **GET**); (ii) se borra la URL que estaba en el cuadro de texto; (iii) se borran todas las definiciones de encabezado, todas las opciones de configuración y todas las definiciones de cuerpo.

Nota: también puede importar una solicitud de un archivo WADL en la ventana HTTP haciendo clic en el botón **Importar**.

Nota: la solicitud se envía con codificación UTF-8. Las demás codificaciones se pasan a UTF-8 y solo se envían datos UTF-8.

Métodos HTTP

Estos son los métodos HTTP compatibles:

GET
El método **GET** solicita el recurso situado en la URL especificada. También puede agregar una consulta en la URL (p. ej. **http://www.altova.com?nombre1=valor1&nombre2=valor2**. El recurso se devuelve en un mensaje que contiene un encabezado y un cuerpo.

HEAD
El método **HEAD** es idéntico al método **GET**, pero no devuelve el cuerpo del mensaje, solo su encabezado con metadatos sobre el recurso que está situado en la URL especificada.

POST
El método **POST** sirve para actualizar un recurso que ya existe en la URL especificada o para crear un recurso nuevo en la URL especificada. Los datos que se deben enviar al recurso se colocan en el cuerpo de la solicitud HTTP (*véase* *El cuerpo de las solicitudes POST y PUT* 807).

PUT
El método **PUT** sirve para crear un recurso nuevo en la URL especificada. Los datos que se deben enviar al recurso se colocan en el cuerpo de la solicitud HTTP (*véase* *El cuerpo de las solicitudes POST y PUT* 807).

DELETE
El método **DELETE** elimina el recurso situado en la URL especificada.

OPTIONS
El método **OPTIONS** devuelve una lista de métodos HTTP que son compatibles con el servidor.

El cuerpo de las solicitudes POST y PUT

En el caso de las solicitudes **POST** y **PUT**, se habilita una pestaña más llamada *Cuerpo,* donde se puede especificar el cuerpo de la solicitud **POST** o **PUT** (*imagen siguiente*). La pestaña *Cuerpo* tiene dos modos: el **modo de edición** y el **modo de archivo**. Puede alternar entre ambos modos con los botones de la barra de herramientas, situados en la esquina izquierda de la pestaña *Cuerpo* (*imagen siguiente*). En el modo de edición (el que está seleccionado en el ejemplo de la imagen), puede editar la solicitud HTTP en el panel directamente, mientras que en el modo de archivo deberá seleccionar un archivo que contenga el cuerpo de la solicitud HTTP

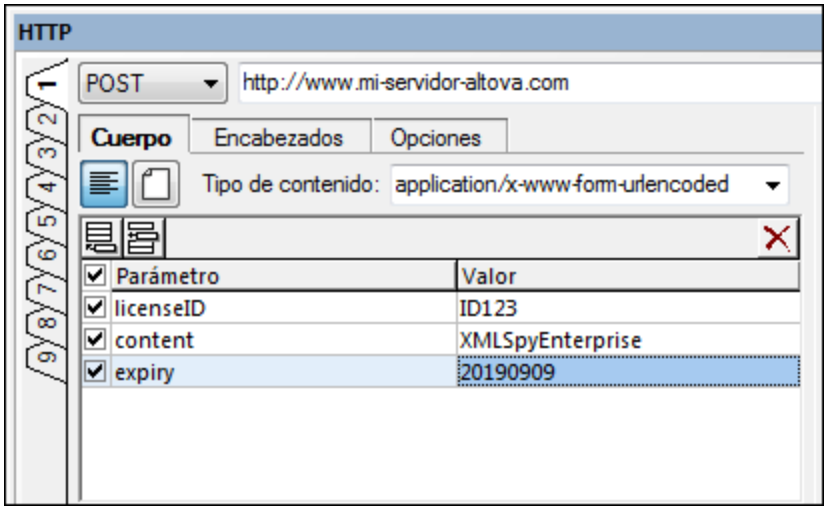
El campo *Tipo de contenido* permite especificar el encabezado **Content-Type** de la solicitud. Las opciones del cuadro combinado de este campo son distintas en cada modo (modo de edición y modo de archivo). Puede seleccionar una opción de la lista del cuadro combinado o introducir un tipo MIME. Observe que el valor

especificado en este campo invalida cualquier otro encabezado **`Content-Type`** que se especificara en las pestañas *Encabezados* o *Cuerpo*.



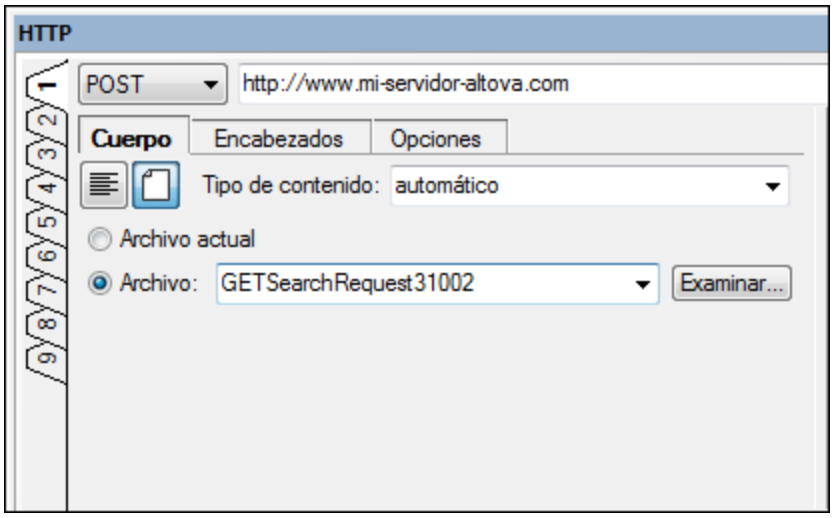
Modo de edición
El estilo del editor dependerá del tipo de contenido seleccionado:

- Para los tipos de contenido **`text/plain`**, **`text/xml`**, **`application/xml`** y **`application/json`** se trata de un editor de texto con funciones de edición inteligente como color de sintaxis para documentos XML y JSON y numeración de líneas. En la imagen anterior puede ver el editor para tipos de contenido `text/xml`. El cuerpo de la solicitud se especifica en el campo *Tipo de contenido* y no se puede invalidar con ninguna otra entrada de la solicitud.
- Para los tipos de contenido **`application/x-www-form-urlencoded`** el editor es una vista en forma de cuadrícula (*imagen siguiente*) donde cada línea representa un par nombre/valor en el cuerpo de la solicitud.



Modo de archivo
En el modo de archivo (*imagen siguiente*) el cuerpo de la solicitud será el contenido del archivo seleccionado. Este archivo puede ser el archivo que está activo en la ventana principal (opción *Archivo actual*) o un archivo externo.

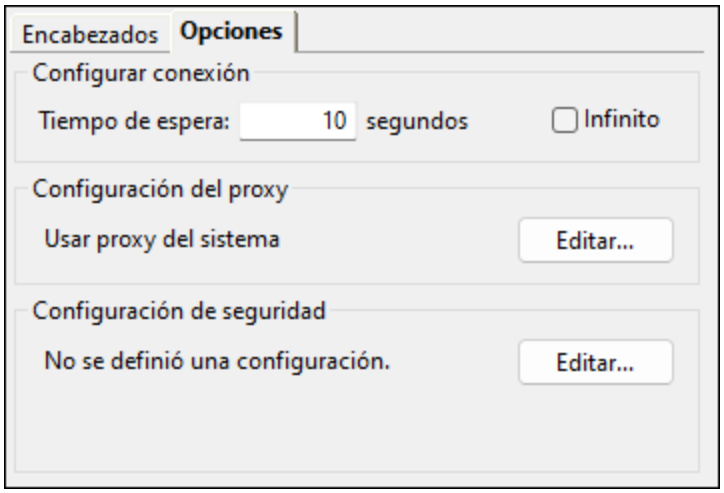
 © 2016-2022 Altova GmbH



Los encabezados y las opciones de configuración de la solicitud se pueden especificar igual que para las demás solicitudes (es decir, en las pestañas *Encabezados* y *Opciones* respectivamente).

Configuración de la solicitud HTTP

En la pestaña *Opciones* de la ventana HTTP (*imagen siguiente*) puede definir (i) el tiempo de espera de la conexión y (ii) las opciones de seguridad de una solicitud. Recuerde que puede definir distintas opciones de configuración en cada una de las nueve pestañas de la ventana HTTP. En la imagen que aparece a continuación puede ver las opciones de configuración para una URL HTTPS.

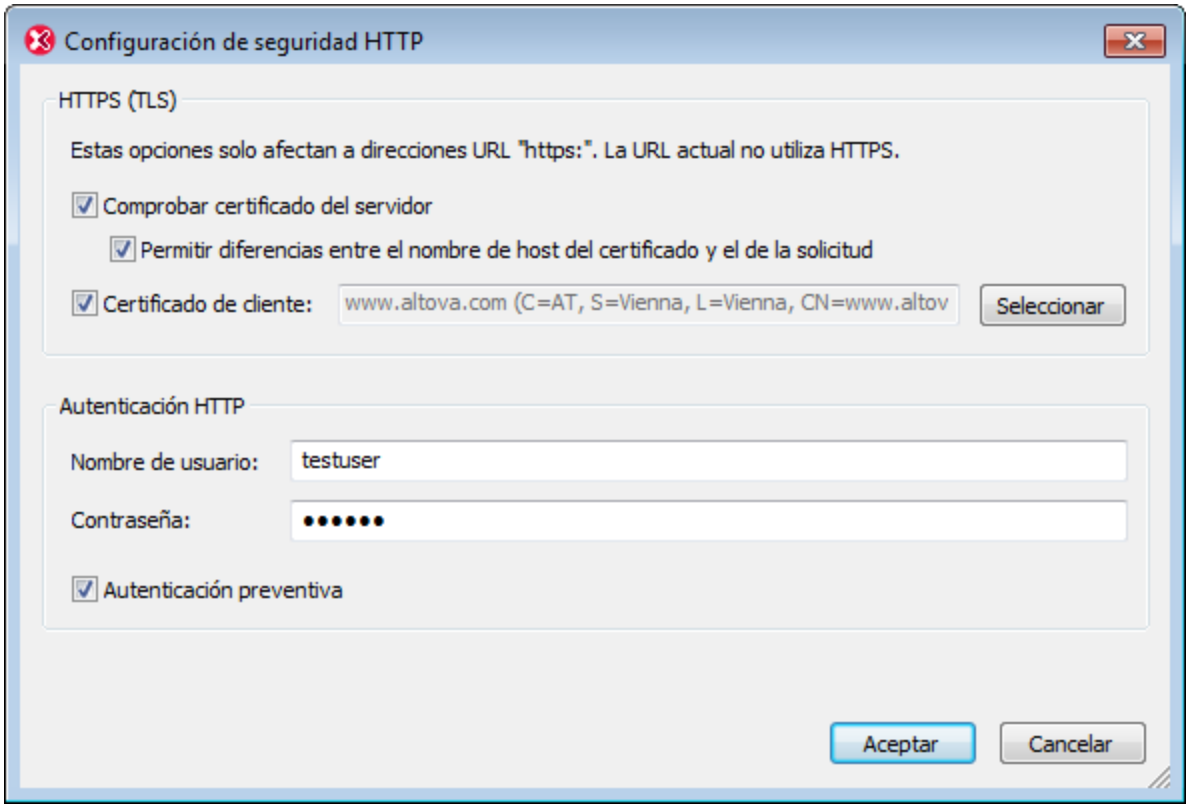


Tiempo de espera

Puede especificar durante cuántos segundos intentará XMLSpy establecer la conexión con el servidor web. Si se supera este plazo de tiempo sin que se consiga establecer la conexión, entonces recibirá el error **`Error de entrada/salida 28: Se alcanzó el tiempo de espera`**. Si prefiere no especificar el tiempo de espera, marque la casilla *Infinito*.

Configuración del proxy
Aquí puede ver la configuración actual del proxy y un botón que le permite acceder a la sección del proxy del cuadro de diálogo Opciones[1627].

Configuración de seguridad
Haga clic en el botón Editar para editar la configuración de seguridad de una solicitud en el cuadro de diálogo "Configuración de seguridad HTTP" (*imagen siguiente*). Aquí puede especificar opciones de seguridad HTTPS y definir las credenciales de autenticación HTTP para la solicitud que se hace desde la pestaña actual. Si el servidor web de destino de la solicitud no usa SSL, entonces solo se usarán las credenciales de autenticación HTTP. Si el servidor web de destino utiliza SSL, entonces se usarán ambas credenciales.



- *Configuración HTTPS:* la opción *Comprobar certificado del servidor* está activada por defecto y puede especificar si el nombre de host de la solicitud puede diferir del nombre de host del certificado. Si está apuntando a una URL de una intranet (p. ej. de la red de su compañía), entonces se puede usar un certificado cliente (que suele estar en el almacén de certificados local) para verificar el certificado del servidor de la intranet.
- *Autenticación HTTP:* algunas solicitudes requieren la autenticación del usuario. Cuando así sea, puede introducir un nombre de usuario y una contraseña y cuando el servidor exija una autenticación, recibirá automáticamente las credenciales dadas aquí. De lo contrario, recibirá un aviso solicitando credenciales. Cuando la solicitud inicial al servidor contiene la información de autenticación, este proceso se conoce como *autenticación preventiva*. Si el servidor requiere este tipo de autenticación, marque la casilla *Autenticación preventiva.*

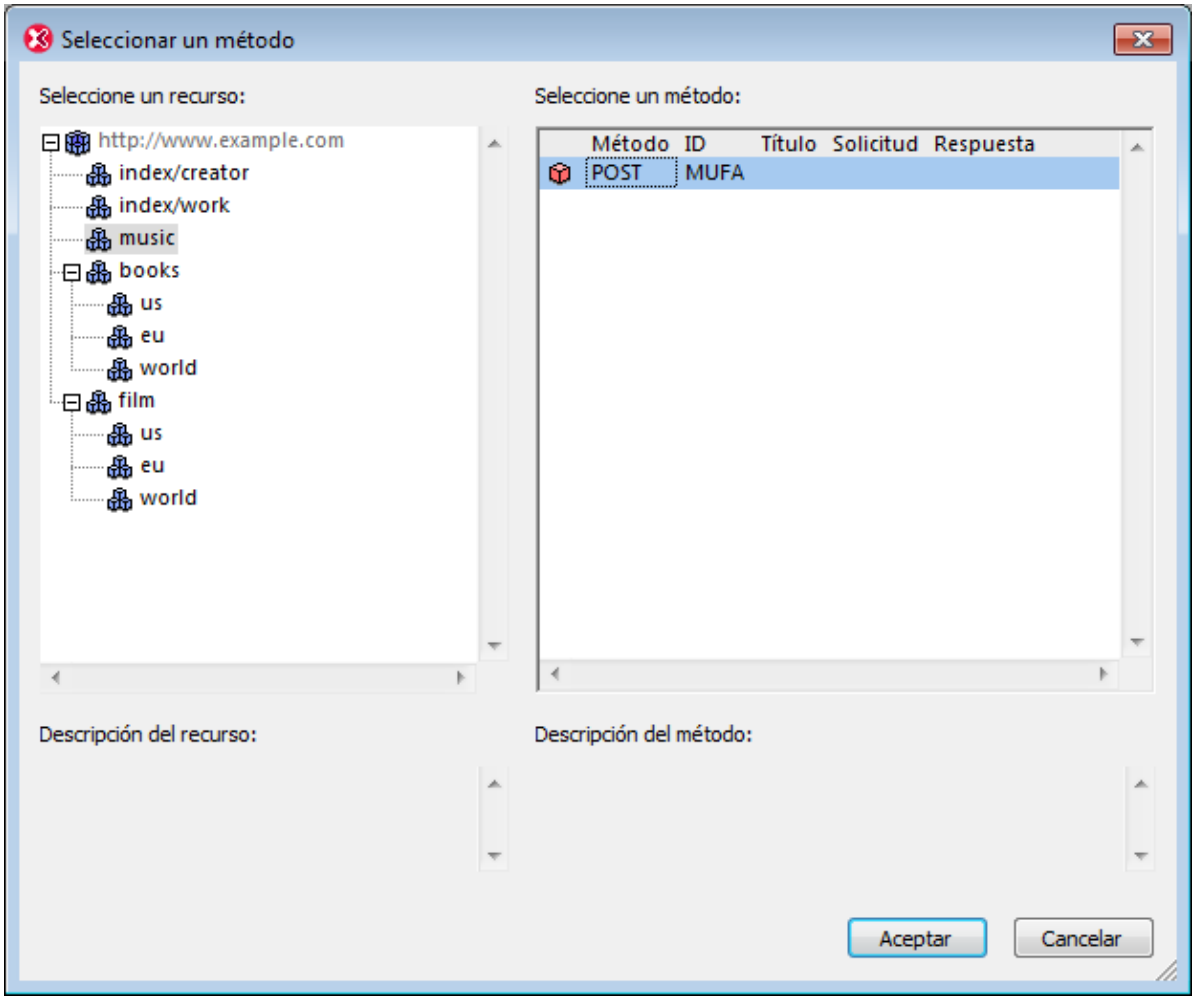
© 2016-2022 Altova GmbH

16.2 Importar una solicitud para enviarla

En la ventana HTTP [805] puede importar una solicitud de un archivo WSDL 1.1, WSDL 2.0 o WADL (Web Application Development Language) y después enviarla. Esto se consigue con la ayuda del asistente de importación WSDL/WADL de XMLSpy. El asistente abre un archivo WSDL WADL, selecciona una solicitud de uno de los extremos WSDL del archivo o de uno de los recursos WADL, ofrece al usuario la posibilidad de modificar los parámetros editables de la solicitud y después importa esa solicitud a la ventana HTTP [805].

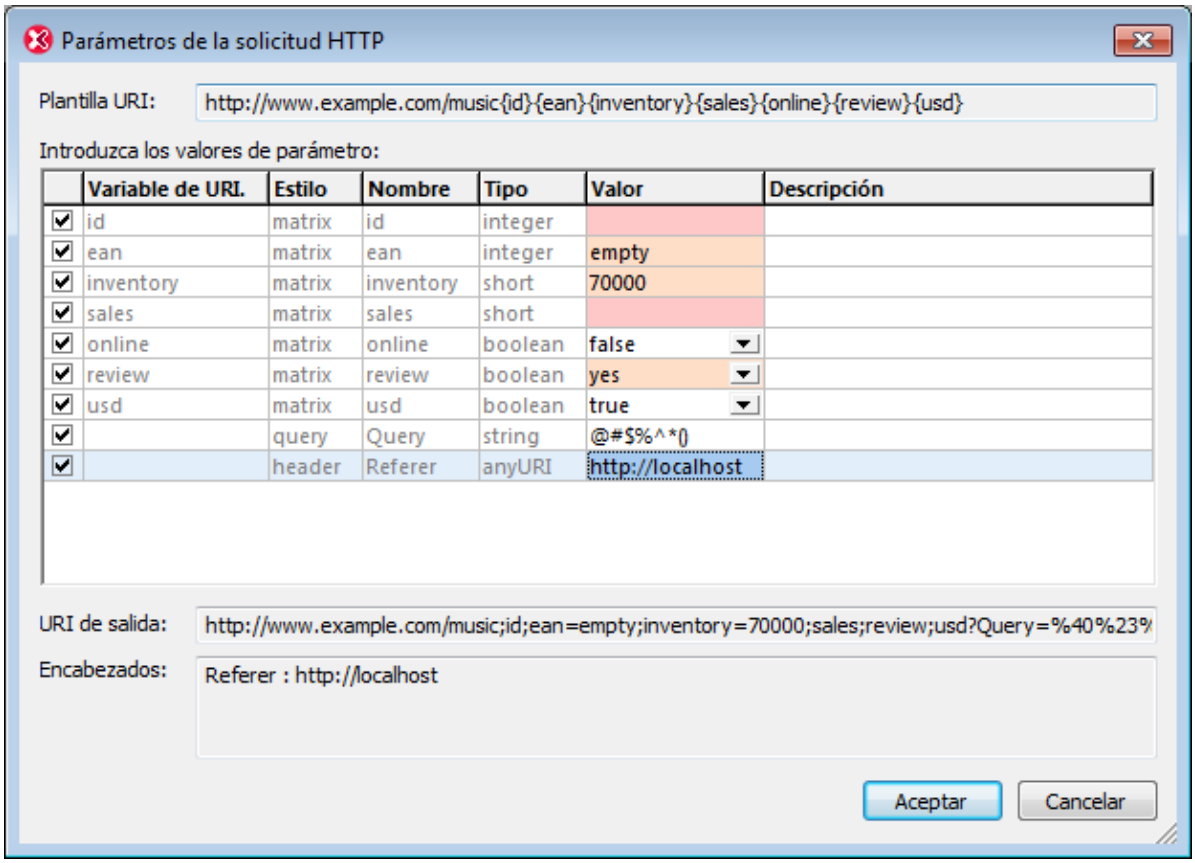
Para importar una solicitud de un archivo WADL:

1. En la ventana HTTP [805] haga clic en el botón **Importar** para iniciar el asistente de importación WSDL/WADL.
2. Aparece un cuadro de diálogo de selección de archivos. Navegue hasta el archivo WSDL o WADL que contiene la solicitud que desea importar y haga clic en **Aceptar**. Esto inicia el asistente de importación WSDL/WADL (*imagen siguiente*).



3. En el panel izquierdo (*imagen anterior*) seleccione el extremo WSDL o recurso WADL correspondiente (el que contiene la solicitud que desea importar).

4. En el panel derecho (*imagen anterior*) seleccione la solicitud (la operación WSDL o el método WADL) que quiere importar. Tenga en cuenta que para importar por WDSL: (i) la importación HTTP solo está disponible para extensibilidades SOAP o HTTP y (ii) en este panel (a la derecha) solo se muestran los enlaces compatibles (SOAP y HTTP)
5. Haga clic en **Aceptar**. Si la solicitud contiene uno o más parámetros, entonces la siguiente pantalla del asistente (*imagen siguiente*) muestra los parámetros de la solicitud que seleccionó; en caso contrario la solicitud se importa a la ventana HTTP 805 y se cierra el ayudante (*véase el punto 7 más abajo*). Los parámetros son parte de la solicitud. Por ejemplo, en una solicitud de búsqueda un parámetro puede ser el término de búsqueda. El asistente valida el valor de un parámetro con su tipo de datos y puede devolver tres estados distintos: (i) si el color de fondo es rosa, se trata de un valor no válido; (ii) si el color de fondo es beige, entonces el valor es incorrecto pero se introducirá y enviará de todas maneras en la solicitud; (iii) y, si el color de fondo es blanco, se trata de un valor válido.



6. En la pantalla *Parámetros de la solicitud HTTP* (*imagen siguiente*), introduzca o edite los valores de parámetros según corresponda (en la columna *Valores*). Quizás algunos valores de parámetro no se puedan editar (las celdas de estos valores están deshabilitadas). Si hay algún parámetro que no quiera usar, desactive su casilla *Activar* (primera columna). Recuerde que algunos parámetros son obligatorios y su casilla *Activar* está bloqueada. Además, tenga en cuenta que, a medida que edite los valores de parámetro, la solicitud se irá generando en el campo *URI de salida:* (situada en la parte inferior de la pantalla). La cuadrícula de parámetros también incluye filas para los encabezados (al final de la cuadrícula). Estos encabezados proceden del archivo WADL y sus valores se pueden editar en la cuadrícula si así lo permiten las definiciones del archivo WADL. En el cuadro *Encabezados* situado al final de la pantalla puede ver una lista resumida de todos los encabezados.
7. Haga clic en **Aceptar**. La solicitud se importa en la ventana HTTP 805 y aparece de la siguiente manera: (i) en el cuadro combinado del método aparece el método HTTP de la solicitud; (ii) la URL se

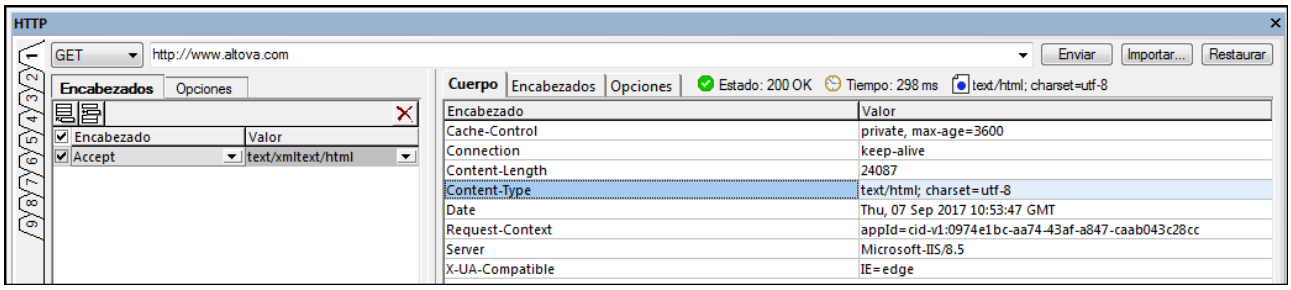
 © 2016-2022 Altova GmbH

construirá a partir de los parámetros de la solicitud; (iii) los encabezados HTTP de la solicitud se introducirán en la pestaña *Encabezados*. Tenga en cuenta que si la solicitud es de tipo **POST** o **PUT**, entonces el cuerpo de la solicitud no se introducirá en la pestaña *Cuerpo* (deberá añadirse a mano).

8. Repase la configuración de la pestaña *Opciones* y compruebe si se deben hacer cambios.
9. Haga clic en **Enviar** para enviar la solicitud.

16.3 Recibir la respuesta

La respuesta de una solicitud HTTP se recibe en el panel derecho de la ventana HTTP (*imagen siguiente*).

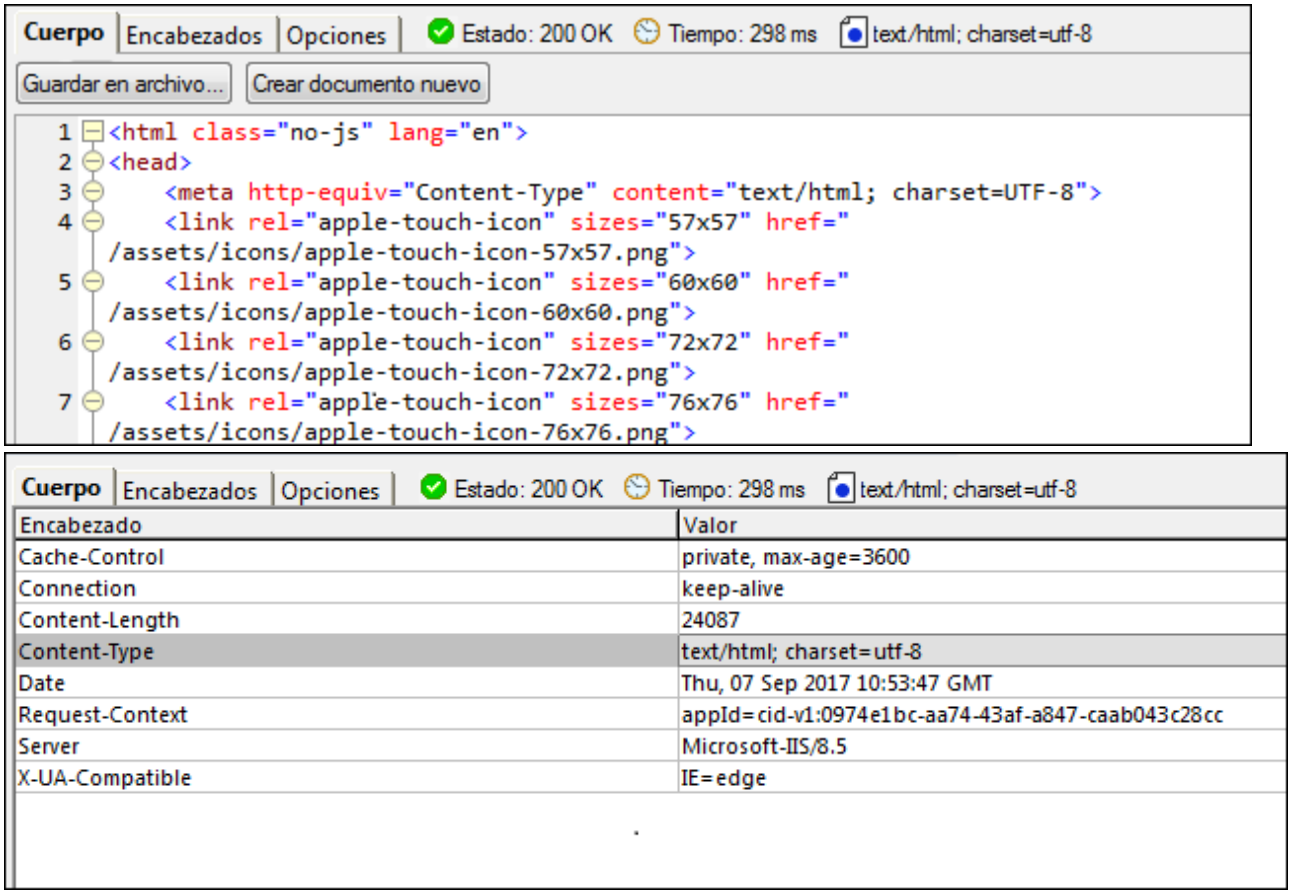


A la derecha de las pestañas *Cuerpo* y *Encabezados* aparece importante información sobre la respuesta:

- el código de estado HTTP (*ver más abajo* 815),
- el tiempo transcurrido entre que se estableció la conexión y se recibió la última porción de la respuesta y
- el tipo de contenido **`Content-Type`** de la respuesta.

El panel de respuesta tiene dos pestañas: *Cuerpo* (*imagen izquierda*) y *Encabezados* (*imagen derecha*).

© 2016-2022 Altova GmbH



- *Pestaña Cuerpo:* el cuerpo de la respuesta aparece con color de sintaxis si el documento es HTML, XML o JSON y con numeración de líneas. La pestaña cuenta con dos botones: (i) **Guardar en archivo** para guardar el cuerpo de la respuesta en un archivo y (ii) **Crear documento nuevo** para crear un documento nuevo en XMLSpy y verlo en la ventana principal de la interfaz gráfica del usuario. Los documentos nuevos creados con este botón se pueden editar y guardar como cualquier otro documento. Si no se puede crear un documento nuevo a partir del cuerpo de la respuesta (p. ej. si el cuerpo es una imagen), entonces se deshabilita el botón **Crear documento nuevo**.
- *Pestaña Encabezados:* contiene los encabezados de la respuesta. En la parte superior del panel también aparece el encabezado `Content-Type`.
- *Pestaña Registro:* en esta pestaña puede ver los eventos y la información relativos a la solicitud. Cada línea viene numerada. Puede guardar el registro en un archivo o crearlo como documento nuevo en la ventana principal. Si lo crea como documento nuevo lo puede editar y guardar como de costumbre.

Códigos de estado HTTP

Los códigos de estado se pueden dividir en varias categorías:

- Los códigos 2XX se usan para solicitudes procesadas correctamente.
- Los códigos 3XX se usan para redireccionamientos.
- Los códigos 4XX se usan si se detectó un problema con la solicitud.
- Los códigos 5XX se usan si se detectó un problema con el servidor.

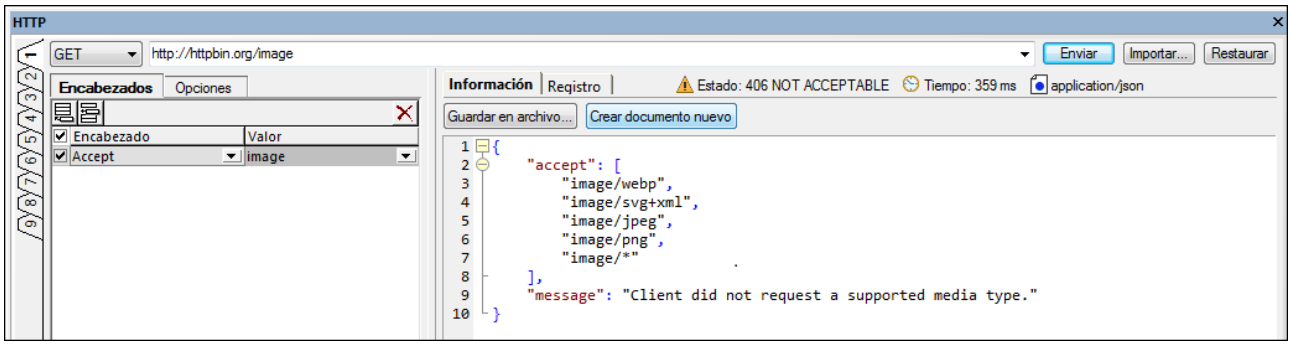
Estos códigos suelen encontrarse con bastante frecuencia:

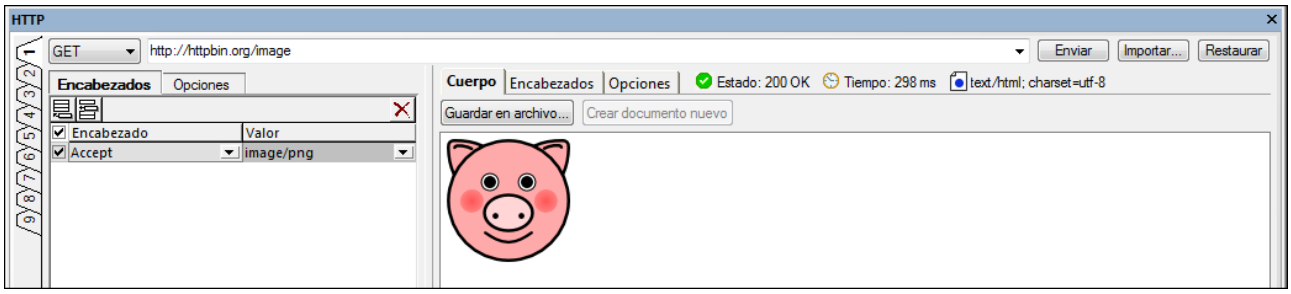
- *200 Correcto*: se envía como respuesta a una solicitud que se procesó correctamente.
- *206 Contenido parcial*: el servidor envía solo parte del recurso porque solo se solicitó parte del recurso.
- *301 Movido permanentemente*: la solicitud debería redireccionarse a la URL dada.
- *401 No autorizado*: el recurso requiere autenticación y la autenticación dio error o no se aportaron credenciales.
- *403 Prohibido*: solicitud válida, pero el servidor se niega a actuar. Esto puede deberse a que el usuario no tiene las credenciales necesarias.
- *404 No se encontró*: no se encontró el recurso.
- *500 Error interno del servidor*: mensaje de error genérico. Se envía cuando no se dispone de más información.

Consulte también: Lista completa de códigos de estado HTTP.

El encabezado Accept

El encabezado Accept de la solicitud especifica el tipo de contenido que se debe aceptar en la respuesta. Por ejemplo, en las imágenes que aparecen a continuación puede ver la diferencia entre las respuestas cuando se usa **`Accept=image`** (*primera imagen*) y cuando se usa **`Accept=image/png`** (*segunda imagen*). En la primera imagen, como se especifica qué formato de imagen debe enviarse en la respuesta, se envía un código de estado de error y un mensaje JSON con más información.





© 2016-2022 Altova GmbH

17 XBRL

La vista [264] **XBRL** [264] de XMLSpy es un editor de taxonomías XBRL que ofrece un resumen gráfico de sus taxonomías XBRL e incluye funciones de edición inteligente. En esta sección describimos cómo se crean y editan taxonomías en la vista XBRL.

Esta sección está organizada de la siguiente manera:

- Gestor de taxonomías [818], que describe cómo usar la herramienta Gestor de taxonomías para instalar, actualizar y administrar taxonomías para usarlas con XMLSpy.
- Procedimientos básicos [834]: esta sección describe cómo crear taxonomías que contienen los componentes más básicos.
- Procedimientos adicionales [859]: esta sección describe procedimientos adicionales, como las etiquetas preferidas o los hechos duplicados.
- Editor de fórmulas XBRL [864]: esta sección explica cómo trabajar con fórmulas XBRL en la vista **XBRL**.
- Editor de definiciones de tabla XBRL: [889] describe la estructura de las tablas [892], cómo usar el editor para definir tablas XBRL y cómo funciona la vista previa del diseño de las tablas XBRL [916]. Esta sección también explica cómo usar parámetros de tabla [912] y cómo usar parámetros de tabla con conjuntos de tabla.
- Buscar en XBRL [930]: esta sección describe las potentes funciones de búsqueda de XMLSpy para XBRL.
- OIM [936], que proporciona una vista general de las funcionalidades OIM de XMLSpy.
- Notas sobre la validación de instancias y taxonomías XBRL [937].

Para más información, consulte el apartado *Vistas de edición | Vista XBRL* [264] y la descripción de los comandos del menú [1515] **XBRL** [1515]. Por ejemplo, si desea saber más sobre la generación de documentación de la taxonomía, consulte el apartado *Comandos de menú | Menú XBRL | Generar Documentación* [1523].

En la vista **XBRL** puede crear firmas XML [411] para archivos XBRL como archivos de firma externos. Para más información sobre cómo trabajar con firmas digitales, consulte el apartado Firmas XML [411].

Compatibilidad con taxonomías US-GAAP y otras taxonomías

XMLSpy es compatible con estas taxonomías:

- US-GAAP 1.0, 2005, 2008, 2009, 2011 to 2022
- NIIF

Las versiones más recientes de US-GAAP se instalan con XMLSpy. Además, Altova ofrece un programa de instalación para instalar varias taxonomías más, incluidas las taxonomías US-GAAP de años anteriores. Este programa de instalación se puede descargar del sitio web de Altova.

Certificación XBRL

XMLSpy tiene la certificación XBRL de

has been certificación XBRL de XBRL International. Para más información sobre esta certificación consulte XBRL Software Certification.

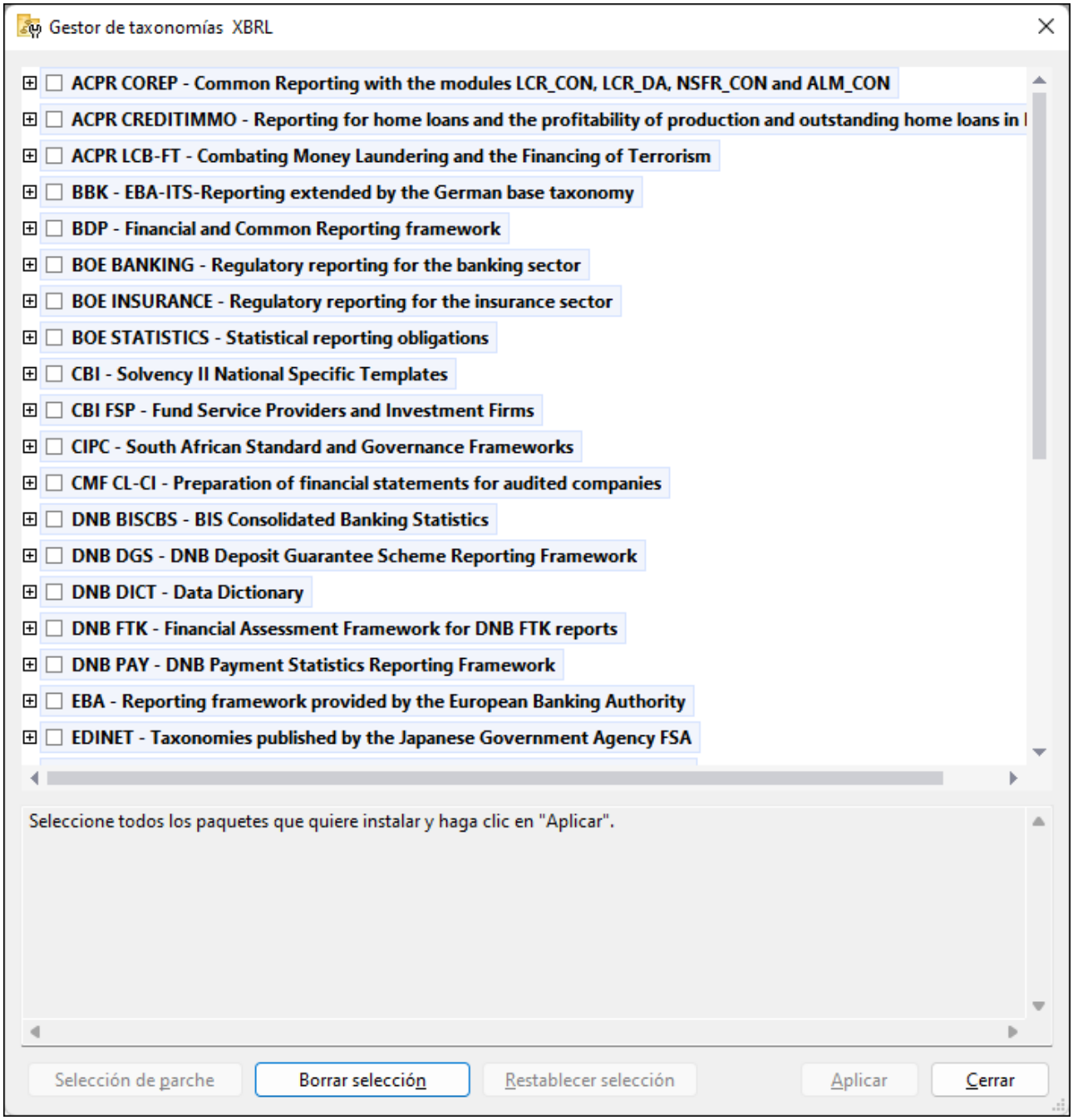
Sitio web de Altova: Editor de taxonomías XBRL, Validador XBRL

17.1 Gestor de taxonomías

El Gestor de taxonomías XBRL es una herramienta que ofrece una forma centralizada de instalar y administrar taxonomías XBRL para usarlas en todas las aplicaciones de Altova compatibles con XBRL, incluido XMLSpy.

- En Windows, el gestor tiene una interfaz gráfica del usuario (*imagen siguiente*) a la que también puede acceder desde la línea de comandos. (Las aplicaciones de escritorio de Altova solo están disponibles para Windows; *consulte la lista siguiente*.)
- En Linux y macOS el Gestor de taxonomías solo está disponible en la línea de comandos. (Las aplicaciones de escritorio de Altova están disponibles para Windows, Linux y macOS; *consulte la lista siguiente*.)

© 2016-2022 Altova GmbH



Aplicaciones de Altova compatibles con XBRL

Aplicaciones de escritorio (solo para Windows)	Aplicaciones de servidor (Windows, Linux, macOS)
Complementos XBRL de Altova para Excel (EBA, ESEF, Solvency II)	MapForce Server (ediciones Standard y Advanced)

MapForce Enterprise Edition	RaptorXML+XBRL Server
StyleVision Enterprise Edition	StyleVision Server
XMLSpy Enterprise Edition	

Instalación y desinstalación del Gestor de taxonomías

El Gestor de taxonomías se instala automáticamente al instalar cualquiera de las aplicaciones de Altova compatibles con XBRL o el Altova Mission Kit Enterprise Edition (*véase la tabla de más arriba*). También se elimina automáticamente si desinstala todas las aplicaciones de Altova compatibles con XBRL o el Altova Mission Kit Enterprise Edition del equipo.

Características de Gestor de taxonomías

El Gestor de taxonomías permite:

- Ver las taxonomías XBRL que hay instaladas en su equipo y comprobar si hay versiones nuevas para descargar.
- Descargar las versiones más recientes de las taxonomías XBRL independientemente del ciclo de versiones de Altova. Altova guarda todas las taxonomías en un sistema de almacenamiento en línea al que tiene acceso el Gestor de taxonomías y desde donde puede descargarlas tan pronto como estén disponibles.
- Instalar o desinstalas cualquiera de las múltiples versiones de una taxonomía en concreto (o todas ellas, si las necesita).
- Una sola taxonomía XBRL representa un "paquete", pero puede tener dependencias en otras taxonomías. Al instalar o desinstalar una taxonomía, se detectan e instalan o desinstalan también automáticamente todas sus dependencias. La interfaz gráfica del usuario (o la línea de comandos, en su caso) le informa cuando se añaden o eliminan taxonomías.
- Las taxonomías XBRL administradas con el Gestor de taxonomías pueden usar el catálogo XML, que permite resolver referencias a URI en documentos de instancia o esquema desde archivos locales, en vez de a través de Internet.
- Todas las taxonomías principales están incluidas en Gestor de taxonomías y se actualizan de forma periódica a la versión más reciente. De esta forma puede administrar todas las taxonomías desde un punto común y tenerlas siempre listas para las aplicaciones de Altova que las usan.
- Los cambios que se realizan en el Gestor de taxonomías afectan a todos los productos de Altova que estén instalados en ese equipo.

Taxonomías XBRL personales

Si necesita trabajar con taxonomías XBRL que no estén incluidas en el Gestor de taxonomías, puede habilitarlas así:

- *Desde las aplicaciones de escritorio de Altova:* ejecute el comando de menú **Herramientas | Opciones** y después vaya a la página de opciones *XBRL* | *Paquetes de taxonomía*. Desde allí navegue hasta el paquete .zip de la taxonomía XBRL personal que quiera usar. Para más información consulte la descripción de este comando en la documentación del producto correspondiente.
- *Desde las aplicaciones servidor de Altova:* desde una línea de comandos compatible con XBRL proporcione las opciones **`-taxonomy-package`** o **`---taxonomy-package-config-file`**. Por ejemplo, estas opciones son compatibles con los comandos de validación XBRL como **`valxbrl`** o **`valxbrltaxonomy`** en RaptorXML+XBRL Server, o el comando **`run`** en MapForce Server.

© 2016-2022 Altova GmbH

Funcionamiento

Altova mantiene un almacenamiento en línea donde guarda todas las taxonomías XBRL de los productos de Altova. Este almacenamiento se actualiza de forma periódica, por ejemplo, poco después de que las organizaciones correspondientes publiquen las versiones nuevas de las taxonomías respectivas. Al ejecutar Gestor de taxonomías desde la interfaz gráfica del usuario aparece información sobre las taxonomías más recientes disponibles en un cuadro de diálogo en el que puede visualizarlas, instalarlas, actualizarlas o desinstalarlas. También puede llevar a cabo las mismas acciones desde la línea de comandos.

También puede instalar las taxonomías ejecutando los archivos **.altova_taxonomies** descargados desde el sitio web de Altova (https://www.altova.com/taxonomy-manager). El sitio web prepara un archivo de tipo **.altova_taxonomies** que puede descargar y que contiene la información sobre las taxonomías seleccionadas. Al hacer doble clic en este archivo o pasarlo a Gestor de taxonomías desde la línea de comandos como argumento del comando **install** 829, Gestor de taxonomías instala las taxonomías que contiene.

Memoria caché local: seguimiento de taxonomías

Independientemente de cómo se instalen las taxonomías, toda la información sobre las taxonomías instaladas se almacena en una ubicación centralizada de su equipo, el directorio caché. El directorio caché local está en:

Windows	`C:\ProgramData\Altova\pkgs\.cache`
Linux	`/var/opt/Altova/pkgs\.cache`
macOS	`/var/Altova/pkgs`

El directorio caché local se actualiza automáticamente de vez en cuando para que el estado más actual del equipo corresponda con el del almacenamiento en línea. Más concretamente, el caché se actualiza:

- al ejecutar el Gestor de taxonomías.
- al ejecutar XMLSpy por primera vez en un mismo día natural.
- si XMLSpy ya se está ejecutando, el directorio caché se actualiza cada 24 horas.
- también puede actualizar el caché local desde el almacenamiento en línea manualmente ejecutando el comando de actualización **update** 832 desde la línea de comandos.

Si instala o desinstala taxonomías, el directorio caché local del Gestor de taxonomías se actualiza automáticamente con información sobre las taxonomías disponibles e instaladas, además de con los propios archivos de taxonomía.

No modifique la memoria caché manualmente

El directorio caché local se mantiene automáticamente en base a las taxonomías que instale o desinstale; no debe modificarlo ni eliminarlo manualmente. Si necesita restaurar el Gestor de taxonomías a su estado original, ejecute el comando reset 831 desde la línea de comandos y después ejecute el comando initialize 829. (También puede ejecutar el comando **reset** con la opción **--i**.)

17.1.1 Ejecutar el gestor de taxonomías

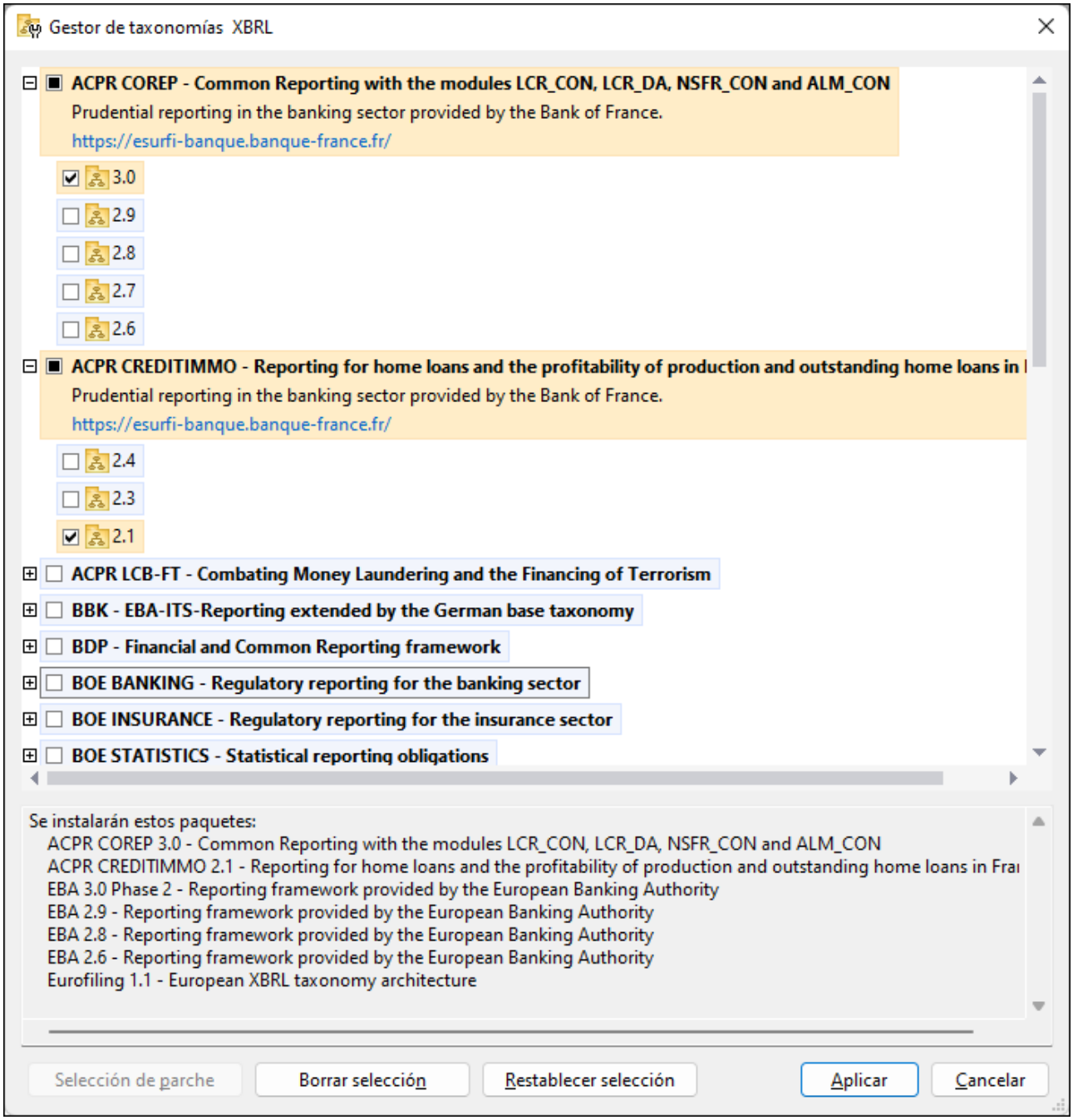
Interfaz gráfica del usuario

Hay varias formas de acceder a la IGU del Gestor de taxonomías:

- *Durante la instalación de XMLSpy:* al final del proceso de instalación, seleccione la casilla Invocar al Gestor de taxonomías para acceder directamente al gestor de taxonomías XBRL. Con él puede instalar taxonomías durante el proceso de instalación de su aplicación de Altova.
- *Después de la instalación de XMLSpy:* una vez haya instalado la aplicación puede acceder al Gestor de taxonomías en cualquier momento desde el comando de menú Herramientas | **Gestor de taxonomías XBRL**
- Mediante el archivo `.altova_taxonomies` que descargó del *Centro de descargas de taxonomías XBRL de Altova:* haga doble clic en el archivo para ejecutar Gestor de taxonomías, que instalará las taxonomías que haya seleccionado.

Cuando se abra la IGU del Gestor de taxonomías (*imagen siguiente*) podrá ver en ella las taxonomías que ya se han instalado. Si quiere instalar más solo tiene que seleccionarlas, y al contrario si quiere desinstalar alguna. Una vez haya terminado, puede aplicar los cambios. Las taxonomías que se vayan a instalar o desinstalar aparecerán resaltadas y un mensaje le avisará de los cambios que está a punto de hacer en la ventana *Mensajes*, en la parte inferior de la ventana de Gestor de taxonomías (*véase imagen*).

© 2016-2022 Altova GmbH



Interfaz de la línea de comandos

Para ejecutar el Gestor de taxonomías desde una interfaz de la línea de comandos debe usar su archivo ejecutable, `taxonomymanager.exe`.

Puede encontrar este archivo:

- *en Windows:* `C:\ProgramData\Altova\SharedBetweenVersions`
- *en Linux o macOS (solo para aplicaciones de servidor):* `%INSTALLDIR%/bin`, donde `%INSTALLDIR%` es el directorio de instalación del programa.

Puede usar cualquiera de los comandos de la referencia de la línea de comandos, a continuación.

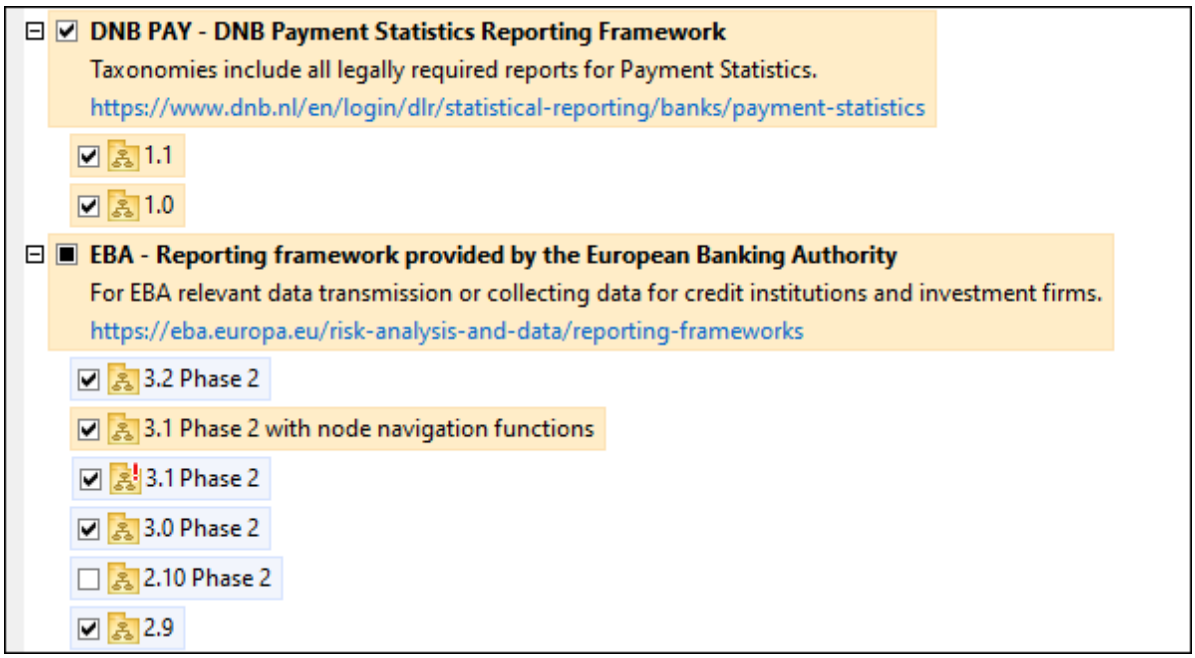
Para mostrar al ayuda de los comandos ejecute:

- *en Windows:* `taxonomymanager.exe --help`
- *en Linux o macOS (solo para aplicaciones de servidor):* `sudo ./taxonomymanager --help`

17.1.2 Categorías de estado

Gestor de taxonomías diferencia las taxonomías que administra entre:

- *Taxonomías instaladas:* estas aparecen en la IGU con sus casillas marcadas (*en la imagen siguiente las versiones marcadas de la taxonomía EBA son las que están instaladas*). Si se seleccionan todas las versiones de una taxonomía, en la casilla de la taxonomía aparece una marca de verificación. Si hay al menos una taxonomía sin seleccionar, en la en la casilla de la taxonomía aparece un cuadrado negro. Para **desinstalar** una taxonomía debe desmarcar la casilla correspondiente.
- *Taxonomías disponibles no instaladas:* Estas aparecen en la IGU con las casillas correspondientes sin seleccionar. Para **instalar** taxonomías, marque la casilla correspondiente.



- *Taxonomías que pueden actualizarse:* son las que han sido revisadas por sus emisores. Aparecen indicadas en la IGU con el icono  (*imagen anterior*). Puede aplicar **parches** a la taxonomía seleccionada con la revisión que esté disponible.

Puntos importantes

© 2016-2022 Altova GmbH

- En la imagen anterior se han marcado tanto las taxonomías DNB como las EBA. Las que tienen un fondo azul ya están instaladas. Las que tienen el fondo amarillo no están instaladas pero se han seleccionado para instalarlas. Observe que (i) la taxonomía EBA 2.10 Phase 2 no está instalada ni se ha seleccionado para instalarla, (ii) la taxonomía EBA 3.1 Phase 2 se ha instalado pero su emisor ha generado un parche que no se ha instalado aún.
- Al ejecutar el Gestor de taxonomías desde la línea de comandos puede usar el comando `list` 830 con distintas opciones para ver distintas categorías de taxonomías:

`taxonomymanager.exe list`	Muestra todas las taxonomías instaladas y disponibles; también indica qué taxonomías se pueden actualizar
`taxonomymanager.exe list -i`	Muestra solo las taxonomías instaladas; también indica qué taxonomías se pueden actualizar
`taxonomymanager.exe list -u`	Muestra qué taxonomías se pueden actualizar

Nota: en Linux y macOS **`use sudo ./taxonomymanager list`**

17.1.3 Aplicar parches o instalar una taxonomía

Aplicar un parche a una taxonomía instalada

A veces los emisores de las taxonomías XBRL generan parches. Cuando el Gestor de taxonomías XBRL detecta que hay parches disponibles, estos aparecen en las listas de taxonomías, desde donde puede instalarlos.

En la IGU

Los parches se indican con el icono . (*Consulte también el apartado anterior sobre categorías de estado* 824 .) Si hay parches disponibles se habilita el botón **Seleccionar parches**. Haga clic en él para seleccionar y preparar los parches. En la IGU, el icono de las taxonomías correspondientes cambia de  a  y el cuadro de diálogo le informa de qué parches se van a aplicar. Las listas del panel principal y las del panel Mensajes están ordenadas alfabéticamente, por lo que puede ver y revisar las taxonomías antes de aplicar los parches. Una vez esté listo para instalar los parches seleccionados, haga clic en **Aplicar**.

En la línea de comandos

Para aplicar un parche desde lal ínea de comandos:

1. Ejecute el comando `list -u` 830. Aparece una lista con las taxonomías para las que hay parches disponibles.
1. Ejecute el comando `upgrade` 833 para instalar todos los parches.

Instalar una taxonomía disponible

Para instalar taxonomías puede usar la IGU del Gestor de taxonomías o enviar las instrucciones al Gestor de taxonomías desde la línea de comandos.

Nota: si la taxonomía actual tiene dependencias en otras taxonomías, también se instalan (o desinstalan, según el caso) las taxonomías dependientes.

En la IGU
Para instalar taxonomías con la IGU del Gestor de taxonomías, seleccione las taxonomías que quiere instalar y haga clic en **Aplicar**.

También puede seleccionar las taxonomías que quiere instalar en el sitio web de Altova y generar desde allí un archivo `.altova_taxonomies`. Al hacer doble clic en este archivo se abre el Gestor de taxonomías con las taxonomías que indicó preseleccionadas. Solo tiene que hacer clic en **Aplicar**.

En la línea de comandos
Para instalar taxonomías desde la línea de comandos ejecute el comando `install`[829]:

```
taxonomymanager.exe install [opciones] FILTER+
```

donde `FILTER` es la taxonomía (o las taxonomías) que quiere instalar o un archivo `.altova_taxonomies`. Para hacer referencia a una taxonomía se usa un identificados con el formato `<nombre>-<versión>` que aparece junto a cada taxonomía que muestra el comando `list`[830]. Puede introducir tantas taxonomías como quiera. Para más detalles consulte la descripción del comando `install`[829] .

Nota: en Linux o macOS, use el comando `sudo ./taxonomymanager`.

Instalar una taxonomía requerida
Si ejecuta un comando en XMLSpy y XMLSpy descubre que una de las taxonomías que necesita para ejecutar el comando falta o está incompleta, el Gestor de taxonomías incluirá información sobre ese componente de taxonomía que falta. Entonces puede aplicar el parche indicado y/o instalar la taxonomía que falta.

Siempre puede ver todas las taxonomías instaladas previamente ejecutando el Gestor de taxonomías desde **Herramientas | Gestor de taxonomías**.

17.1.4 Desinstalar o restaurar taxonomías

Desinstalar una taxonomía

Para desinstalar taxonomías puede usar la IGU del Gestor de taxonomías o enviar las instrucciones al Gestor de taxonomías desde la línea de comandos.

Nota: si la taxonomía actual tiene dependencias en otras taxonomías, también se instalan (o desinstalan, según el caso) las taxonomías dependientes.

En la IGU
Para desinstalar taxonomías con la IGU del Gestor de taxonomías, seleccione las taxonomías que quiere desinstalar y haga clic en **Aplicar**. Las taxonomías seleccionadas y sus dependencias se desinstalarán.

Para desinstalar todas las taxonomías haga clic en **Deseleccionar todas** y haga clic en Aplicar.

© 2016-2022 Altova GmbH

En la línea de comandos
Para desinstalar taxonomías desde la línea de comandos ejecute el comando **uninstall**:

```
taxonomymanager.exe uninstall [options] FILTER+
```

donde **FILTER** es la taxonomía (o las taxonomías) que quiere desinstalar o un archivo **.altova_taxonomies**. Para hacer referencia a una taxonomía se usa un identificador con el formato **<nombre>-<versión>** que aparece junto a cada taxonomía que muestra el comando **list**[830]. Puede introducir tantas taxonomías como quiera. Para más detalles consulte la descripción del comando uninstall[831].

Nota: en Linux o macOS, use el comando **sudo ./taxonomymanager**.

Restaurar el Gestor de taxonomías

Puede restaurar el Gestor de taxonomías, es decir, eliminar todas las taxonomías instaladas, así como el directorio caché.

- En la IGU, haga clic en **Restaurar selección**.
- En la línea de comandos, use el comando reset[831].

Una vez haya ejecutado este comando, asegúrese de que ejecuta también el comando initialize[829] para recrear el directorio caché. También puede ejecutar el comando reset[831] con la opción -i.

Recuerde que reset -i[831] restaura la instalación original del producto, por lo que es recomendable ejecutar el comando update[832] después de restaurar el gestor. Puede ejecutar el comando reset[831] con las opciones -i o -u.

17.1.5 Interfaz de la línea de comandos (ILC)

Para llamar a Gestor de taxonomías desde la línea de comandos necesita saber la ruta del ejecutable. Por defecto, el ejecutable del Gestor de taxonomías se encuentra en:

```
C:\ProgramData\Altova\SharedBetweenVersions\TaxonomyManager.exe
```

Nota: en los sistemas Linux y macOS una vez haya cambiado el directorio al que contiene el ejecutable, puede llamar al ejecutable con **sudo ./taxonomymanager**.El prefijo **./** indica que el ejecutable está en el directorio actual. El prefijo **sudo** indica que el comando se debe ejecutar con derechos de administrador.

Sintaxis de la línea de comandos

La sintaxis general para usar la línea de comandos es:

```
<exec> -h | --help | --version | <command> [opciones] [argumentos]
```

En el código anterior la barra vertical **|** separa elementos que se excluyen mutuamente. Los corchetes **[]** indican elementos opcionales. Básicamente, puede teclear la ruta del ejecutable seguida por las opciones --h,

`--help` o `--version`, o por un comando. Cada comando puede tener opciones y argumentos. Los comandos se describen en los apartados siguientes.

17.1.5.1 help

Este comando ofrece ayuda contextual sobre los comandos del ejecutable del Gestor de taxonomías.

Sintaxis

```
<exec> help [command]
```

Donde `[command]` es un argumento opcional que indica cualquier nombre válido de comando.

Tenga en cuenta que:

- Puede invocar la ayuda tecleando un comando seguido por `--h` or `--help`, por ejemplo: **`<exec> list -h`**
- Puede invocar la ayuda general tecleando `--h` o `--help` directamente después del ejecutable, por ejemplo:

Ejemplo

Este comando muestra la ayuda del comando `list`:

```
taxonomymanager help list
```

17.1.5.2 info

Este comando muestra información detallada sobre cada una de las taxonomías dadas como argumento. Esta información incluye el título, la versión, la descripción, el editor y las referencias de las dependencias.

Sintaxis

```
<exec> info [opciones] Taxonomy+
```

- El argumento **`Taxonomy`** es el nombre de una taxonomía o parte del nombre de una taxonomía. (Para ver el ID de un paquete de taxonomías y la información relativa a su estado de instalación use el comando `list` 830.)
- Use **`<exec> info -h`** para ver la ayuda sobre este comando en la línea de comandos..

Ejemplo

Este comando muestra información detallada sobre las taxonomías **`eba-2.10`** y **`us-gaap-2020`**:

```
taxonomymanager info eba-2.1.0 us-gaap-2020.0
```


© 2016-2022 Altova GmbH

17.1.5.3 initialize

Este comando inicializa el entorno del Gestor de taxonomías y crea un directorio caché donde se guardan todas las taxonomías localmente. El Gestor de taxonomías se inicializa automáticamente la primera vez que instale una aplicación de Altova compatible con él, por lo que normalmente no es necesario ejecutar este comando. Por lo general solo es necesario ejecutarlo después de haber ejecutado el comando `reset`.

Sintaxis

```
<exec> initialize | init [opciones]
```

Opciones
Estas son las opciones del comando **`initialize`**:

`--help, --h`	Muestra la ayuda sobre este comando en la línea de comandos.
`--silent, --s`	Muestra solamente los mensajes de error. El valor predeterminado es **`false`**.
`--verbose, --v`	Muestra información suplementaria durante la ejecución. El valor predeterminado es **`false`**.

Ejemplo

Este comando inicializa el Gestor de taxonomías:

```
taxonomymanager initialize
```

17.1.5.4 install

Este comando instala una o más taxonomías.

Sintaxis

```
<exec> install [options] Taxonomy+
```

Para indicar varias taxonomías, repita `FILTER` tantas veces como sea necesario.

El argumento de **`Taxonomy`** puede ser:

1. Un identificador de taxonomía en el formato **`<name>-<version>`**, por ejemplo: **`eba-2.10`**). Para ver todos los identificadores de taxonomías y sus versiones ejecute el comando `list`[830]. También puede usar el nombre de la taxonomía abreviado, si este es único, por ejemplo **`eba`**. Si usa una abreviación del nombre se desinstalan todas las taxonomías que contengan esa abreviación.
2. La ruta de acceso a un archivo **`.altova_taxonomies`** descargado desde el sitio web de Altova. Para más información sobre estos archivos consulte la *Introducción al Gestor de taxonomías: funcionamiento*[818].

Opciones
Estas son las opciones del comando **install**:

--help, --h	Muestra la ayuda sobre este comando en la línea de comandos.
--silent, --s	Muestra solamente los mensajes de error. El valor predeterminado es **false**.
--verbose, --v	Muestra información suplementaria durante la ejecución. El valor predeterminado es **false**.

Ejemplo

Este comando instala las taxonomías **eba** (Autoridad Bancaria Europea) y **us-gaap** (principios de contabilidad generalmente aceptados en EE UU):

```
taxonomymanager install eba us-gaap
```

17.1.5.5 list

Use este comando para ver las taxonomías del Gestor de taxonomías; tiene varias opciones:

- lista de todas las taxonomías disponibles
- lista de taxonomías específicas
- lista de las taxonomías instaladas
- lista de las taxonomías que se pueden actualizar.

Sintaxis

```
<exec> list | ls [options] Taxonomy?
```

Si no se indica ningún argumento **Taxonomy** la lista incluye todas las taxonomías. De lo contrario la lista incluye las taxonomías indicadas en las opciones (*véase el ejemplo de más abajo*). Recuerde que puede usar el argumento **Taxonomy** tantas veces como quiera.

Opciones
Estas son las opciones del comando **list**:

--help, --h	Muestra la ayuda sobre este comando en la línea de comandos.
--installed, --i	Muestra solamente las taxonomías instaladas. El valor predeterminado es **false**.
--upgradeable, --u	Muestra solamente las taxonomías para las que hay disponible una versión más reciente (parches). El valor predeterminado es **false**.

Ejemplos

- Para ver todas las taxonomías disponibles ejecute: **taxonomymanager list**

© 2016-2022 Altova GmbH

- Para ver solamente las taxonomías instaladas ejecute: **taxonomymanager list -i**
- Para ver todas las taxonomías cuyos nombres contienen "eba" o "us-gaap" ejecute: **taxonomymanager list eba us-gaap**

17.1.5.6 reset

Este comando elimina todas las taxonomías instaladas, así como el directorio de caché. Este comando elimina todas las taxonomías instaladas y su información. Una vez haya ejecutado este comando, asegúrese de que ejecuta el comando initialize[829] para volver a crear el directorio de caché. También puede ejecutar el comando `reset` con la opción `-i`. Tenga en cuenta que `reset -i` restaura la instalación original del producto, por lo que se recomienda ejecutar también el comando update[832] después de una restauración. También puede ejecutar el comando `reset` con las opciones `-i` y `-u`.

Sintaxis

```
<exec> reset [opciones]
```

Opciones
Estas son las opciones del comando **reset**:

`--help, --h`	Muestra la ayuda sobre este comando en la línea de comandos.
`--init, --i`	Inicializa el entorno del Gestor de taxonomías después de una restauración. El valor predeterminado es **false**.
`--silent, --s`	Muestra solamente los mensajes de error. El valor predeterminado es **false**.
`--update, --u`	Inicializa y actualiza el entorno del Gestor de taxonomías XBRL después de una restauración. El valor predeterminado es **false**.
`--verbose, --v`	Muestra información suplementaria durante la ejecución. El valor predeterminado es **false**.

Ejemplos

- Para restaurar el Gestor de taxonomías, ejecute: **taxonomymanager reset**
- Para restaurar el Gestor de taxonomías e inicializarlo, ejecute: **taxonomymanager reset -i**
- Para restaurar el Gestor de taxonomías, inicializarlo y actualizar la lista de taxonomías, ejecute: **taxonomymanager reset -i -u**

17.1.5.7 uninstall

Este comando desinstala una o más taxonomías. Por defecto, cualquier taxonomía a la que haga referencia la taxonomía actual también se desinstala. Para desinstalar solamente la taxonomía actual y mantener aquellas a las que se hace referencia, use la opción `--k`.

Sintaxis

```
<exec> uninstall [opciones] Taxonomy+
```

Para indicar varias taxonomías, repita `FILTER` tantas veces como sea necesario.

El argumento de **`Taxonomy`** puede ser:

1. Un identificador de taxonomía en el formato **`<name>-<version>`**, por ejemplo: **`eba-2.10`**). Para ver todos los identificadores de taxonomías y sus versiones ejecute el comando `list -i`[830]. También puede usar el nombre de la taxonomía abreviado, si este es único, por ejemplo **`eba`**. Si usa una abreviación del nombre se desinstalan todas las taxonomías que contengan esa abreviación.
2. La ruta de acceso a un archivo **`.altova_taxonomies`** descargado desde el sitio web de Altova. Para más información sobre estos archivos consulte la *Introducción al Gestor de taxonomías: funcionamiento*[818]..

Opciones
Estas son las opciones del comando **`uninstall`**:

`--help, --h`	Muestra la ayuda sobre este comando en la línea de comandos.
`--keep-references, --k`	Si usa esta opción, las taxonomías referenciadas no se desinstalan. El valor predeterminado es **`false`**.
`--silent, --s`	Muestra solamente los mensajes de error. El valor predeterminado es **`false`**.
`--verbose, --v`	Muestra información suplementaria durante la ejecución. El valor predeterminado es **`false`**.

Ejemplo

Este comando desinstala las taxonomías **`eba-2.10`** y **`us-gaap-2020.0`** y sus dependencias:

```
taxonomymanager uninstall eba-2.10 us-gaap-2020.0
```

Este comando desinstala la taxonomía **`eba-2.10`** pero no las taxonomías a las que hace referencia:

```
taxonomymanager uninstall --k eba-2.10
```

17.1.5.8 update

Este comando consulta la lista de taxonomías disponibles en el almacenamiento en línea y actualiza el directorio de caché local. Esta información se actualiza de forma implícita, por lo que no es necesario ejecutar este comando a no ser que haya ejecutado `reset` e `initialize`[829].

Sintaxis

```
<exec> update [opciones]
```

Opciones
Estas son las opciones del comando `update`:

© 2016-2022 Altova GmbH

`--help, --h`	Muestra la ayuda sobre este comando en la línea de comandos.
`--silent, --s`	Muestra solamente los mensajes de error. El valor predeterminado es **`false`**.
`--verbose, --v`	Muestra información suplementaria durante la ejecución. El valor predeterminado es **`false`**.

Ejemplo

Este comando actualiza la lista de taxonomías en la memoria caché local:

```
taxonomymanager update
```

17.1.5.9 upgrade

Este comando actualiza todas las taxonomías aptas para la versión *parche* más reciente disponible. En otras palabras, solamente actualiza a nivel de los parches de un release específico. Se recomienda ejecutar este comando solo si hay taxonomías por actualizar. Puede identificar cuáles lo son con el comando `list -u`**.**

Nota: el comando `upgrade` eliminaría una taxonomía obsoleta si no hay ninguna versión disponible.

Sintaxis

```
<exec> upgrade [opciones]
```

Opciones

Estas son las opciones del comando `upgrade`:

`--help, --h`	Muestra la ayuda sobre este comando en la línea de comandos.
`--silent, --s`	Muestra solamente los mensajes de error. El valor predeterminado es **`false`**.
`--verbose, --v`	Muestra información suplementaria durante la ejecución. El valor predeterminado es **`false`**.

17.2 Procedimientos básicos

En esta sección encontrará la información necesaria para crear taxonomías con los componentes XBRL más básicos. Se divide en varios apartados:

- Empieza explicando las diferencias a la hora de trabajar con taxonomías nuevas y taxonomías ya disponibles[834] y lo importante que es comprender estas diferencias. Después se explica qué archivos forman parte de una taxonomía XBRL[835] y cómo se presentan en la vista **XBRL**.
- A partir del apartado Crear una taxonomía nueva[838] explicamos los pasos para crear una taxonomía en la vista **XBRL**. Al final de cada apartado también hay instrucciones para crear una taxonomía en la vista **XBRL**. El propósito de estas instrucciones es ayudarle a poner en práctica lo aprendido en el apartado y complementar la información de los apartados anteriores.

17.2.1 Taxonomías nuevas y existentes

En la vista **XBRL** de XMLSpy puede editar taxonomías ya existentes y crear taxonomías nuevas.

- *Taxonomías ya existentes.* Hay dos tipos de taxonomías ya existentes: (i) taxonomías estándar que no se deben editar y (ii) taxonomías no estándar que se pueden editar. Estas taxonomías no estándar las creó usted o algún otro usuario.
- *Taxonomías nuevas*. Puede crear taxonomías nuevas en XMLSpy. Hay dos tipos de taxonomías nuevas: (i) taxonomías nuevas creadas desde cero y (ii) taxonomías nuevas que amplían una taxonomía estándar.

Ambos tipos de taxonomías se pueden visualizar y editar en la vista **XBRL**. En algunos casos, como como cuando se importa una taxonomía estándar a una taxonomía que se está creando (para ampliar la taxonomía importada), no podrá editar la taxonomía importada. Los elementos de las taxonomías importadas que no se pueden editar aparecen en color gris.

Paquetes de taxonomías

Un paquete de taxonomías XBRL es un archivo comprimido que contiene una copia sin conexión de una taxonomía. Cada paquete de taxonomías contiene un archivo XML de catálogo que reasigna los URI a la ubicación de los archivos de la taxonomía sin conexión y, por tanto, pone la taxonomía a disposición de la aplicación sin necesidad de tener conexión a Internet. Las reglas que especifican cómo se debe estructurar y construir un paquete de taxonomías pueden encontrarse en la recomendación sobre paquetes de taxonomías de XBRL.org.

Si descarga un paquete de taxonomías, podrá registrarlo con XMLSpy para que la aplicación pueda usar los recursos del paquete sin conexión a la hora de validar datos. El registro del paquete de taxonomías se hace en la sección *Paquetes de taxonomías XBRL*[1621] del cuadro de diálogo "Opciones"[1621] y el procedimiento se describe en este apartado de la[1621] *Referencia del usuario*[1621].

Pasos para crear una taxonomía nueva

Por lo general las taxonomías nuevas se basan en una taxonomía ya existente. En la taxonomía nueva se añaden elementos nuevos y se crean relaciones entre estos elementos nuevos y entre los elementos nuevos y los elementos importados. A continuación enumeramos los requisitos que debe tener una taxonomía nueva y cómo puede crear una:

© 2016-2022 Altova GmbH

1. La taxonomía nueva debe crearse en su propio espacio de nombres para poder distinguirla de otras taxonomías. Si desea ampliar una taxonomía ya existente con la nueva taxonomía, importe la taxonomía ya existente en la taxonomía nueva.
2. Los conceptos (elementos) nuevos se definen en la taxonomía nueva.
3. Se crean archivos de relaciones (o bases de enlaces) que contienen las relaciones de definición, presentación, cálculo, etiqueta y referencia de la taxonomía nueva.
4. Las relaciones de la nueva taxonomía se deben crear desde cero.

En el párrafo anterior usamos el término *taxonomía* para referirnos a toda la taxonomía, que está formada por varios archivos: los archivos de definiciones de conceptos y los archivos de relaciones. (Consulte el apartado Introducción a los archivos de la taxonomía 835 para leer una descripción de los tipos de archivos que forman parte de una taxonomía.)

Trabajar con la vista XBRL

En los apartados siguientes explicamos cómo se usan las funciones de la vista **XBRL** a la hora de crear y editar taxonomías. Al final del apartado Crear una taxonomía nueva 838 encontrará instrucciones para crear su propia taxonomía. A partir de este apartado, todos los apartados incluyen instrucciones que complementan el trabajo realizado en el apartado inmediatamente anterior. Cuando llegue al final del apartado Crear relaciones: primera parte 853 ya estará familiarizado con la vista **XBRL** y podrá usarla con seguridad.

La taxonomía que creará será idéntica a la taxonomía de ejemplo que viene con XMLSpy (`Nanonull.xsd`), que está en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023Examples\XBRLExamples\Nanonull`. (Tenga en cuenta que el archivo de taxonomía principal siempre tiene la extensión `.xsd`. La extensión de archivo `.xbrl` se utiliza para archivos XBRL de instancia y no para archivos de taxonomía.)

17.2.2 Introducción a los archivos de la taxonomía

Una taxonomía XBRL bien diseñada almacena conceptos de taxonomía en un archivo separado de las relaciones de taxonomía. A este archivo lo llamamos archivo de taxonomía principal o archivo de definición de conceptos. Además, puesto que existen varios tipos de relaciones, las relaciones se almacenan en otros archivos, uno por cada tipo de relación. La tabla que aparece a continuación enumera los diferentes tipos de archivos que suelen formar parte de un documento de taxonomía.

Archivo XBRL	Descripción	Tipo de archivo
Conceptos	Cada concepto se define en un elemento `element` de XML Schema.	Archivo de XML Schema (`.xsd`) Archivo de definición de conceptos
Relaciones de definición	El elemento `definitionLink` contiene todos los localizadores y los arcos de definición para las relaciones entre los conceptos.	Archivo XML (`.xml`)
Relaciones de cálculo	El elemento `calculationLink` contiene todos los localizadores y los arcos de cálculo.	Archivo XML (`.xml`)

Relaciones de presentación	El elemento `calculationLink` contiene todos los localizadores y los arcos de cálculo.	Archivo XML (`.xml`)
Etiquetas	El elemento `labelLink` contiene todos los localizadores, arcos de etiqueta y etiquetas.	Archivo XML (`.xml`)
Referencias	El elemento `referenceLink` contiene todos los localizadores, arcos de referencia y recursos de referencia.	Archivo XML (`.xml`)

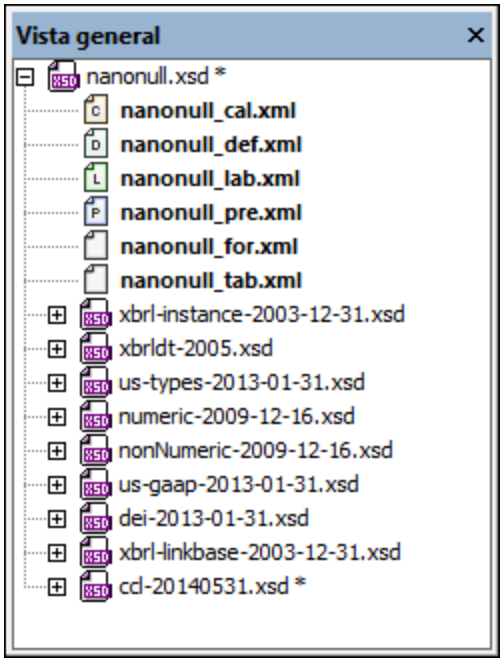
La ubicación de los archivos de relaciones se indica en el archivo de definición de conceptos (el archivo `.xsd`) dentro de un elemento `/schema/annotation/appinfo`:

```
<xsd:annotation>
   <xsd:appinfo>
      <link:linkbaseRef xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"
         xlink:href="NanonullLabels.xml" xlink:type="simple"
         xlink:role="http://www.xbrl.org/2003/role/labelLinkbaseRef" />
      <link:linkbaseRef xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"
         xlink:href="NanonullDefinitions.xml" xlink:type="simple"
         xlink:role="http://www.xbrl.org/2003/role/definitionLinkbaseRef" />
      <link:linkbaseRef xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"
         xlink:href="NanonullPresentations.xml" xlink:type="simple"
         xlink:role="http://www.xbrl.org/2003/role/presentationLinkbaseRef" />
      <link:linkbaseRef xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"
         xlink:href="NanonullCalculations.xml" xlink:type="simple"
         xlink:role="http://www.xbrl.org/2003/role/calculationLinkbaseRef" />
      <link:linkbaseRef xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"
         xlink:href="NanonullReferences.xml" xlink:type="simple"
         xlink:role="http://www.xbrl.org/2003/role/referenceLinkbaseRef" />

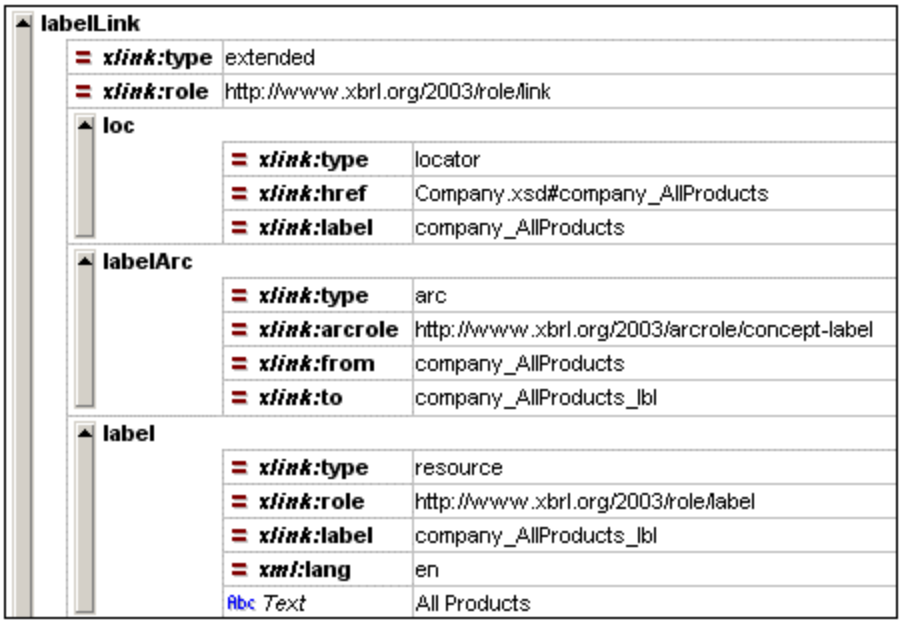
   </xsd:appinfo>
</xsd:annotation>
```

Cuando se abre el archivo de definición de conceptos (el archivo `.xsd`) en la vista XBRL, los archivos de la taxonomía aparecen en forma de árbol en el ayudante de entrada Vista general[272] (*imagen siguiente*).

© 2016-2022 Altova GmbH



Observe que a la izquierda de cada archivo aparece un icono. Los archivos de esquema XML (`.xsd`) llevan el icono XSD. Los iconos de los archivos de relaciones son de colores y llevan una letra que se corresponde con la inicial del tipo de relación: Por ejemplo, el icono indica que el archivo es de relaciones de definición, mientras que el icono indica que es un archivo de relaciones de presentación. Al hacer doble clic en el nombre de los archivos, el archivo se abre en XMLSpy. Después puede editar el archivo en la vista Cuadrícula (*imagen siguiente*) o en la vista Texto.



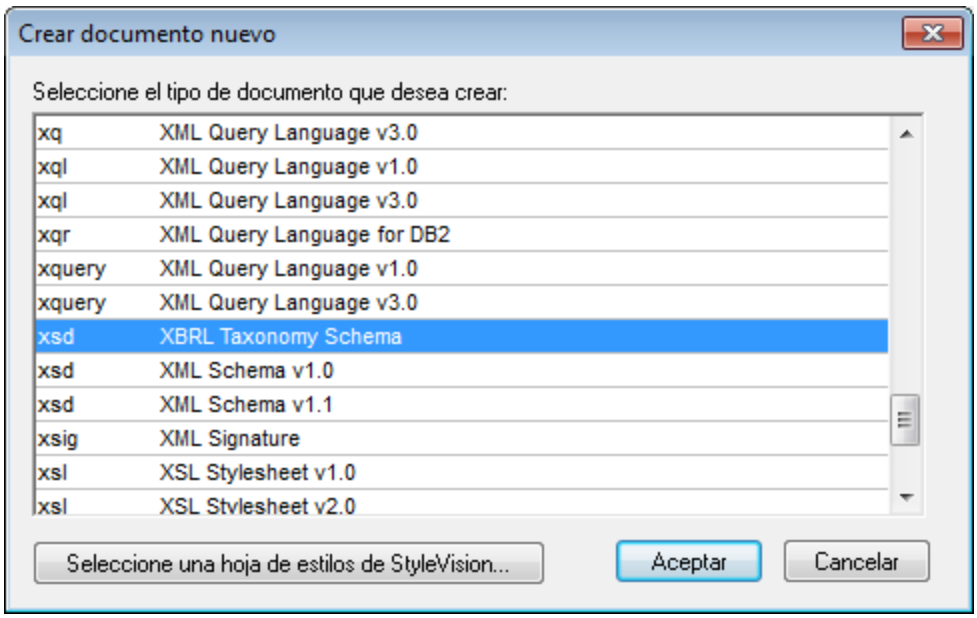
17.2.3 Crear una taxonomía nueva

Normalmente se crean taxonomías nuevas para ampliar una o varias taxonomías estándar. Si una taxonomía nueva amplía una taxonomía estándar o una taxonomía ya existente, es necesario importar estas últimas en la taxonomía nueva. Otra opción es crear una taxonomía nueva desde cero. En la vista XBRL de XMLSpy puede importar las taxonomías US-GAAP y NIIF (IFRS) con el asistente para la creación de taxonomías. La taxonomía se puede modificar más adelante en la interfaz gráfica de la vista XBRL.

El primer paso para crear una taxonomía nueva es crear su archivo de definición de conceptos, que es un archivo de esquema XML (.xsd). Además de los conceptos, este archivo define y declara el espacio de nombres de la taxonomía nueva, localiza las taxonomías que se deben importar y los archivos de relaciones de la taxonomía y declara el espacio de nombres de las taxonomías importadas, así como otros espacios de nombres utilizados en ellas.

Crear el archivo de definición de conceptos

Para crear una taxonomía XBRL nueva, seleccione el comando de menú **Archivo | Nuevo**. Aparece el cuadro de diálogo "Crear documento nuevo" (*imagen siguiente*).



Seleccione *xsd: XBRL Taxonomy Schema* y haga clic en **Aceptar**. Se crea una taxonomía nueva.

Recomendamos que guarde la taxonomía en su propio directorio para que también pueda guardar los componentes relacionados en una misma carpeta.

Vista general de los pasos para crear una taxonomía

Estos son a grandes rasgos los pasos para crear una taxonomía:

1. Seleccione una taxonomía base para su taxonomía con el comando **XBRL | Importar o hacer referencia** 839. Si quiere generar la taxonomía desce cero, omita este paso.

© 2016-2022 Altova GmbH

2. Asigne a la taxonomía un espacio de nombres de destino con el comando de menú **XBRL | Establecer espacio de nombres de destino** 843.
3. Los espacios de nombres de la taxonomía base que importó se declaran automáticamente en la taxonomía. También puede agregar más espacios de nombres si quiere, como se describe en Configurar los archivos de la taxonomía 845.
4. Entonces puede ampliar la taxonomía base con los elementos 847 y las relaciones 851 que quiera.

Archivo de ejemplo: paso 1

Cree una taxonomía nueva y guárdela. Este archivo es el archivo de taxonomía principal (o archivo de definición de conceptos). Es un archivo de esquema XML y, por tanto, debe tener la extensión de archivo `.xsd`. En la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023Examples\XBRLExamples\Nanonull` hay un archivo de taxonomía con ese mismo nombre.

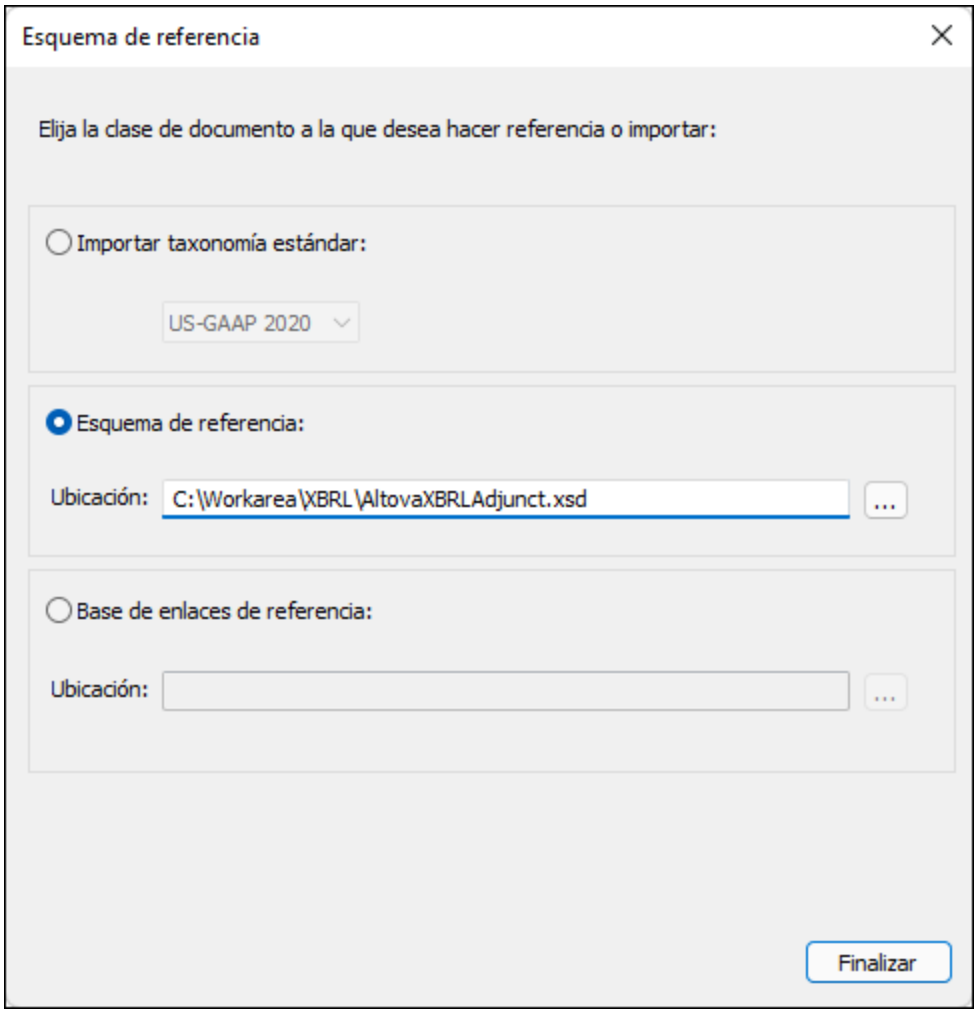
En el paso siguiente vamos a importar una taxonomía base para crear la nuestra.

17.2.4 Importar una taxonomía

Si el objetivo de la taxonomía nueva es ampliar una taxonomía existente, esta se debe importar a la taxonomía nueva. Si la taxonomía nueva se crea con el asistente para la creación de taxonomía, puede importar taxonomías US-GAAP y NIIF (IFRS) a la vez que se crea la taxonomía. También puede importar la taxonomía más tarde.

Para importar una taxonomía siga estos pasos:

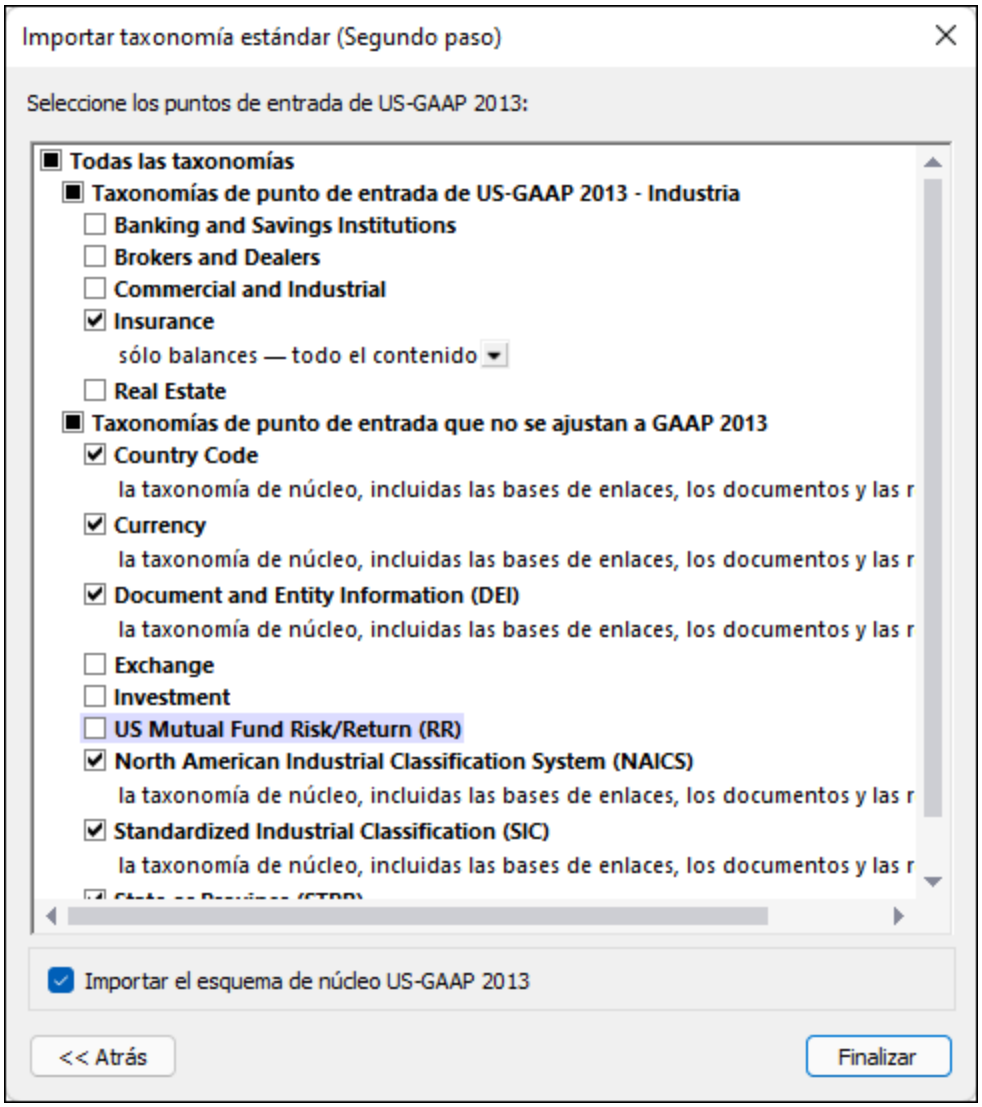
1. Haga clic con el botón derecho en el ayudante de entrada Vista general de la vista **XBRL** y seleccione el comando de menú **XBRL | Importar o hacer referencia** 1521.
2. Aparece el cuadro de diálogo "Importar taxonomía estándar" (*imagen siguiente*). Seleccione la taxonomía que desea importar o la base de enlaces que desea usar como referencia (dependiendo de la opción seleccionada el diálogo tiene un nombre u otro).



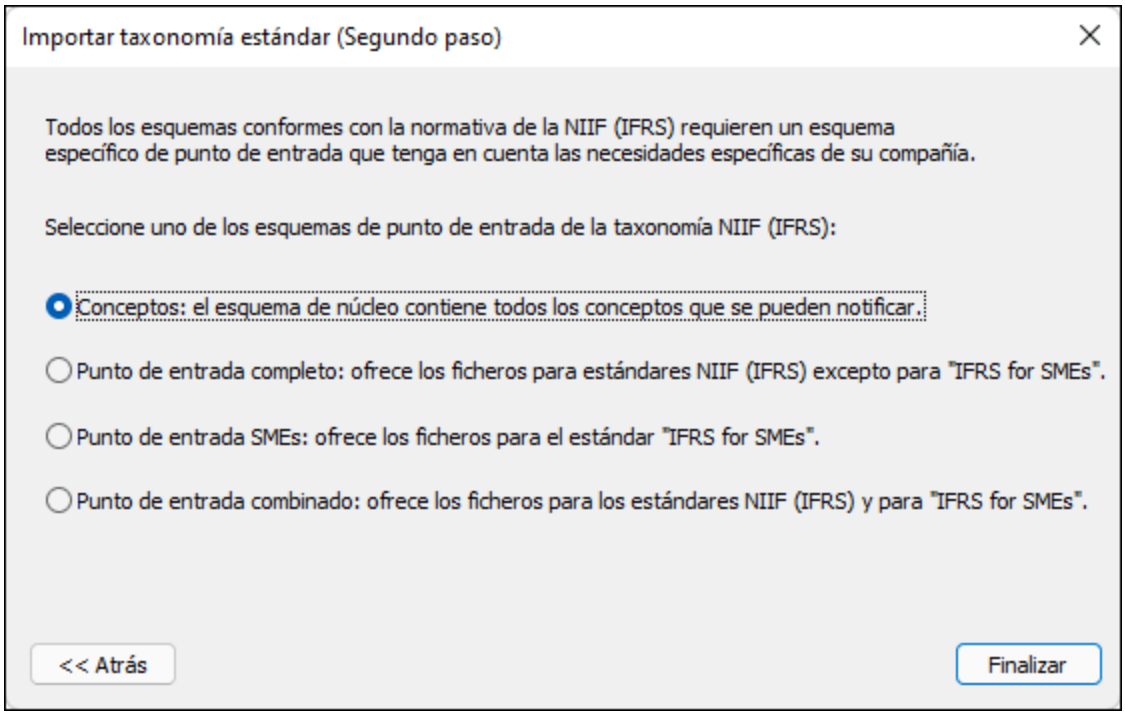
Este cuadro de diálogo ofrece tres opciones de importación: (i) una taxonomía estándar (US-GAAP o NIIF), (ii) otra taxonomía (esquema de referencia) y (iii) una base de enlaces. Si quiere importar una taxonomía no estándar, seleccione el botón de opción *Esquema de referencia* y busque la ubicación del esquema.

3. Para terminar haga clic en **Finalizar**. La taxonomía seleccionada se importa y sus elementos y relaciones aparecen en la vista **XBRL**.
4. Si seleccionó una taxonomía US-GAAP aparecerá una pantalla nueva en la que puede (i) marcar las casillas de los puntos que quiera incluir en la taxonomía e (ii) indicar si quiere importar el US-GAAP Core Schema (en la casilla que hay en la parte inferior del cuadro de diálogo).

© 2016-2022 Altova GmbH



Si seleccionó IFRS como base para la taxonomía puede seleccionar un punto de entrada específico para IFRS.



5. Haga clic en **Finalizar** para importar los esquemas de punto de entrada seleccionados, de forma que la taxonomía pueda hacer referencia a ellos. La taxonomía se abre en la vista XBRL, donde la puede editar.

Es necesario tener en cuenta algunos aspectos:

- El ayudante de entrada Vista general enumera las taxonomías que la taxonomía importada importa a su vez, así como las bases de enlaces que usa la taxonomía importada.
- El ayudante de entrada Elementos globales enumera los conceptos definidos en la taxonomía importada.
- La ventana Diseño y el ayudante de entrada Detalles muestra los conceptos importados en color gris.
- Para eliminar una taxonomía, haga clic con el botón derecho en la taxonomía en el ayudante de entrada Vista general. Seleccione el comando **Quitar**.

Nota: si nota que la edición se vuelve más lenta al trabajar con una taxonomía grande como US-GAAP, utilice el filtro de la ventana principal 265 para ver sólo los elementos creados en la taxonomía nueva.

Mecanismo de importación

Al importar una taxonomía estándar tal y como se explica en este apartado, se añade un elemento `xs:import` al archivo de taxonomía nueva. El elemento `xs:import` indica el espacio de nombres y la ubicación de la taxonomía importada (*ver código de ejemplo*).

```
<xs:import namespace="http://fasb.org/us-gaap/2011-01-31"
           schemaLocation="http://xbrl.fasb.org/us-gaap/2011/elts/us-gaap-2011-01-31.xsd"/>
```

En este fragmento de código, el atributo `schemaLocation` especifica que la taxonomía se debe cargar desde Internet. Pero este URI, mediante el mecanismo de catalogación de XMLSpy 456, equivale a una copia local de la taxonomía US-GAAP (que viene con XMLSpy).

© 2016-2022 Altova GmbH

Para localizar una taxonomía guardada en el disco local puede usar una dirección local. Otra opción es usar una dirección web asignada a una dirección local mediante un archivo de catálogo[456]. Si accede a las taxonomías desde ubicaciones locales podrá trabajar mucho más rápido.

Archivo de ejemplo: paso 2

En el ejemplo que nos ocupa usamos la taxonomía US-GAAP 2011. Esta taxonomía ya se importó cuando creamos la taxonomía inicial[838]. Observe con atención todas las taxonomías importadas y bases de enlaces referenciadas que aparecen en el ayudante de entrada Vista general. Cambie a la vista **Texto** y busque los elementos `xs:import`. Vuelva a la vista **XBRL** y observe que en la ventana principal de la vista **XBRL** los conceptos importados aparecen en color gris. Además el ayudante de entrada Vista general enumera las bases de enlaces y los esquemas importados de la taxonomía US-GAAP.

En el paso siguiente[845] nos centraremos en los archivos de base de enlaces y en el mecanismo de referencia.

17.2.5 Espacios de nombres de la taxonomía

El espacio de nombres de destino

El espacio de nombres de destino de una taxonomía se define en el atributo `xs:targetNamespace` del elemento `xs:schema` de la taxonomía (*ver el ejemplo de código que aparece a continuación*). (El elemento `xs:schema` es el elemento de documento del archivo de definición de conceptos.)

```
<xs:schema targetNamespace="http://www.altova.com/XBRL/Taxonomies">
        ...
</xs:schema>
```

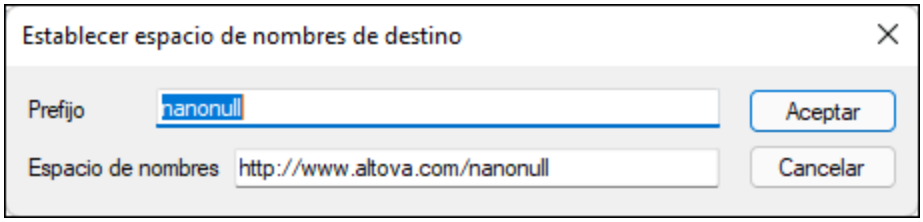
Además de definir el espacio de nombres de destino (es decir, de especificarlo), el espacio de nombres de destino debe **declararse** en el elemento `xs:schema` para que esté en el ámbito de aplicación en todo el documento. El fragmento de código que aparece a continuación declara el espacio de nombres como espacio de nombres de destino.

```
<xs:schema targetNamespace="http://www.altova.com/XBRL/Taxonomies"
xmlns:ns1="http://www.altova.com/XBRL/Taxonomies" >
        ...
</xs:schema>
```

En este fragmento de código el espacio de nombres se declara en el elemento `xs:schema` y se le da el prefijo `ns1`.

Establecer el espacio de nombres de destino

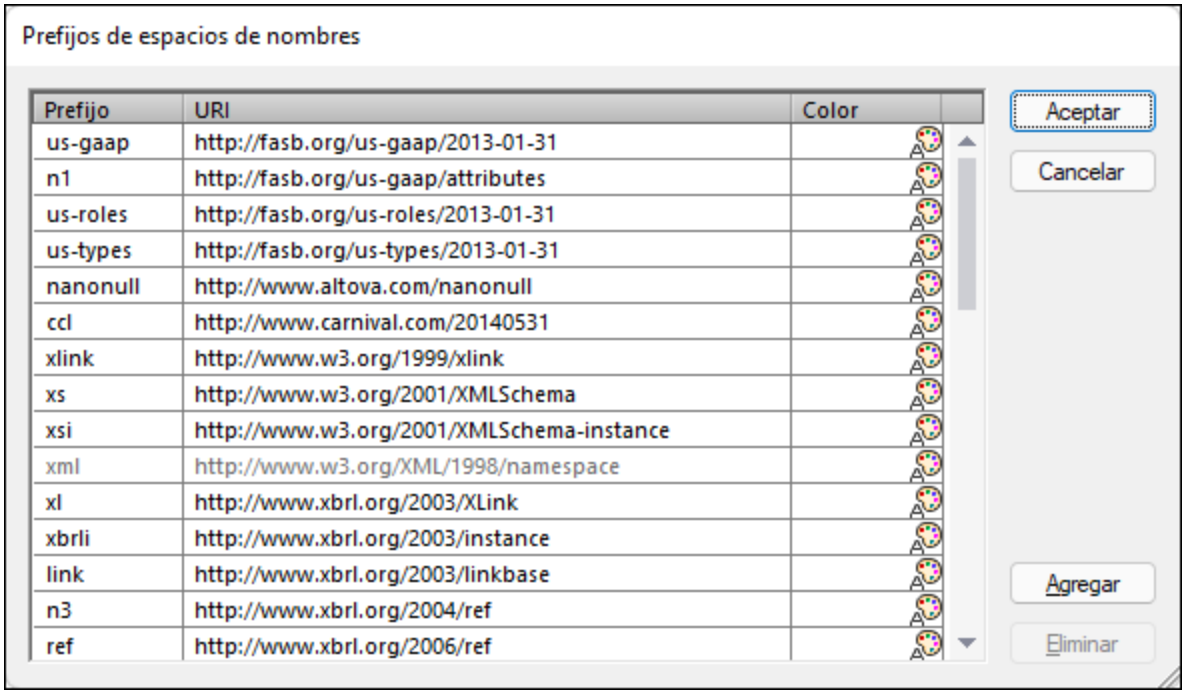
Cuando una taxonomía se crea con el asistente para la creación de taxonomías, el asistente crea un espacio de nombres y un prefijo predeterminados para la taxonomía. El espacio de nombres de destino predeterminado está basado en los datos introducidos en la primera pantalla del asistente. El prefijo del espacio de nombres de destino predeterminado será del tipo `nX`, siendo `X` un número entero. La declaración del espacio de nombres y el prefijo predeterminados se puede editar en el cuadro de diálogo "Establecer espacio de nombres de destino" (**XBRL | Establecer espacio de nombres de destino**) (*ver imagen siguiente*). Los cambios introducidos en este cuadro de diálogo no sólo modifican la **definición** del espacio de nombres de destino (el valor del atributo `targetNamespace`), también modifica la **declaración** del espacio de nombres de destino.



Si sólo desea modificar la declaración del espacio de nombres (y no su definición) o de cualquier otro espacio de nombres, edite el prefijo y el valor del espacio de nombres en el cuadro de diálogo "Prefijos de espacios de nombres" (**XBRL | Prefijos de espacios de nombres**).

Espacios de nombres de las taxonomías

Puede administrar los espacios de nombres de la taxonomía en el cuadro de diálogo "Prefijos de espacios de nombres" (*imagen siguiente*). Para abrir este cuadro de diálogo seleccione el comando de menú **XBRL | Prefijos de espacios de nombres**. En este cuadro de diálogo puede declarar espacios de nombres y asociar prefijos y colores de fondo a cada espacio de nombres. Los cambios realizados en este cuadro de diálogo modifican las declaraciones de espacios de nombres existentes en la taxonomía.



El cuadro de diálogo "Prefijos de espacios de nombres" enumera todos los espacios de nombres de la taxonomía.

- Para agregar o eliminar un espacio de nombres, use los botones **Agregar** o **Eliminar** respectivamente. Después de agregar un espacio de nombres, edite el prefijo y el URI predeterminados. Para ello haga doble clic en estos campos y escriba el prefijo y el URI nuevo.
- Puede asignar un color a cada espacio de nombres con la paleta de colores. Si un espacio de nombres tiene asignado un color, todos los componentes de ese espacio de nombres tendrán ese color de fondo en la ventana principal y en los ayudantes de entrada de la vista **XBRL**. Tenga en

© 2016-2022 Altova GmbH

cuenta que el color se le aplica al espacio de nombres en todos los documentos de taxonomía que estén abiertos en la vista **XBRL**.

Cuando termine de realizar cambios en el cuadro de diálogo "Prefijos de espacios de nombres" haga clic en **Aceptar**.

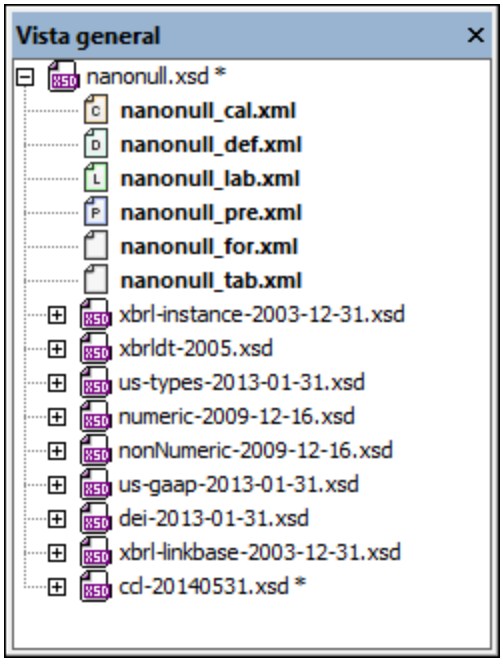
Archivo de ejemplo: paso 3

Abra el cuadro de diálogo "Establecer espacio de nombres de destino" (**XBRL | Establecer espacio de nombres de destino**). El espacio de nombres de destino es `http://xbrl.nanonull.com/2011` y su prefijo es `n1`. Haga doble clic en el espacio de nombres y escriba el espacio de nombres nuevo. Haga lo mismo con el prefijo. Para este ejemplo vamos a usar `http://www.altova.com/nanonull` y el prefijo `nanonull` (*ver imagen anterior*). Al hacer clic en el botón **Aceptar**, se asigna el espacio de nombres de destino y este se declara con el prefijo que se asignó en el cuadro de diálogo. Recuerde que en nuestro ejemplo el espacio de nombres de destino es `http://www.altova.com/nanonull` y el prefijo `nanonull`.

En el siguiente paso nos centraremos en el mecanismo de importación[839] de la vista **XBRL**.

17.2.6 Configurar los archivos de la taxonomía

El ayudante de entrada Vista general enumera, en forma de árbol, los archivos que constituyen la taxonomía (*imagen siguiente*). El componente raíz del árbol es el archivo de taxonomía principal (el archivo de definición de conceptos), que además es el archivo activo en este momento. Los archivos del siguiente nivel son de dos tipos: (i) archivos de base de enlaces que indican las relaciones presentes en la taxonomía (estos archivos aparecen codificados con iconos de colores[272]) y (ii) esquemas importados (los archivos `.xsd`).



En el apartado Importar una taxonomía 839 aprendimos a importar una taxonomía desde el ayudante de entrada Vista general. La taxonomía importada aparece junto a los esquemas importados del ayudante de entrada Vista general.

En este apartado explicamos cómo usar el ayudante de entrada Vista general para gestionar los archivos de bases de enlaces. El menú contextual de la Vista general incluye cuatro comandos para gestionar las bases de enlaces. Las operaciones que puede realizar con estos comandos son:

- Agregar bases de enlaces nuevas 846 y guardarlas en la taxonomía 846.
- Establecer el tipo de base de enlaces 846. Cuando XMLSpy no reconoce el tipo de base de enlaces (cálculo, definición, presentación, etiqueta o referencia), el usuario puede especificar el tipo de base de enlaces.
- Establecer una base de enlaces como base de enlaces predeterminada 847 para ese tipo de relación. Por ejemplo, si hay varias bases de enlaces de etiqueta, las etiquetas nuevas creadas en el editor de taxonomías se crearán en la base de enlaces de etiqueta predeterminada.
- Eliminar bases de enlaces 847.

Nota: recuerde que existen cinco tipos de relaciones: (i) definición, (ii) cálculo, (iii) presentación, (iv) etiqueta y (v) referencia. Puede crear archivos de bases de enlaces diferentes por cada tipo de relación.

Agregar una base de enlaces nueva

Siga estos pasos para crear una base de enlaces nueva:

1. En el ayudante de entrada Vista general haga clic con el botón derecho y seleccione el comando **Agregar base de enlaces nueva | <tipo de relación>**. En la Vista general se crea un archivo de base de enlaces del tipo seleccionado con un nombre predeterminado. La base de enlaces nueva se convierte en la base de enlaces predeterminada para ese tipo de relación (el nombre del archivo está en negrita).
2. Haga clic con el botón derecho en el nombre de archivo, seleccione **Cambiar de nombre** y escriba el nombre nuevo.
3. El archivo de base de enlaces nuevo propiamente dicho no se guarda en la ubicación correspondiente hasta que guarde el archivo de taxonomía principal. Consulte el resto de este apartado para obtener más información.

Guardar archivos de bases de enlaces

Si un archivo de base de enlaces está sin guardar, aparece un asterisco después del nombre del archivo de base de enlaces. Cuando se guarda el archivo de taxonomía principal, ocurre esto:

1. Aparece el cuadro de diálogo "Confirmar rutas de acceso de la base de enlaces". Este diálogo enumera el nombre y la ubicación (ruta de acceso) de todas las bases de enlaces de la taxonomía, incluidos los archivos de base de enlaces recién creados. Los archivos de base de enlaces que estén sin guardar tienen la misma ruta de acceso que el archivo de taxonomía principal. Puede editar la ruta de acceso de cada archivo si desea guardar alguno en otra ubicación. También puede editar el nombre de los archivos.
2. Cuando termine de realizar cambios en este diálogo, haga clic en **Aceptar**. Los archivos de bases de enlaces se guardan en las rutas de acceso indicadas.

Establecer una clase de base de enlaces

La clase de base de enlaces de un archivo (o tipo de base de enlaces) se puede configurar con este comando. Haga clic con el botón derecho en el archivo para cambiar su clase de base de enlaces y seleccione el

© 2016-2022 Altova GmbH

comando **Establecer clase de base de enlaces | <tipo de relación>**. La opción **Todos** indica que el archivo de base de enlaces puede contener más de un tipo de relación.

Establecer una base de enlaces predeterminada

Puede establecer cualquier archivo de base de enlaces como base de enlaces predeterminada para cada tipo de relación. Cuando se define un tipo de relación en el editor de taxonomías, la relación se guarda en el archivo de base de enlaces predeterminado para ese tipo de relación. Para establecer un archivo como base de enlaces predeterminada, haga clic con el botón derecho en el archivo y seleccione **Establecer base de enlaces predeterminada**. Las bases de enlaces predeterminadas aparecen en negrita en la Vista general.

Eliminar una base de enlaces

Para quitar una base de enlaces de la taxonomía, haga clic con el botón derecho en la base de enlaces y seleccione **Quitar**.

Archivo de ejemplo: paso 4

La taxonomía que estamos creando ya tiene archivos de bases de enlaces. Estos archivos se crearon cuando creamos la taxonomía con el asistente para la creación de taxonomías 838. Ahora cambiamos los nombres predeterminados de estos archivos por estos nombres nuevos:

- Base de enlaces de cálculo: `nanonull_cal.xml`
- Base de enlaces de definición: `nanonull_def.xml`
- Base de enlaces de etiqueta: `nanonull_lab.xml`
- Base de enlaces de presentación: `nanonull_pre.xml`

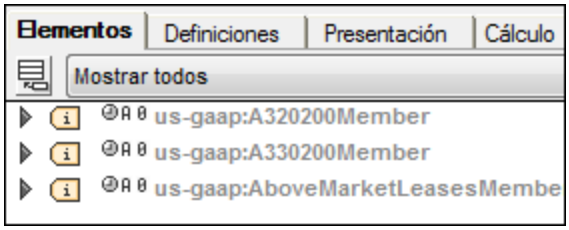
Si no quiere añadir los archivos de bases de enlaces, puede trabajar con la taxonomía `Nanonull.xsd` (que se ubica en `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\XBRLExamples\Nanonull`), que ya tiene archivos de bases de enlaces.

Realice estas operaciones para aprender a usar los comandos descritos en este apartado: cree un archivo de base de enlaces con el comando **Agregar base de enlaces nueva** y seleccione la clase de relación que prefiera. Cambie el nombre del archivo siguiendo las instrucciones dadas anteriormente. Observe que la base de enlaces recién creada pasa a ser la base de enlaces predeterminada para ese tipo de relación (el nombre del archivo aparece en negrita). Seleccione el archivo y cambie su clase de relación (con el comando **Establecer clase de base de enlaces**). Observe que el archivo **no es** la base de enlaces predeterminada de este otro tipo de relación. Ahora elimine la base de enlaces (con el comando **Quitar**). Ya que uno de los archivos de base de enlaces iniciales dejó de ser la base de enlaces predeterminada, configure un archivo de ese tipo de relación como base de enlaces predeterminada de su tipo de relación.

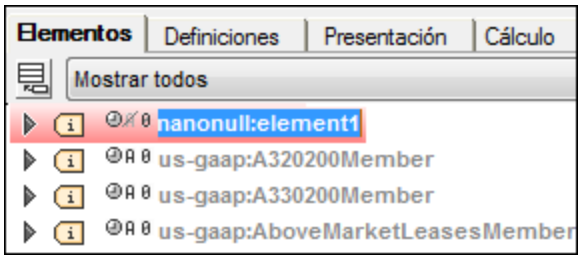
En el paso siguiente aprenderemos a agregar elementos nuevos al archivo de taxonomía principal 847 (o archivo de definición de conceptos).

17.2.7 Agregar elementos a una taxonomía

Para agregar un elemento (o concepto) a la taxonomía, haga clic en el icono **Agregar elemento nuevo** situado en la parte superior izquierda de la ventana principal (*imagen siguiente*).

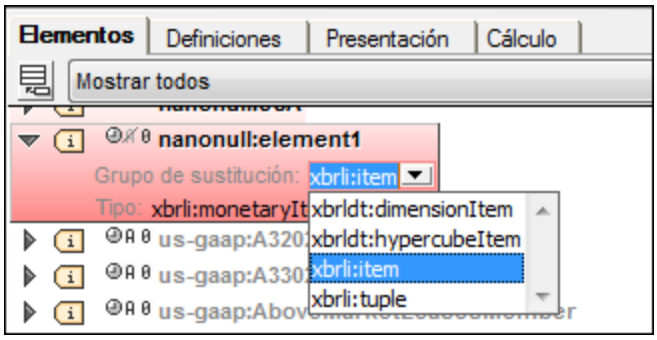


Se añade el elemento nuevo con un grupo de sustitución `xbrli:item` y con un nombre predeterminado a la lista de elementos (*imagen siguiente*).



Para más información sobre el recuadro del elemento consulte el apartado Ventana principal: pestaña Elementos[265]. Hay varias maneras de editar las propiedades del elemento en la ventana principal[252].

- Para cambiar el nombre del elemento, haga doble clic en él y escriba el nombre nuevo. Recuerde que además del nombre debe introducir el prefijo de espacio de nombres[843] correcto.
- Para cambiar el grupo de sustitución del elemento expanda el recuadro del elemento (haciendo clic en la flecha situada a la izquierda del elemento) y seleccione una opción del cuadro combinado `Grupo de sustitución` (*imagen siguiente*).



- Para cambiar las propiedades `Balance`, `Period`, `Abstract` y `Nillable` haga clic en el icono pertinente situado a la izquierda del elemento y seleccione una opción del cuadro que aparece.
- Para agregar una función de vínculo de etiqueta para el elemento, haga clic con el botón derecho dentro del recuadro del elemento y seleccione el comando **Agregar función de vínculo de etiqueta**. Se añade una fila para las etiquetas. En esta fila puede introducir una función de vínculo de etiqueta o seleccionar una en el cuadro combinado. Si la taxonomía no tiene asociado ningún archivo de base de enlaces de etiqueta, se crea uno y aparece en el ayudante de entrada Vista general[272].
- Para añadir una etiqueta a la función de vínculo de etiqueta, haga clic con el botón derecho en la función de vínculo de etiqueta y seleccione el comando **Agregar etiqueta**. Para editar la etiqueta, haga doble clic en el campo pertinente y escriba un valor para el campo o seleccione el valor en los

© 2016-2022 Altova GmbH

cuadros combinados. Los cambios realizados en las etiquetas se guardan en la base de enlaces de etiqueta cuando se guarda el archivo de taxonomía principal.
- El proceso para agregar referencias a la base de enlaces de referencia es el mismo que para agregar etiquetas a la base de enlaces de etiqueta. Primero se añade una función de vínculo de referencia al elemento y después se añade una referencia para la función de vínculo.

Las propiedades del elemento también se pueden editar en el ayudante de entrada Detalles. Para más información, consulte el apartado Ayudantes de entrada de la vista XBRL 272.

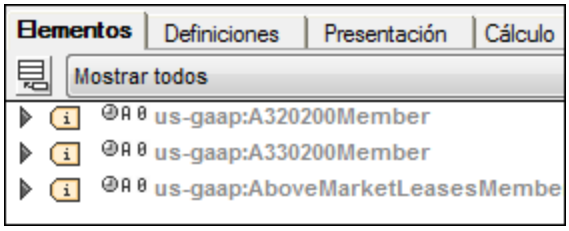
Archivo de ejemplo: paso 5

Ahora vamos a ampliar la taxonomía US-GAAP creando elementos nuevos.

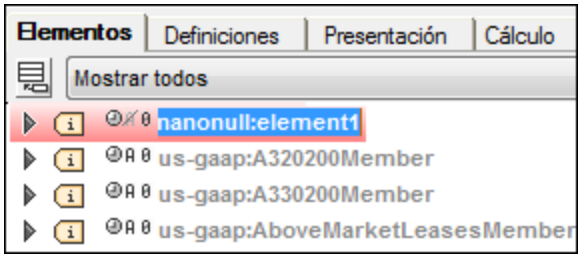
El primer elemento que vamos a crear es el ítem `nanonull:OnboardAndOther`, que representa los ingresos por la venta de productos a bordo de los cruceros Nanonull. Este tipo de ingresos no está disponible en la taxonomía US-GAAP. Por eso debemos crearlo como extensión de US-GAAP. Al ser un elemento nuevo creado especialmente para la taxonomía Nanonull, debemos crear este elemento en el espacio de nombres de Nanonull (`http://www.altova.com/nanonull`), que se declaró con el prefijo `nanonull`. Al crear el elemento con este prefijo, el elemento se coloca en el espacio de nombres Nanonull.

Siga estos pasos para crear el elemento:

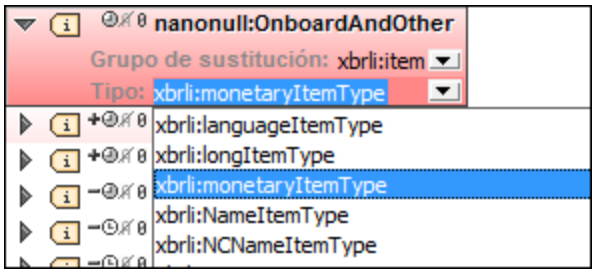
1. Haga clic en el icono **Agregar elemento nuevo** de la ventana principal (*imagen siguiente*).



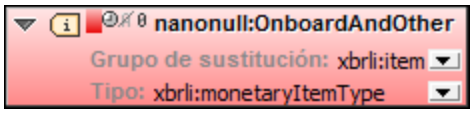
   Se añade un elemento nuevo con un grupo de sustitución `xbrli:item` y un nombre predeterminado a la lista de elementos (*imagen siguiente*).



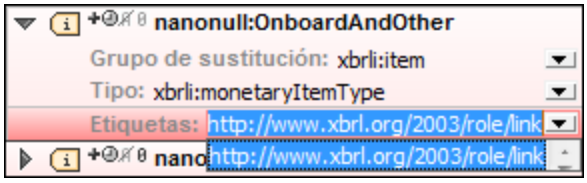
2. Haga doble clic en el nombre del elemento y escriba `nanonull:OnboardAndOther` (*imagen siguiente*). Como resultado se crea el elemento `OnboardAndOther` en el espacio de nombres Nanonull.
3. Expanda el recuadro del elemento y cambie el atributo de tipo `Type` por `xbrli:monetaryItemType` (*imagen siguiente*) ya que el elemento tendrá un valor monetario.



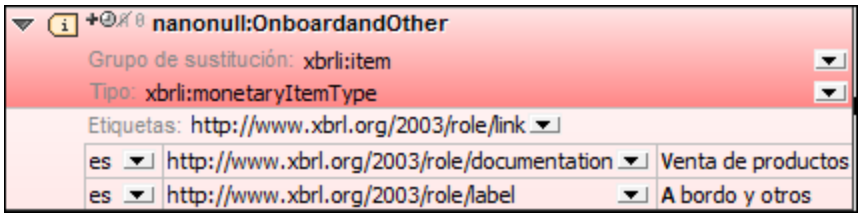
4. Ahora haga clic a la izquierda del icono en forma de reloj (*imagen siguiente*). Aparece un cuadro combinado. Seleccione la opción `credit`.



   El valor del atributo `xbrli:balance` pasa a ser `credit`.
5. Haga clic en el icono en forma de reloj (atributo `xbrli:duration`), en el icono A (atributo `xs:abstract`) y en el icono 0 (atributo `xs:nillable`) y seleccione los valores *duration*, *NOT Abstract* y *Nillable* respectivamente. (En el archivo `.xsd` los valores de estos atributos serán `credit`, `duration`, `false` y `true` respectivamente.)
6. Haga clic con el botón derecho dentro del recuadro del elemento y seleccione **Agregar función de vínculo de etiqueta**. Se crea una fila para la función de vínculo de etiqueta en la parte inferior del recuadro (*imagen siguiente*).



7. Seleccione el URI de XBRL.
8. Haga clic con el botón derecho en la fila de la función de vínculo de etiqueta y seleccione **Agregar etiqueta**. Se añade una fila para una etiqueta dentro de la función de vínculo de etiqueta.
9. En la fila recién creada para la etiqueta (*imagen siguiente*) haga doble clic en el primer campo (para el idioma) y escriba `es`. En el segundo campo, el campo de la función de vínculo, seleccione la función `documentation` en la lista desplegable. En el tercer campo, el campo etiqueta, escriba el texto que debe aparecer en la documentación. Después repita este paso y cree otra etiqueta para la función de vínculo `label`. Cuando se expande el recuadro del elemento (haciendo clic en la punta de flecha situada a la izquierda del nombre del elemento), puede mostrar u ocultar la función `label` haciendo clic en el símbolo +/- situado a la derecha de la etiqueta (Mostrar/Ocultar etiquetas).



   Ya hemos creado el elemento `nanonull:OnboardAndOther`.

 © 2016-2022 Altova GmbH

Recuerde que el elemento `OnboardAndOther` tenía un atributo `xbrli:balance` con valor `credit` (crédito). Esto se debe a que es un componente de ingresos: está entrando dinero. Ya que los productos que se venden a bordo de los cruceros vienen acompañados de gastos (p. ej. los costes de adquisición) vamos a crear también un elemento de débito llamado `nanonull:CostOfOnboardAndOther`. Cree este elemento igual que creó `nanonull:OnboardAndOther` con una sola diferencia: el valor del atributo `xbrli:balance` debe ser `debit` en lugar de `credit`.

Otro gasto que debemos incluir es el de las comisiones a los agentes. Para ello podemos usar un elemento de débito llamado `nanonull:CruiseCommissionsTransportationAndOther`. Cree este elemento siguiendo los mismos pasos que usó con `nanonull:CostOfOnboardAndOther`.

Para terminar añadimos tres elementos abstractos `Asia`, `Europe` y `US` para que los conceptos puedan agruparse por región. Puesto que el objetivo de estos elementos es poder agrupar los conceptos y no tendrán ningún valor, estos elementos son de tipo abstracto. El tipo de dichos elementos es inmaterial. Es mejor dar a un elemento abstracto un tipo que se corresponda con su semántica. Por ejemplo, a los elementos abstractos `Asia`, `Europe` y `US` les damos el tipo `stringItemType`. Cree los elementos `nanonull:Asia`, `nanonull:Europe` y `nanonull:USA` al igual que hizo con los elementos anteriores. La única diferencia es que el valor del atributo `Abstract` debe ser *Abstract* (el valor del atributo propiamente dicho en el archivo XSD será `true`) y que el atributo `xbrli:balance` no estará.

Nota: si hay un atributo `xbrli:balance` en un elemento abstracto, este debe ser de tipo `monetaryItemType`, o de lo contrario la taxonomía no será válida. Por eso recomendamos omitir el atributo opcional `xbrli:balance` de todos los elementos abstractos.

En el paso siguiente especificaremos las funciones de vínculo 851 para la nueva taxonomía. Estas funciones de vínculo serán necesarias a la hora de crear relaciones nuevas.

17.2.8 Relaciones y funciones de vínculo

Cuando se crea un conjunto de relaciones, estas se crean dentro de un elemento contenedor. Por ejemplo, cuando se crean relaciones de definición, los elementos que definen las relaciones de definición (los localizadores y arcos de definición) se crean dentro de un elemento `definitionLink` como este:

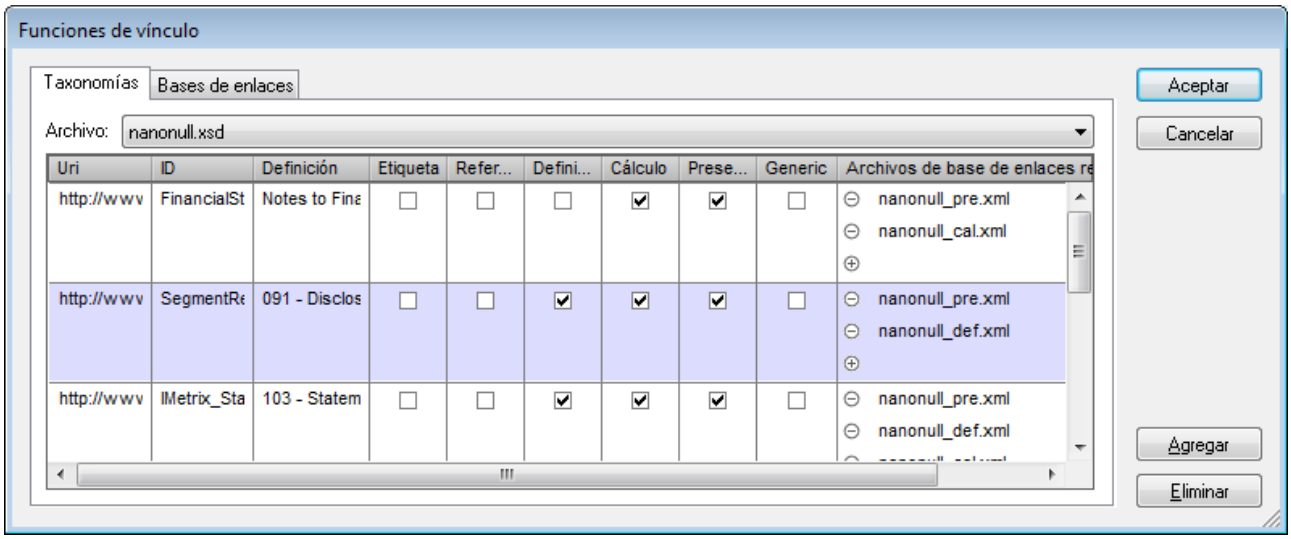
```
<link:definitionLink xlink:type="extended"
xlink:role="http://www.nanonull.com/taxonomy/role/SegmentRevenueAndOperatingIncome">
```

El valor del atributo `xlink:role` del vínculo de definición (como en el ejemplo anterior) debe ser el valor del atributo `roleURI` de una de las funciones de vínculo definidas para ser utilizadas en relaciones de definición (*ver el siguiente fragmento de código*). Una función de vínculo (como la del ejemplo siguiente) está dentro del elemento `appinfo` de la taxonomía.

```
<xs:appinfo>
   <link:roleType id="SegmentRevenueAndOperatingIncome"
   roleURI="http://www.nanonull.com/taxonomy/role/SegmentRevenueAndOperatingIncome">
      <link:definition>006091 - Disclosure - Segment Revenue and Operating
Income</link:definition>
      <link:usedOn>link:calculationLink</link:usedOn>
      <link:usedOn>link:definitionLink</link:usedOn>
      <link:usedOn>link:presentationLink</link:usedOn>
   </link:roleType>
</xs:appinfo>
```

Una función de vínculo se puede usar en los elementos contenedores de otras clases de relación, no sólo en los elementos `definitionLink` (en los elementos `calculationLink` y `presentationLink`, por ejemplo). Observe que en el fragmento de código anterior hay elementos `usedOn` que indican en qué clase de relaciones se puede utilizar esta función de vínculo.

Para crear funciones de vínculo en un archivo de definición de conceptos (el archivo de taxonomía principal) en la vista XBRL, haga clic en el comando de menú **XBRL | Funciones de vínculo**. Aparece el cuadro de diálogo "Funciones de vínculo" (*imagen siguiente*).



En la pestaña Taxonomías, en el cuadro combinado *Archivo*, seleccione la taxonomía en la que desea agregar una función de vínculo y haga clic en **Agregar**. Escriba el URI y el ID de la función de vínculo (*véase el fragmento de código anterior*). Después indique para qué clase de relaciones debe estar disponible esta función de vínculo marcando las casillas pertinentes (*ver imagen anterior*).

Archivo de ejemplo: paso 6

Siga las instrucciones anteriores y cree estas dos funciones de vínculo desde el cuadro de diálogo "Funciones de vínculo" (**XBRL | Funciones de vínculo**):

1. `id="SegmentRevenueAndOperatingIncome"` `URI="http://www.nanonull.com/taxonomy/role/SegmentRevenueAndOperatingIncome"` (que se podrá usar en relaciones de definición, cálculo y presentación)
2. `id="FinancialStatements"` `URI="http://www.nanonull.com/taxonomy/role/FinancialStatements"` (que se podrá usar en relaciones de cálculo y presentación)

En el paso siguiente aprenderemos a crear relaciones [853] para la taxonomía nueva.

© 2016-2022 Altova GmbH

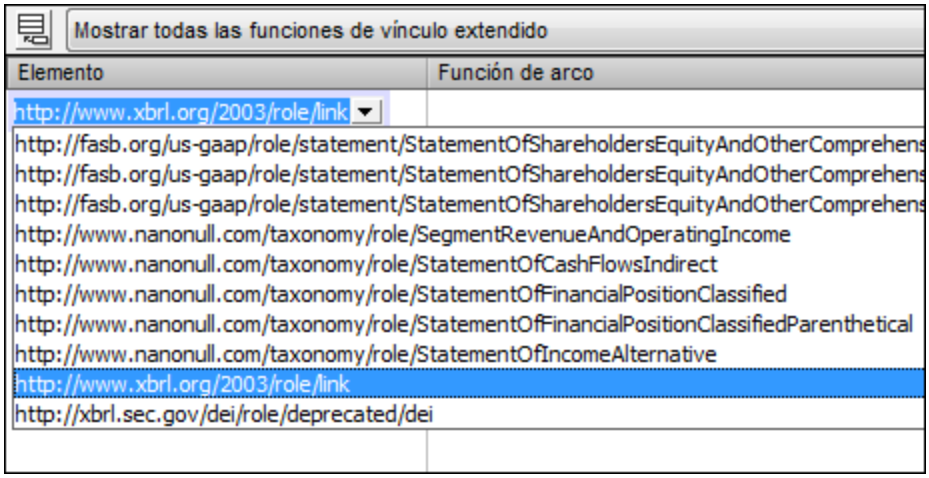
17.2.9 Crear relaciones: primera parte

Las relaciones se crean en las pestañas del editor de taxonomías: Definiciones, Presentación y Cálculo. Estos tres tipos de relaciones se crean de forma similar, pero la principal diferencia entre ellas es que las relaciones de definición tienen funciones de arco (*arcrole*), mientras que las relaciones de presentación y cálculo no tienen funciones de arco. En este apartado explicamos cómo se crean relaciones usando relaciones de definición. En el siguiente apartado 855 explicamos las diferencias con las relaciones de cálculo y presentación, así como otras características propias de las relaciones.

Le aconsejamos que abra una taxonomía en la vista **XBRL** mientras lee este apartado. Por ejemplo, puede utilizar la taxonomía Nanonull (`nanonull.xsd`) que está guardada en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023Examples\XBRLExamples\Nanonull`.

Agregar la función de vínculo

Haga clic en la pestaña pertinente de la ventana principal (Definiciones, Presentación o Cálculo). Después haga clic con el botón derecho y seleccione el comando **Agregar función de vínculo extendido**. Como resultado se añade una línea que contiene el URI de una función de vínculo predeterminada (*ver imagen*). Abra la lista desplegable y seleccione la función de vínculo que desea agregar.

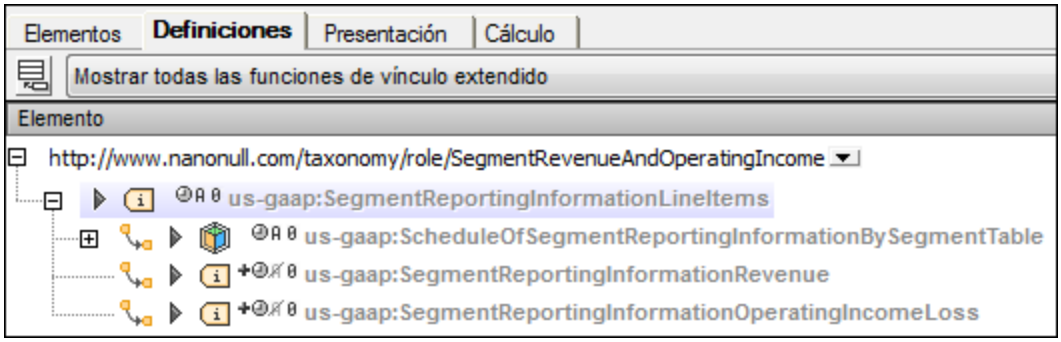


Si la función de vínculo que busca no está disponible es porque no se definió en la taxonomía o no se definió para esta clase de relación. Consulte el apartado Relaciones y funciones de vínculo 851 para más información sobre las funciones de vínculo y cómo se crean.

Puede agregar todas las funciones de vínculo que quiera.

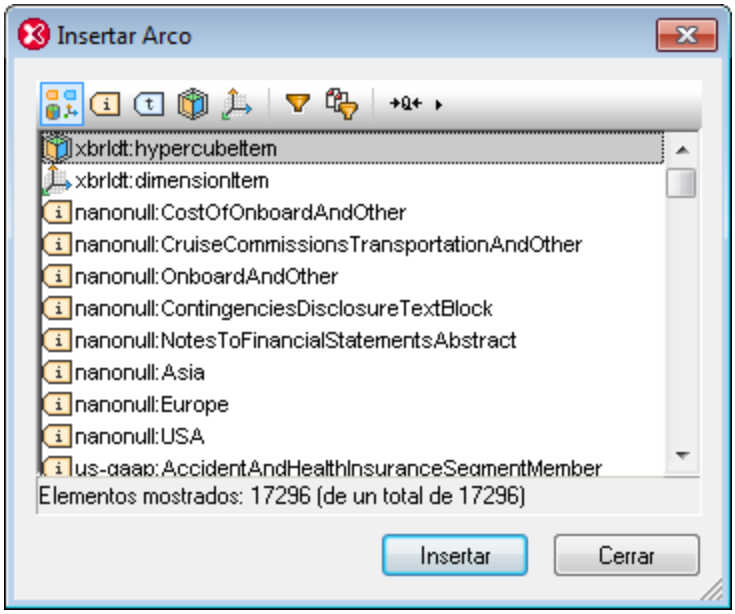
Insertar referencias de elemento y arcos dentro de una función de vínculo

El primer elemento que debemos crear dentro de una función de vínculo es un elemento **desde** el cual se creará una relación con otro elemento (*imagen siguiente*). Este primer elemento suele ser un elemento abstracto que agrupa otros elementos (por ejemplo, un elemento para un balance general). Este elemento no tendrá una entrada en la columna "Función de arco" porque se encuentra en el extremo `from` (origen) de un arco. Las funciones de arco aparecen en los elementos situados en el extremo `to` (destino) de un arco.



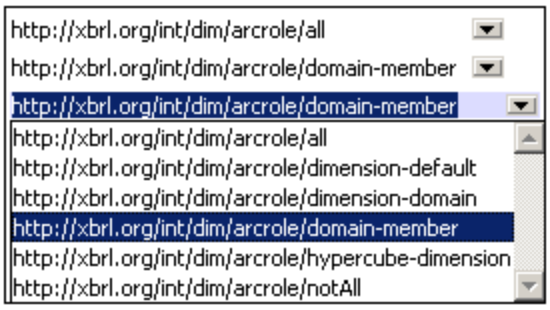
En la imagen anterior el elemento resaltado es la referencia de elemento insertada. Tiene tres arcos, uno cuyo destino es un hipercubo y dos cuyo destino es un ítem. Estos tres elementos están en el extremo `to` (destino) de sus arcos respectivos y la relación `from-to` se define mediante las funciones de arco, que aparecen en la columna "Función de arco".

Para insertar un arco en una referencia de elemento o en un elemento, haga clic con el botón derecho en el elemento `from` (origen) y seleccione el comando **Insertar arco**. Aparece el cuadro de diálogo "Insertar arco" (*imagen siguiente*). Seleccione el elemento que desea crear en el extremo `to` (destino) del arco. Para filtrar la vista en este diálogo, active el filtro y seleccione una condición (para más información consulte el apartado Ayudantes de entrada de la vista XBRL [274]).



El elemento se inserta con una función de arco predeterminada. Puede elegir la función de arco que prefiera de la lista desplegable (*imagen siguiente*).

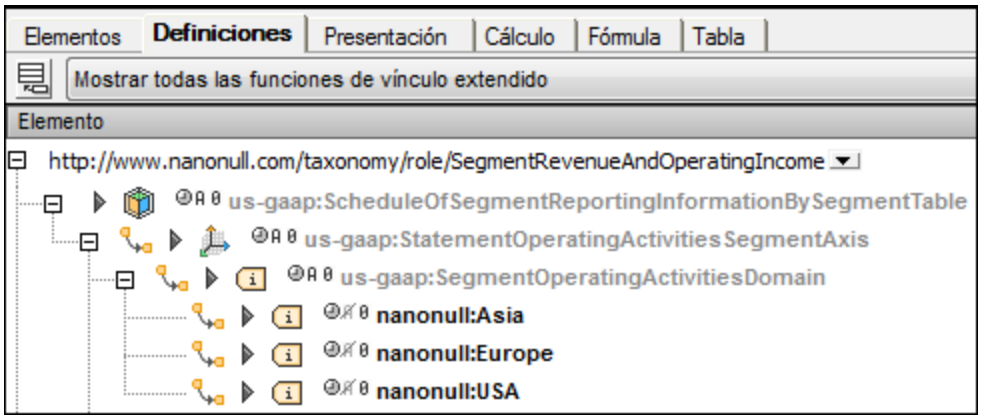
© 2016-2022 Altova GmbH



Nota: también puede agregar elementos, junto con sus arcos, arrastrándolos desde el ayudante de entrada Elementos globales.

Archivo de ejemplo: paso 7

Siguiendo las instrucciones de este apartado, cree las relaciones de definición que aparecen en esta imagen.



La imagen anterior muestra los elementos que debe añadir con sus arcos. La imagen siguiente muestra las funciones de arco de los elementos recién añadidos.

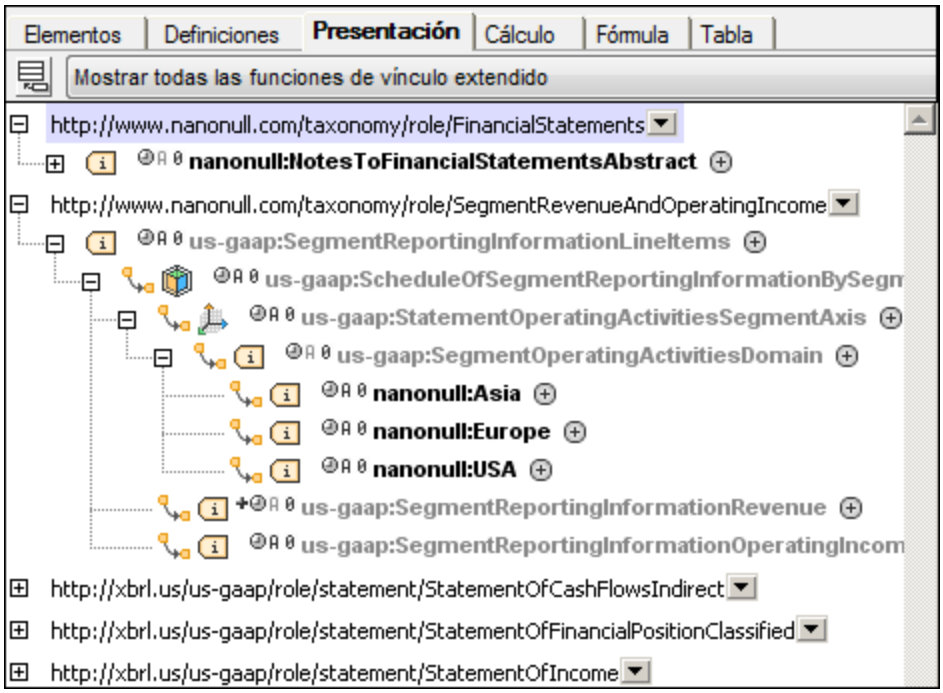


Puede comparar su taxonomía con la taxonomía de ejemplo que viene con XMLSpy. La taxonomía de ejemplo (`nanonull.xsd`) está en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023Examples\XBRLExamples\Nanonull`.

17.2.10 Crear relaciones: segunda parte

En el apartado anterior, Crear relaciones: primera parte 853, explicamos cómo se crean relaciones usando relaciones de definición. Las relaciones de presentación (*imagen siguiente*) y de cálculo se crean de manera

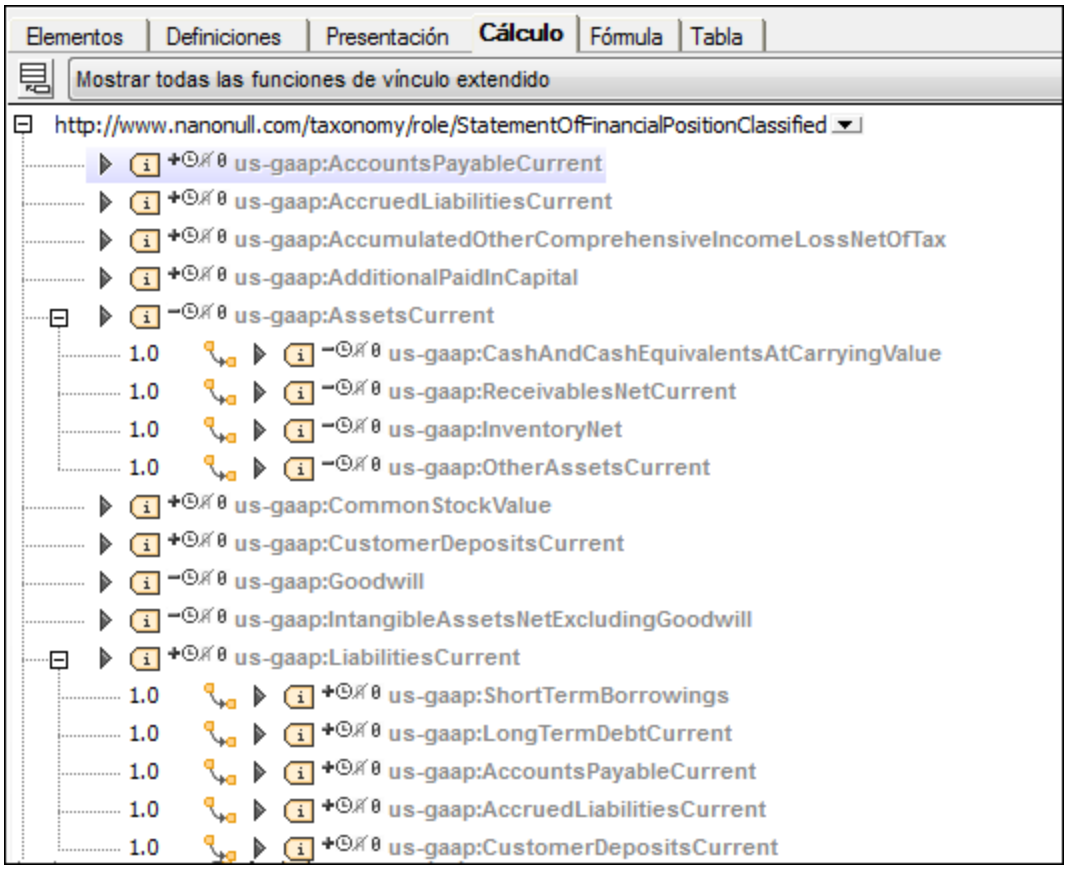
parecida. La única diferencia es que para las relaciones de presentación y cálculo no hay una columna "Función de arco".



Debe tener en cuenta estos aspectos:

- Las relaciones de presentación y cálculo se pueden entender como un mero arco entre dos elementos, como una relación primario-secundario. Los iconos en forma de arco representan esta relación. Esto quiere decir que insertar un arco en un elemento equivale a crear un elemento secundario en la representación gráfica. De esta forma podemos usar los arcos para construir una jerarquía.
- También puede arrastrar elementos del ayudante de entrada Elementos globales hasta la vista principal. Estos elementos se colocan siempre en la posición `to` (destino) del arco. Al arrastrar el elemento sobre una posición donde se puede colocar, aparece una flecha.
- Los arcos de cálculo tienen atributos de peso (`weight`), que indican cómo se debe sumar el valor del elemento `to` (destino) del arco (*ver imagen siguiente*). Por ejemplo, si el atributo de peso tiene el valor `+1.0` entonces se debe sumar el 100% del valor del elemento al valor del elemento `from` (o elemento de suma). Si el valor es `-1.0` entonces se debe sustraer el 100% del valor del elemento al valor del elemento de suma. Si hace doble clic en el valor del atributo de peso puede cambiar su valor.

© 2016-2022 Altova GmbH



El atributo de peso (`weight`) también se puede cambiar en el ayudante de entrada Detalles (*imagen siguiente*).

Prohibir el uso de un arco

Todos los arcos, ya sean de definición, presentación o cálculo, tienen un atributo de uso (`use`) que puede llevar el valor `optional` o `prohibited`. Si se usa el valor `prohibited`, entonces se invalida el arco.

Color y menú contextual

Cuando los elementos se crearon en la taxonomía activa y se pueden editar, entonces aparecen en negro. De lo contrario (cuando forman parte de taxonomías importadas, que no se pueden editar), los elementos aparecen en gris.

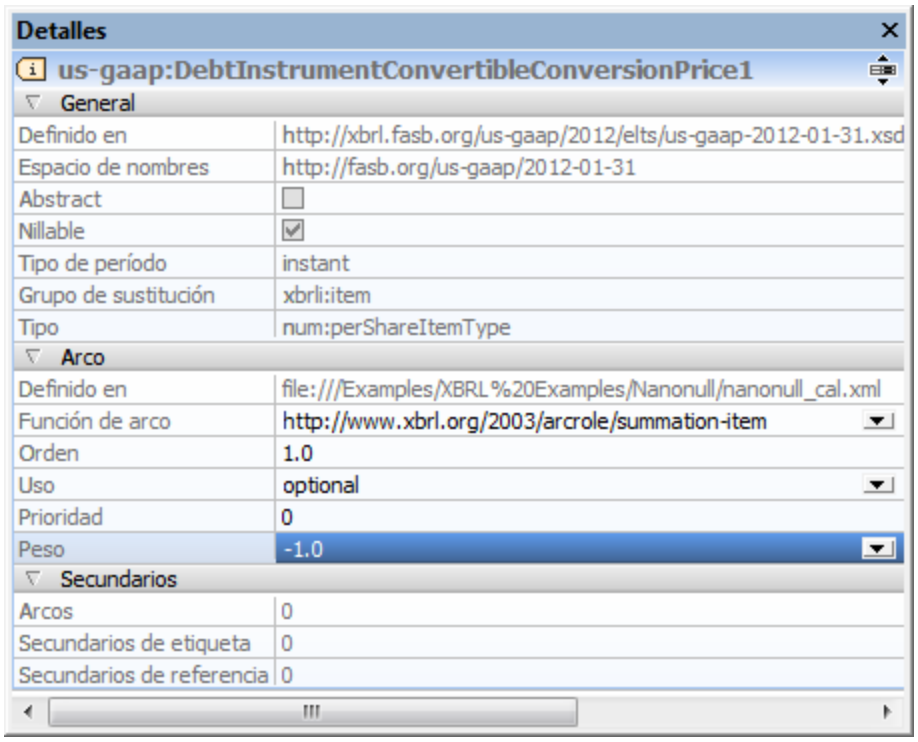
En las pestañas de la ventana principal los menús contextuales incluyen estos comandos:

- *Insertar referencia del elemento* (para las funciones de vínculo extendido). Agrega un elemento bajo la función de vínculo, que siempre estará en el extremo `from` (origen) de los arcos.
- *Eliminar referencia del elemento* (para las referencias de elemento situadas justo debajo de una función de vínculo).
- *Insertar arco* (para los elementos). Inserta un arco y aparece un cuadro de diálogo donde puede seleccionar el elemento que debe estar en el extremo `to` (destino) del arco.
- *Establecer función de destino.* Establece una función de destino en el elemento seleccionado.

- *Agregar función de vínculo de etiqueta.* Añade una función de vínculo de etiqueta al elemento seleccionado.
- *Agregar función de vínculo de referencia.* Añade una función de vínculo de referencia al elemento seleccionado.
- *Invalidar arco.* Reemplaza el valor `optional` (implícito) del atributo de uso (`use`) del arco con el valor `prohibited` y por tanto se invalida el arco.
- *Eliminar arco.* Elimina el arco seleccionado.
- *Mostrar en elementos globales.* Resalta el elemento seleccionado en el ayudante de entrada Elementos globales.

Ayudante de entrada Detalles

Cuando se selecciona un elemento de una relación, los atributos del arco se pueden editar en el ayudante de entrada Detalles (*ver imagen*).



Los atributos que no se pueden editar en la vista gráfica de la ventana principal (como `order` y `priority`) se pueden editar en el ayudante de entrada Detalles.

© 2016-2022 Altova GmbH

17.3 Procedimientos adicionales

El apartado Procedimientos adicionales reúne varias funciones de utilidad:

- Etiquetas preferidas 859
- Dominios con tipo 861
- Detectar duplicados y deduplicar 862
- Inline XBRL 863

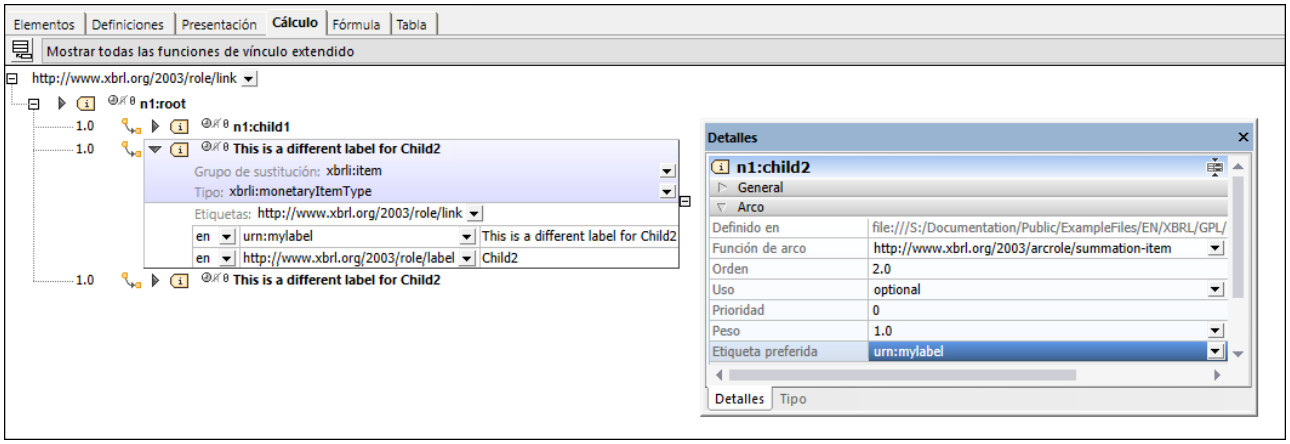
17.3.1 Etiquetas preferidas

Existen distintas etiquetas que se pueden asignar a un concepto o recurso genérico (fórmulas, tablas, etc.). En un arco de relación se selecciona la etiqueta deseada para el nodo de destino o nodo secundario mediante el rol de esa etiqueta. El mecanismo es el siguiente:

- si se ha definido una etiqueta preferida para el arco de relación, entonces se usa esta etiqueta
- si no se ha definido ninguna etiqueta preferida para una relación, entonces se usa la etiqueta predeterminada

Según la especificación de complementos Generic Preferred Label 1.0 Recommendation, el atributo **`gpl:preferredLabel`** se puede definir en las definiciones de cualquier arco y en cálculos, fórmulas y bases de enlaces de tablas. El valor del atributo es el rol de la etiqueta que se debe usar para seleccionar la etiqueta del nodo de destino. Como consecuencia, las etiquetas preferidas no solo se admiten para relaciones de presentación (habilitado por especificaciones anteriores), sino también para otras relaciones (como las de cálculo).

Etiquetas preferidas
Para definir etiquetas preferidas seleccione la relación en la pestaña principal y a continuación, en el ayudante de entrada Detalles, seleccione el rol de etiqueta que prefiera como valor de la propiedad *Etiqueta preferida* (*imagen siguiente*). En el ejemplo siguiente se aprecia que se han definido dos etiquetas para la segunda relación de cálculo. En el ayudante de entada Detalles, la etiqueta preferida es `urn:mylabel`, por lo que esta es la etiqueta que se usa para esa relación. (En XMLSpy puede indicar, en Configurar la vista XBRL 1527, que en lugar de nombres se muestre la etiqueta en la pantalla de conceptos y/o recursos.)



Nota: en bases de enlaces de tablas la propiedad `Etiqueta preferida` no se muestra para nodos de relación, nodos de aspecto y nodos de reglas combinadas, ya que no es posible definir etiquetas para esos nodos.

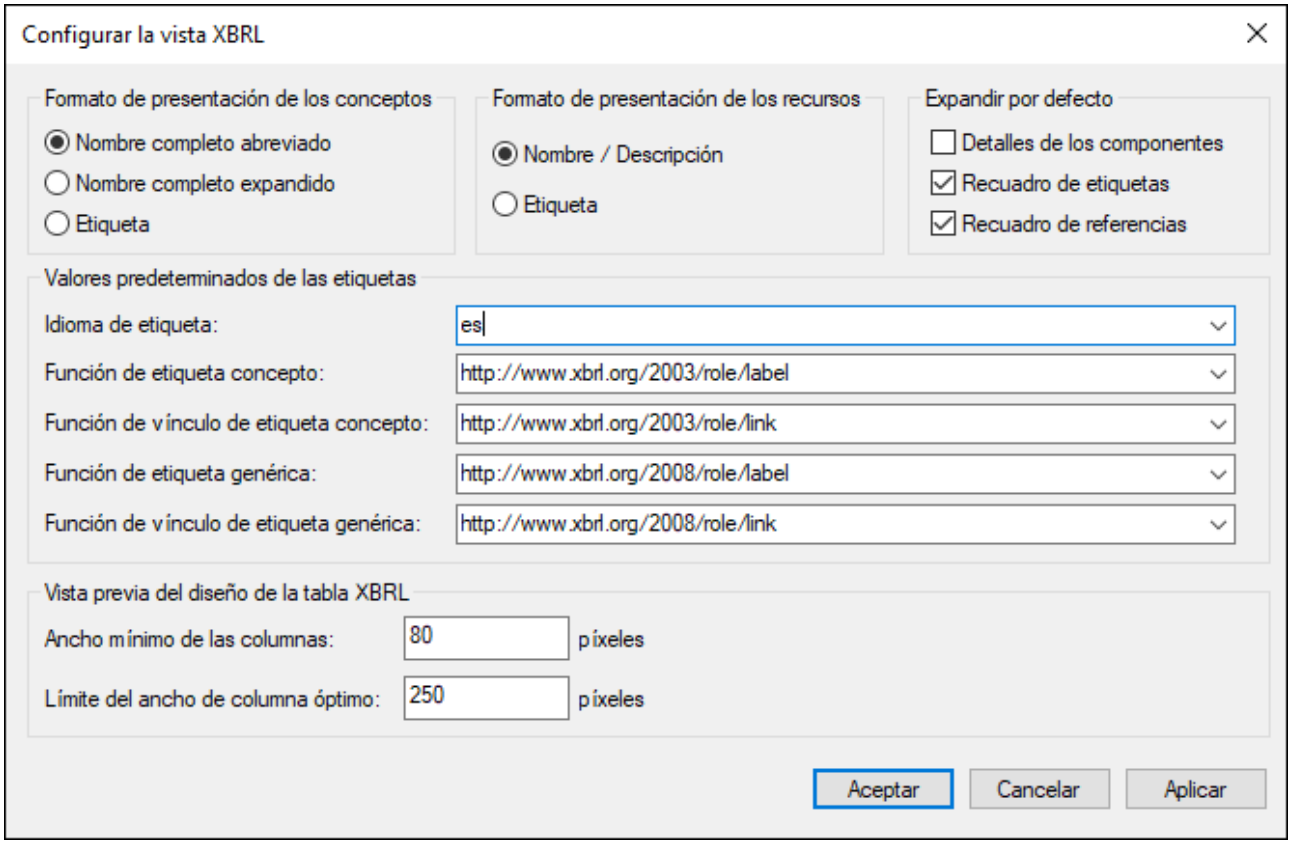
Etiquetas predeterminadas

Si la propiedad `Etiqueta preferida` de un concepto o recurso no está definida, entonces se usará la etiqueta predeterminada. Puede consultar las etiquetas predeterminadas de conceptos y recursos en el cuadro de diálogo Configurar la vista XBRL[1527].

Opciones de etiquetas al configurar la vista XBRL

Se accede al cuadro de diálogo Configurar la vista XBRL[1527] (*imagen siguiente*) con el comando de menú **XBRL | Configurar la vista**.

© 2016-2022 Altova GmbH

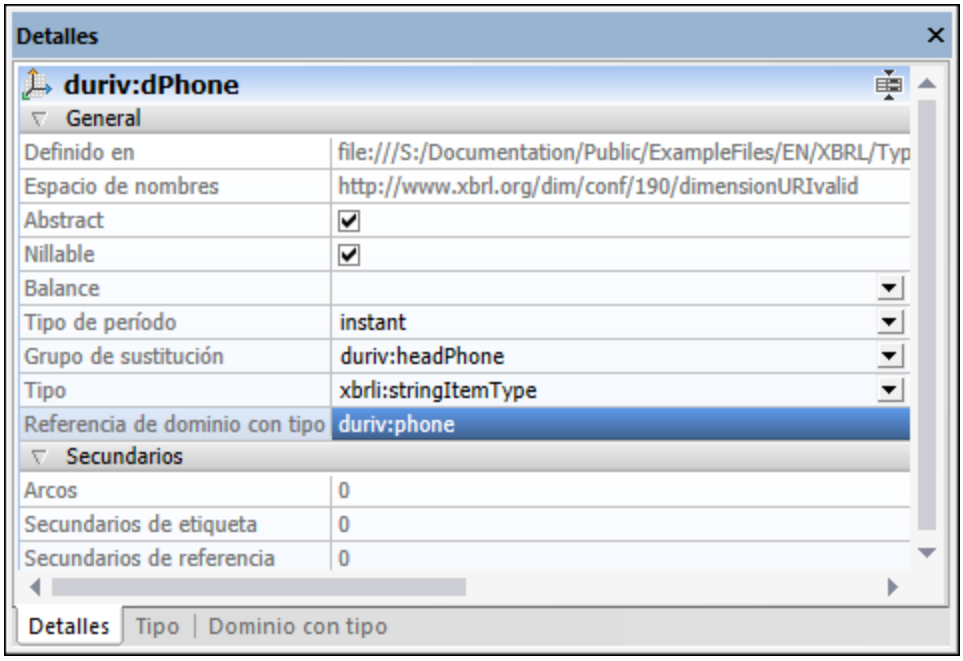


En este cuadro de diálogo existen dos opciones relevantes para las etiquetas:

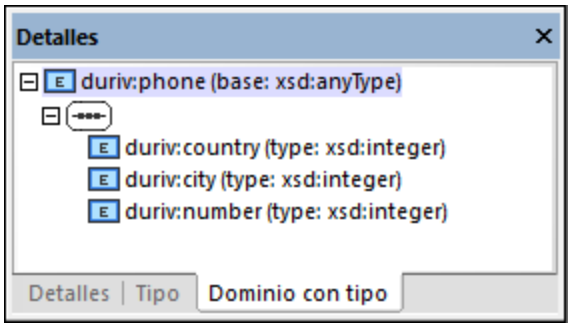
- puede especificar que se muestren las etiquetas de los conceptos y/o recursos en lugar de sus nombres.
- puede indicar etiquetas de conceptos y genéricas predeterminadas (que se usan para los recursos).

17.3.2 Dominios con tipo

Un dominio con tipo es la declaración de un elemento de que es referenciado desde una dimensión con tipo. En la imagen siguiente del ayudante de entrada Detalles, por ejemplo, se aprecia que la dimensión con tipo **`duriv:dPhone`** hace referencia al dominio con tipo **`duriv:phone`**.



Para ver información sobre el dominio con tipo al que se hace referencia haga clic en la pestaña *Dominio con tipo* (*imagen siguiente*). Puede usar los comandos del menú contextual de los elementos de esta pestaña para abrir el elemento seleccionado en la Vista esquema o copiar su ubicación en el portapapeles.



Nota: la pestaña *Dominio con tipo* solo aparece si se ha seleccionado una dimensión con tipo que haga referencia a un dominio con tipo.

17.3.3 Detectar duplicados y deduplicar

A continuación describimos las distintas formas de gestionar hechos duplicados en documentos de instancia XBRL:

- en documentos de instancia XBRL los duplicados se pueden detectar y mostrar en una lista en la ventana Mensajes. Para ello ejecute el comando de menú **XBRL | Detectar duplicados** o **XBRL | Detectar duplicados en el servidor (alto rendimiento)**. Para más información consulte la descripción de los comandos 1533.

- al validar un documento de instancia XBRL puede indicar que se ignoren los hechos duplicados que afecten a cálculos. Esta preferencia se puede habilitar como opción de validación XBRL[1619].
- al ejecutar fórmulas, generar tablas o transformar XBRL inline se pueden ignorar los duplicados que existen. Esta preferencia se puede especificar en el cuadro de diálogo "Opciones de procesamiento XBRL"[1536].

Los duplicados se determinan en base a las reglas definidas en la especificación Handling Duplicate Facts in XBRL and Inline XBRL 1.0.

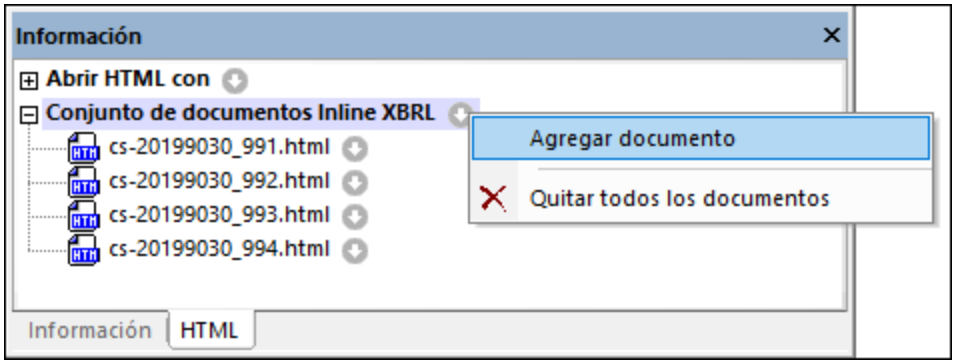
17.3.4 Inline XBRL

Los documentos Inline XBRL son documentos HTML que contienen datos XBRL (que está marcado con etiquetas XBRL). Puede validar el documento Inline XBRL activo (con el comando **XML | Validar**), además de procesarlo (con el comando **XBRL | Transformar Inline XBRL**). Al procesar los documentos se extraen los datos XBRL del documento HTML.

Varios documentos Inline XBRL

También puede validar/procesar varios documentos Inline XBRL. Para ello:

1. Abra el documento Inline XBRL principal para que sea el documento activo. En la Ventana Información[124] aparece la pestaña HTML (*imagen siguiente*).

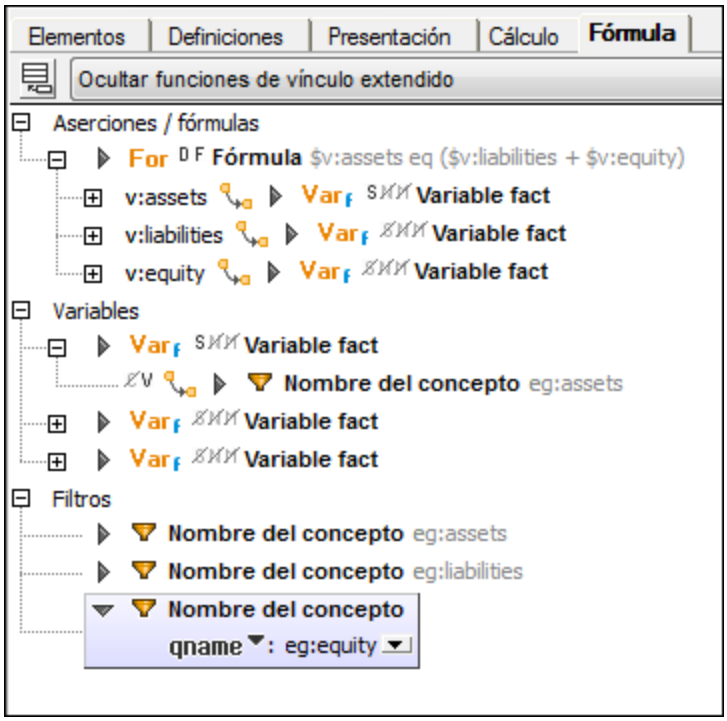


2. En la pestaña HTML haga clic en el botón de menú *Conjunto de documentos Inline XBRL*, después en **Agregar documento** (*imagen siguiente*) y luego navegue hasta los archivos Inline XBRL que quiera añadir.
3. Ejecute el comando de validación o procesamiento.

17.4 Editor de fórmulas XBRL

Las especificaciones XBRL Formula, XBRL Variables y XBRL Filters ofrecen una sintaxis para expresar las reglas que se pueden usar para derivar valores de hechos nuevos a partir de los datos de un informe XBRL. Las especificaciones Generic Labels y Generic references de XBRL permiten etiquetar todo tipo de construcciones XBRL. En el marco de XBRL Formula, estas etiquetas y referencias se pueden usar para asociar documentación humana con fórmulas, con sus variables y con filtros que definen qué hechos del informe XBRL son seleccionados por una variable para usarlos en la evaluación de una fórmula. Las especificaciones Validation y Assertion ofrecen una sintaxis para expresar reglas sobre el contenido esperado de informes, en lo que a variables, conjuntos de variables y fórmulas se refiere. En el borrador de trabajo [borrador de trabajo de XBRL Formula Overview 1.0] puede encontrar un resumen sobre la sintaxis y la semántica de XBRL Formula.

El editor de fórmulas XBRL de XMLSpy está implementado en el editor de taxonomías XBRL. Se trata de la pestaña **Fórmula** de la vista **XBRL** (*imagen siguiente*).



Para crear y editar fórmulas en la pestaña **Fórmula** puede ayudarse de los ayudantes de entrada Vista general y Detalles. En el ayudante Vista general puede definir la base de enlaces predeterminada para fórmulas XBRL (el archivo predeterminado donde se guardan las fórmulas), mientras que en el ayudante Detalles puede editar las propiedades y el contenido de los componentes de las fórmulas (aunque esto también se puede hacer en la pestaña **Fórmula** directamente).

17.4.1 Bases de enlaces y funciones de vínculo de fórmulas

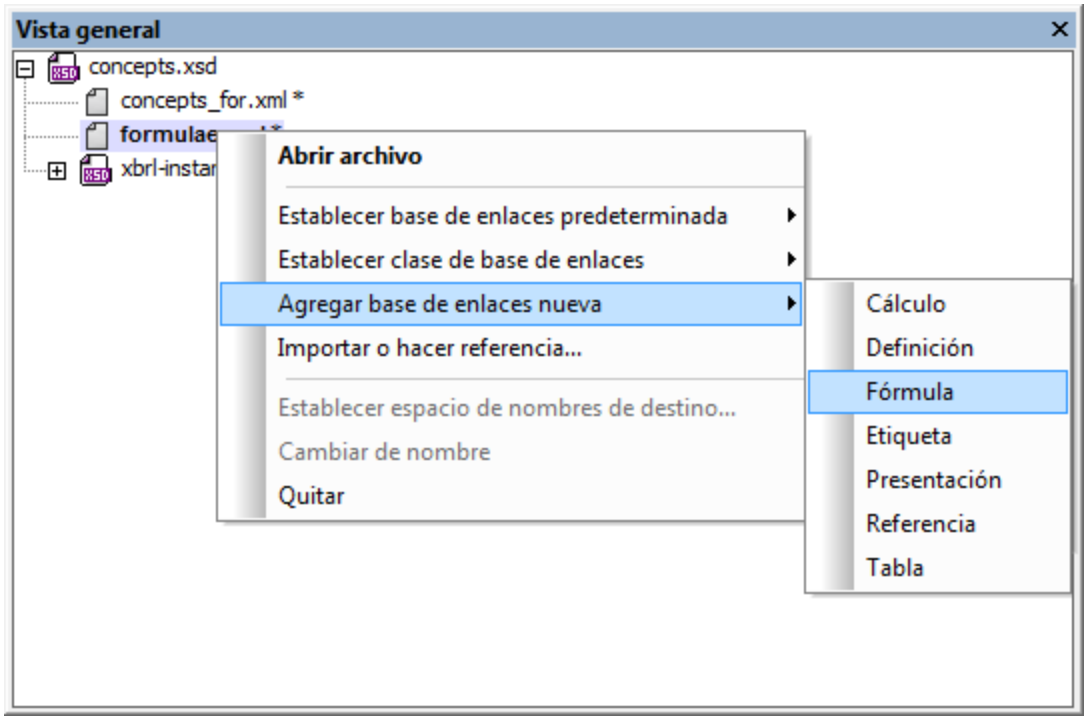
Mientras que las bases de enlaces estándar de XBRL (Definiciones, Presentaciones, Cálculos) definen relaciones entre conceptos a través de localizadores y arcos estándar en vínculos extendidos estándar, una

© 2016-2022 Altova GmbH

base de enlaces de fórmula define componentes de fórmula (fórmulas, variables, filtros, aserciones, etc.) y sus relaciones. Estas definiciones se especifican por medio de recursos y arcos genéricos en vínculos extendidos genéricos.

Agregar una base de enlaces de fórmula

En el ayudante de entrada Vista general (*imagen siguiente*), haga clic con el botón derecho en el archivo de taxonomía o en una base de enlaces y seleccione el comando **Agregar base de enlaces nueva | Fórmula**. La base de enlaces se convierte en la base de enlaces de fórmula predeterminada, es decir, el archivo en el que se guardan las nuevas definiciones de fórmula cuando se guarde la taxonomía. Si quiere usar otra base de enlaces de fórmula como base de enlaces predeterminada, haga clic con el botón derecho en la base de enlaces pertinente y seleccione **Establecer base de enlaces predeterminada | Fórmula** (*imagen siguiente*).



Recuerde que las bases de enlaces predeterminadas aparecen en negrita en este ayudante de entada y que las bases de enlaces con cambios no guardados llevan un asterisco.

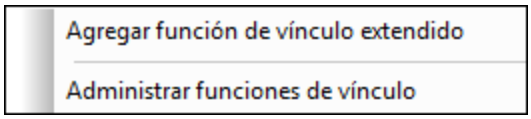
La base de enlaces de fórmula se abre en la pestaña **Fórmula** del editor de taxonomías.

Nota: si se añade un componente de fórmula a la taxonomía 866 cuando todavía no existe una base de enlaces de fórmula, XMLSpy crea una base de enlaces de fórmula automáticamente.

Funciones de vínculo

Al igual que los vínculos extendidos estándar (para Definiciones, Presentaciones y Cálculos), los vínculos genéricos deben definir un valor de función de vínculo extendido, que divide las relaciones del mismo tipo en redes inconexas. Todos los vínculos extendidos genéricos que tengan la misma función de vínculo se combinan bajo un solo nodo de función de vínculo en el diagrama de la pestaña *Fórmula, incluso si residen en archivos de base de enlaces distintos.*

Para crear funciones de vínculo genéricas en el diagrama de la pestaña **Fórmula** utilice el menú contextual que aparece al hacer clic con el botón derecho (*imagen siguiente*). No obstante, este menú contextual solo aparece si seleccionó **Mostrar todas las funciones de vínculo extendido** en el cuadro combinado situado en la parte superior de la pestaña **Fórmula**.

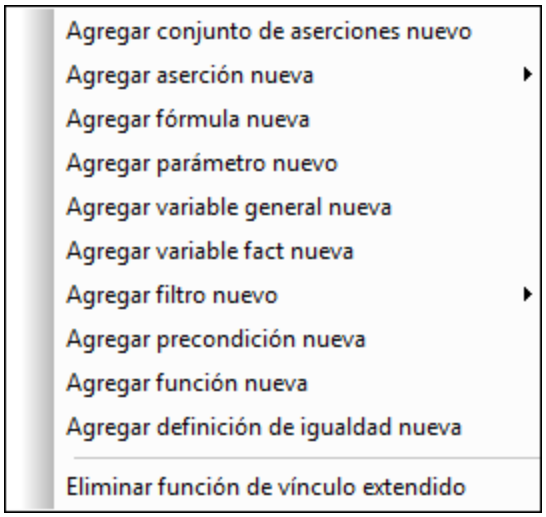


Este menú también aparece al hacer clic en el icono **Agregar función de vínculo extendido / administrar funciones de vínculo** de la barra de herramientas. Puesto que las redes de relaciones no son tan importantes para una base de enlaces de fórmula, la vista predeterminada de la pestaña **Fórmula** tiene activada la opción **Ocultar funciones de vínculo extendido**, que oculta las funciones de vínculo y, en su lugar, muestra los componentes de las fórmulas sin sus funciones de vínculo.

Si al crear la función de vínculo extendido no hay una base de enlaces de fórmula predeterminada, XMLSpy crea la base de enlaces de fórmula predeterminada automáticamente. Si al crear la función de vínculo no hay una función de vínculo en la base de enlaces predeterminada, XMLSpy crea una función de vínculo automáticamente en la base de enlaces predeterminada.

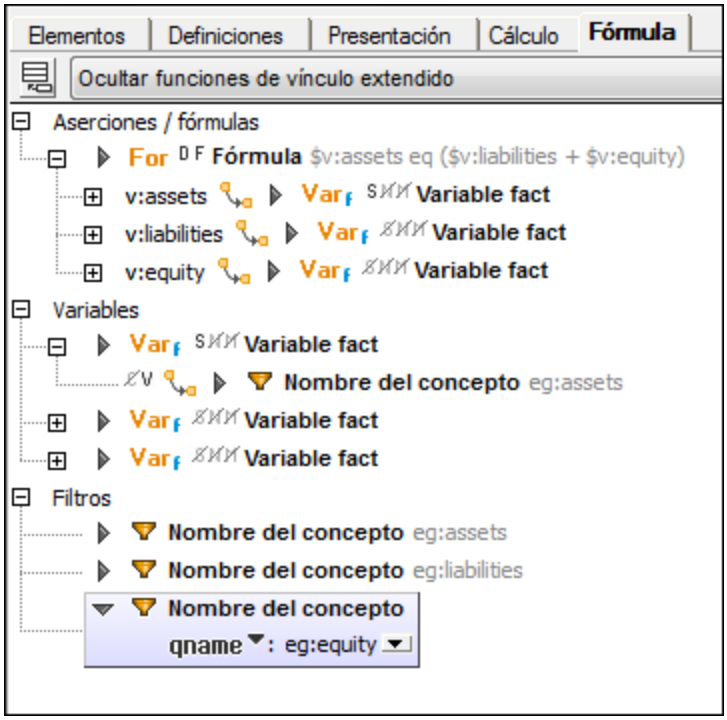
17.4.2 Componentes de fórmulas

Para crear nuevos componentes de fórmula utilice el menú contextual que aparece al hacer clic con el botón derecho en un nodo de función de vínculo (*imagen siguiente*) o con el icono **Agregar componente nuevo** de la barra de herramientas cuando la pestaña **Fórmula** está en modo **Ocultar funciones de vínculo extendido**.



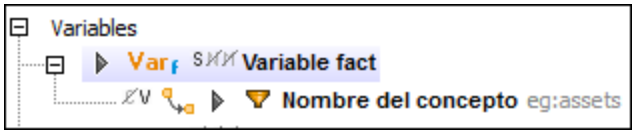
Los apartados de esta sección incluyen instrucciones para insertar cada tipo de componente. Una vez añadido, el componente de fórmula aparece en el diagrama de la pestaña **Fórmula** (*imagen siguiente*).

© 2016-2022 Altova GmbH



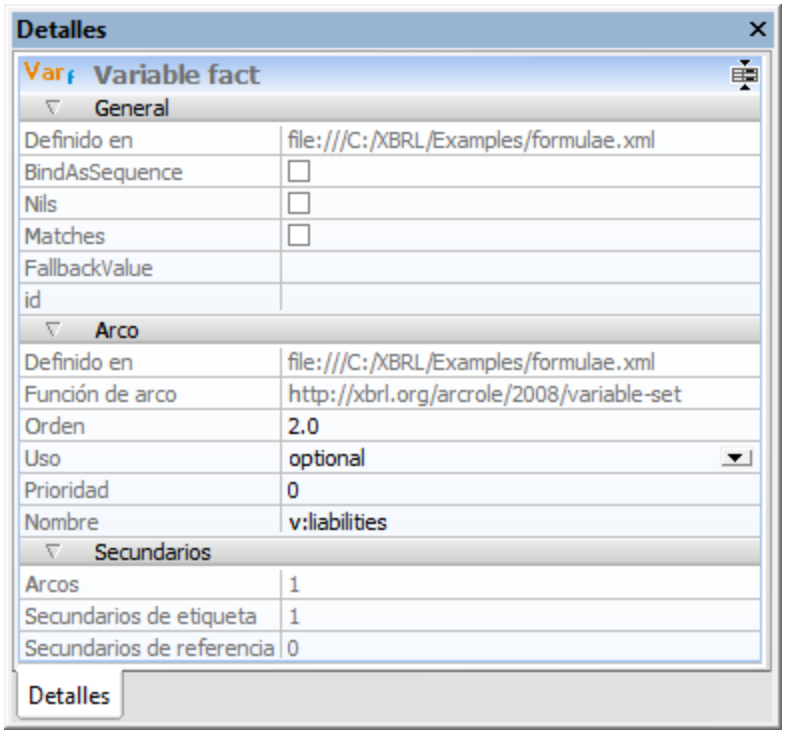
Por motivos de claridad, los componentes de fórmula se dividen en secciones y sus relaciones con otros componentes (los arcos) se presentan dentro de una estructura jerárquica (*imagen siguiente*).

Las propiedades de los componentes y de las relaciones (arcos) se representan como iconos a la izquierda del componente o arco respectivamente (*imagen siguiente*).



Por ejemplo, en la imagen anterior, el componente **`Variable fact`** tiene tres propiedades: `BindAsSequence` (el icono `S`), `Nils` (el icono `N`) y `Matches` (el icono `M`). Todas ellas son propiedades binarias. La primera (`BindAsSequence`) tiene el valor `true` y el icono `S` no está tachado. Las otras dos propiedades tienen el valor `false` (y sus iconos están tachados). El arco (situado debajo de la variable) tiene dos propiedades: la primera tiene el valor booleano `false` y la segunda `true`.

En el ayudante Detalles de **`Variable fact`** (*imagen siguiente*), las propiedades de la variable aparecen bajo la sección *General*. Los valores de las propiedades binarias se indican mediante casillas de verificación (`true` si está marcada, `false` si no lo está).



Para ver las propiedades del arco en el ayudante de entrada Detalles, seleccione en el diagrama el componente `to` (destino). Las propiedades del arco se presentan en la sección *Arco* (*imagen anterior*).

Menús contextuales del editor de fórmulas XBRL

Dependiendo del tipo de componente, el editor de fórmulas XBRL ofrece diferentes tipos de menús contextuales. Los comandos del menú se organizan por grupos según su función:

- Modificar el contenido (para fórmulas, algunos filtros y funciones personales). P. ej.: **Anexar/Insertar regla de aspecto**
- Modificar relaciones (solo para subelementos): **Invalidar/Eliminar arco**
- **Agregar etiquetas/referencias**
- Crear componentes secundarios nuevos (relaciones incluidas). P. ej.: **Agregar filtro nuevo**
- Eliminar componentes (relaciones incluidas)
- **Buscar la siguiente repetición/la repetición anterior** (del componente)

Nota: los elementos que se pueden crear o eliminar desde el menú contextual aparecen en el ayudante Detalles en secciones adicionales (p. ej. Regla de aspecto concepto).

17.4.2.1 Aserciones y conjuntos de aserciones

Hay tres tipos de aserciones

- Aserciones de valores
- Aserciones de existencia

© 2016-2022 Altova GmbH

- Aserciones de coherencia

Aserciones de valores

Las aserciones de valores son la característica más utilizada de las bases de enlaces de fórmula porque permiten comparar hechos de la instancia XBRL de entrada con una expresión XPath. Incluye los iconos de las propiedades `Aspect Model` y `Implicit Filtering`. El valor de la propiedad `test` es una expresión XPath.

Aserciones de existencia

Las aserciones de existencia permiten realizar comprobaciones de existencia estática, p. ej. para asegurarse de que los hechos descriptivos del documento están presentes (como el tipo de formulario, la identificación de la compañía y el número de identificación de la declaración de impuestos). Incluye los iconos de las propiedades `Aspect Model` y `Implicit Filtering`. El valor de la propiedad `test` es una expresión XPath.

Aserciones de coherencia

Las aserciones de coherencia definen cómo saber si un hecho de salida, generado por la fórmula asociada, es coherente con todos los hechos de aspecto equivalente de la instancia XBRL de entrada. Incluye el icono de la propiedad `strict`. El valor de las propiedades `Absolute Acceptance Radius` (radio absoluto de aceptación) y `Proportional Acceptance Radius` (Radio proporcional de aceptación) es una expresión XPath.

Mensajes de aserción satisfecha/no satisfecha

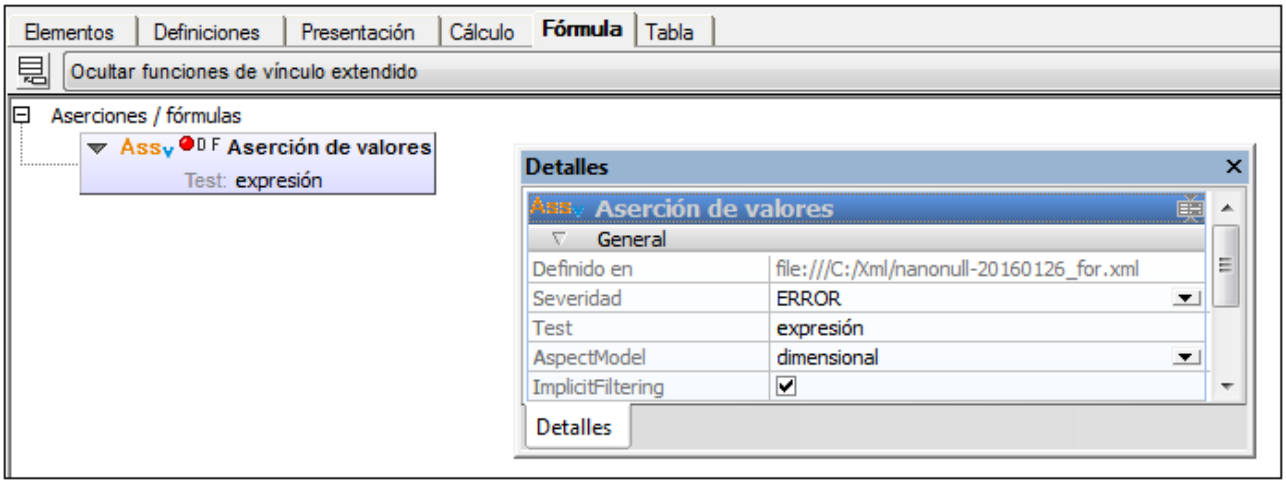
Estos subcomponentes de las aserciones permiten asociar mensajes a la evaluación de la aserción: mensajes satisfactorios a la evaluación que finaliza correctamente y mensajes no satisfactorios a evaluaciones que dan un resultado incorrecto. Estos mensajes se pueden añadir desde el menú contextual que aparece al hacer clic en las aserciones con el botón derecho.

Relaciones de gravedad de aserción no satisfecha

Una aserción puede ser satisfecha o no satisfecha. Sin embargo, como las aserciones tienen reglas con distintos niveles de importancia, las aserciones no satisfechas se clasifican según el nivel de gravedad. Hay tres niveles de gravedad estándar: `ERROR`, `ADVERTENCIA` y `OK`. El nivel de gravedad predeterminado es `ERROR` y se invoca cuando una aserción no tiene asociado un nivel de gravedad definido.

La relación de gravedad de aserción no satisfecha existe entre una aserción y uno de los recursos de gravedad definidos. Se expresa por medio de un arco XLink con: (i) el valor de función de arco `http://xbrl.org/arcrole/PR/2015-11-180/assertion-unsatisfied-severity`, (ii) una aserción como recurso de partida y (iii) un recurso de gravedad como recurso de destino.

En el editor de taxonomías, la relación de gravedad puede especificarse haciendo clic en el icono `Gravedad` del componente Aserción en el diagrama (*imagen siguiente*) y seleccionando el nivel de gravedad en el cuadro emergente. También puede seleccionar el nivel de gravedad en el ayudante de entrada Detalles de la aserción.



Conjuntos de aserciones

Un conjunto de aserciones contiene una o varias aserciones. El menú contextual de un conjunto de aserciones incluye comandos para añadir aserciones al conjunto.

17.4.2.2 Fórmulas

Una fórmula expresa un conjunto de reglas para la construcción de un hecho XBRL de salida, transformando los valores que las variables del conjunto de variables de la fórmula dan como resultado. Los valores de las variables se obtienen de una instancia XBRL de entrada y su DTS complementaria o del procesamiento que la aplicación hace de la fórmula.

La regla de valor es una expresión XPath que produce el valor que se debe asignar al hecho. Puede ser una expresión simple (p. ej. una constante) o puede incluir términos que hagan referencia a variables y parámetros del conjunto de variables, valores en cadena de otros conjuntos de variables y valores calculados de funciones personales e integradas.

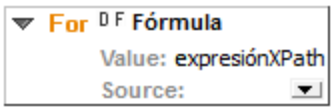
En los informes XBRL los hechos numéricos sin fracción se comunican con información sobre su precisión gracias al atributo `precision`/`decimals`. Por tanto, las fórmulas pueden contener reglas de precisión que determinen la precisión que debe afirmarse para el hecho de salida.

Al igual que las reglas que determinan los valores del hecho de salida y su precisión, las fórmulas especifican o dan a entender reglas de aspecto que establecen los valores para todos los aspectos de salida necesarios para interpretar los valores de salida. Hay reglas de aspecto para determinar el concepto de salida, el contexto de salida y las unidades de medida de salida (para hechos numéricos).

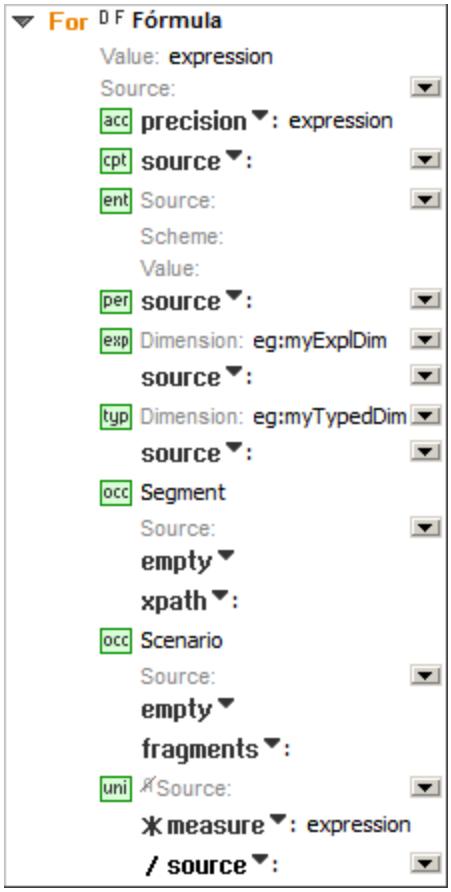
Un aspecto se obtiene (parcial o totalmente) de una variable enlazada de la evaluación especificando un origen. El origen puede indicarse en una regla o se puede heredar de un origen disponible en la fórmula (o tupla). Cuando hay varios orígenes, tiene prioridad el más cercano a la regla de aspecto.

Las fórmulas nuevas se insertan en el editor sin regla de aspecto ni regla de precisión (*imagen siguiente*).

© 2016-2022 Altova GmbH



Las reglas de precisión y aspecto se definen dentro del contenido de la fórmula y se pueden añadir (o eliminar) desde el menú contextual. La imagen siguiente, por ejemplo, muestra una fórmula con todas las reglas de precisión y aspecto posibles.



En el ayudante de entrada Detalles las reglas de precisión y aspecto aparecen en secciones independientes.

Regla de precisión

Kind: `precision` o `decimals`
Valor: expresión XPath

Reglas de aspecto

Las reglas de aspecto se agrupan según su clase.

Reglas de concepto
Kind: `qname`, `expr` o `source`
Valor: QName del concepto, expresión XPath o variable de origen (o el QName uncovered)

Reglas de identificador de identidad
Source: variable de origen (o el QName uncovered)
Scheme/valor: expresiones XPath

Reglas de período
Kind: `instant`, `duration`, `forever` o `source`
Valor: expresión XPath del valor, inicio/fin/origen, ningún valor o variable de origen (o el QName uncovered)

Reglas de dimensión explñicita
Dimension: QName de la dimensión afectada por la regla de dimensión explícita.
Kind: `qname`, `exp`, `omit` o `source`
Valor: QName del miembro, expresión XPath del miembro, ningún valor o variable de origen (o el QName uncovered)

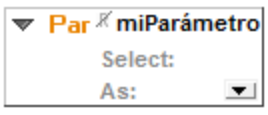
Reglas de dimensión con tipo
Dimension: QName de la dimensión afectada por la regla de dimensión con tipo.
Kind: `xpath, value, omit` o `source`
Valor: expresión XPath, elemento XML, ningún valor o variable de origen (o el QName uncovered)

Regla de aspecto OCC
Las reglas OCC se agrupan según su clase: reglas de segmento OCC y reglas de escenario OCC.
Source: variable de origen definida en la primera regla OCC.
Cada regla OCC debe tener:
Kind: `empty, fragments` o `xpath`
Valor: ningún valor, elementos XML o expresión XPath

Reglas de unidad
La marca binaria *Augment* indica si se debe usar o no el valor del aspecto de origen.
Cada regla de división/multiplicación de unidad debe tener:
Kind: `*measure, /measure` o `*source, /source`
Valor: expresión XPath de la medida o variable de origen (o el QName uncovered)

17.4.2.3 Parámetros

A un parámetro se le puede hacer referencia en expresiones XPath. Los parámetros ofrecen la marca `Required`. Si se aplica la marca `Required`, significa que el parámetro es obligatorio: es decir, su valor debe ser suministrado por la aplicación de procesamiento. Si el parámetro no es obligatorio y la aplicación de procesamiento no aporta ningún valor, el valor debe calcularse utilizando la expresión XPath dada en la propiedad `Select`. La propiedad opcional `As` especifica el tipo de datos que exige el parámetro.



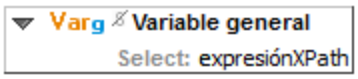
 © 2016-2022 Altova GmbH

17.4.2.4 Variables

Las variables declaran una forma de enlazar datos de entrada (que suelen ser hechos) a un nombre al que puede hacer referencia el nombre de variable desde, por ejemplo, una aserción o una expresión de fórmula. Las variables que enlazan con hechos de entrada (elementos fact) se llaman variables fact y usan filtros para declarar a qué se pueden enlazar en la instancia de entrada. Para los resultados de expresiones intermedios y otras clases de procesamiento se usan variables generales.

Variable general

Las variables generales incluyen el icono de la propiedad binaria `Bind As Sequence`. El valor de la propiedad `Select` es una expresión XPath.



Variable fact

Las variables fact incluyen los iconos de las propiedades binaria `Bind As Sequence`, `Nils` y `Matches`. El valor de la propiedad `Fallback Value` (Valor de reserva) es una expresión XPath.



17.4.2.5 Filtros

Un filtro define criterios de selección para hechos de la instancia XBRL de entrada, es decir, la instancia XBRL con respeto a la que se evalúan las variables. Los filtros expresan criterios que se pueden aplicar a los hechos de entrada. Algunos filtros pueden mostrar contenido XML en sublíneas.

Aspect cover
Estos filtros no realizan ningún filtrado y, por tanto, no tienen ninguna expresión XPath implícita. Se procesan o aplican después de los demás filtros (como filtros de conceptos y de dimensión) y reemplazan el estado de `cover` de los aspectos que resultan de la aplicación de los demás filtros.

Uno o varios elementos `aspect`
Kind: `aspect`, `dim-qname/excl-dim-qname` o `dim-exp/excl-dim-exp`
Valor: tipo de aspecto (enum), QName de la dimensión o expresión XPath
Los elementos se muestran en el ayudante Detalles en secciones independientes.

Filtros booleanos
Los filtros booleano están relacionados con subfiltros.
El filtro `and` filtra los hechos según los criterios expresados por uno de sus subfiltros.



El filtro `or` filtra los hechos según los criterios expresados por uno de sus subfiltros.



Filtros de nombres de conceptos
Estos filtros filtran los hechos según el nombre de sus conceptos.



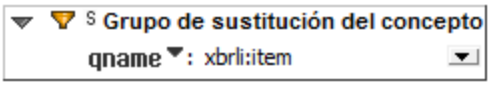
Uno o varios conceptos:
Clase: `qname` o `exp`
Valor: QName del concepto o expresión XPath
Los conceptos se muestran en el ayudante Detalles en secciones independientes.

Tipo de datos del concepto
Estos filtros se pueden usar para filtrar hechos según su tipo de datos XML Schema.



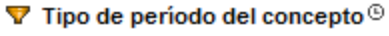
Marca binaria: `strict` indica si el tipo de datos del hecho debe ser no derivado o no.
Clase: `qname` o `exp`
Valor: QName del tipo de datos o expresión XPath

Grupo de sustitución del concepto
Estos filtros se pueden usar para filtrar hechos según su grupo de sustitución XML Schema.



Marca binaria: “strict” indica si el concepto del hecho debe especificar el elemento en su atributo `@substitutionGroup` directamente o no.
Clase: `qname` o `exp`
Valor: QName del grupo de sustitución o expresión XPath

Tipo de período del concepto
Estos filtros se pueden usar para filtrar los hechos que comunican valores para conceptos de tipo `duration` o de tipo `instant`, según lo indicado por su atributo `@xbrli:periodType`.



Balance del concepto
Estos filtros se pueden usar para filtrar los hechos que tienen un atributo `@xbrli:balance`, dependiendo de si su valor es `debit` o `credit`.

© 2016-2022 Altova GmbH

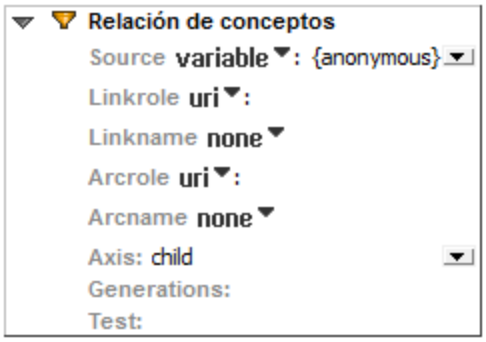


Atributo custom del concepto
Estos filtros se pueden usar para filtrar hechos según la existencia o valor de un atributo personalizado de la declaración de cada concepto.



Clase: `qname` o `exp`
Valor: QName del atributo o expresión XPath

Relación de conceptos
Estos filtros filtran hechos según las relaciones efectivas de sus conceptos con el concepto de origen, en una red de URI de función de vínculos de relaciones efectivas, de un URI de función de arco concreta, en un eje concreto, incluyendo generaciones concretas, y cumpliendo una expresión de prueba opcional.



Source: Kind = `variable`, `qname` o `exp`
Linkrole: Kind = `uri` o `exp`
Linkname: Kind = nada, `qname` o `exp`
Arcrole: Kind = `uri` o `exp`
Arcname: Kind = nada, `qname` o `exp`

Dimensión explícita
El dominio de dimensión explícita se define en el context de una DTD dada como el conjunto de todos los miembros del dominio en la unión de todos los dominios de miembros válidos de la dimensión del filtro. El filtro de dimensión explícita se puede usar para filtrar hechos con cualquier miembro de dominio de un dominio de dimensión explícita como valor para dicha dimensión explícita.



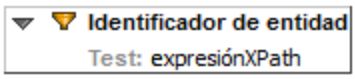
Clase de dimensión: `qname` o `exp`
Uno o más miembros:
Clase: variable, `qname` o `exp`
Los miembros se muestran en el ayudante Detalles en secciones independientes.

Dimensión con tipo
El filtro de dimensión con tipo se puede usar para filtrar hechos según el valor de una dimensión con tipo.

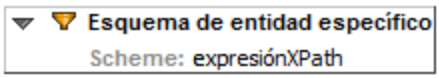


Clase de dimensión: `qname` o `exp`

Filtro de identificador de entidad
Estos filtros se pueden usar para filtrar hechos según las características del esquema de identificación de entidad y el valor de la identificación de entidad.



Esquema de entidad específico
Estos filtros se pueden usar para filtrar los hechos que comunican valores para el esquema identificado por el filtro.



Esquema de entidad regexp
Estos filtros se pueden usar para filtrar hechos según patrones regulares del texto del esquema de entidad.



Identificador de entidad específico
Estos filtros se pueden usar para filtrar los hechos que comunican valores usando el valor del identificador de entidad dado por el filtro.

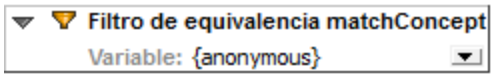
Identificador de entidad regexp
Estos filtros se pueden usar para filtrar hechos según patrones regulares del texto del valor del identificador de entidad.

General
El filtro general no abarca ningún aspecto.

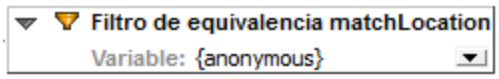


© 2016-2022 Altova GmbH

MatchConcept
Los filtros de concepto equivalente se pueden usar para seleccionar los hechos que comuniquen valores para el mismo concepto.



MatchLocation
Los filtros de ubicación equivalente se pueden usar para seleccionar los hechos que tengan el mismo elemento primario.

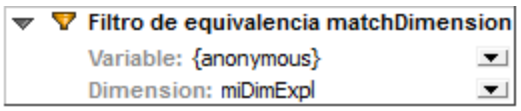


MatchUnit
Los filtros de unidad equivalente se pueden usar para seleccionar los hechos que tengan la misma unidad.

MatchEntityIdentifier
Los filtros de identificador de entidad equivalente se pueden usar para seleccionar los hechos que tienen el mismo identificador de entidad.

MatchPeriod
Los filtros de período equivalente se pueden usar para seleccionar los hechos que tienen el mismo período.

MatchDimension
Los filtros de dimensión equivalente se pueden usar para seleccionar los hechos que tienen el mismo valor para una dimensión XBRL determinada.



MatchSegment
Los filtros de segmento completo equivalente se pueden usar para seleccionar los hechos que tengan el mismo segmento, cuando el contenido del segmento no se interpreta según la especificación XBRL Dimensions.

MatchNonXDTSegment
Los filtros de segmento no XDT equivalente se pueden usar para seleccionar los hechos que tengan el mismo segmento, después de excluir el contenido XBRL Dimension de la comparación.

MatchScenario
Los filtros de escenario completo equivalente se pueden usar para seleccionar los hechos que tengan el mismo escenario, cuando el contenido del escenario no se interpreta según la especificación XBRL Dimensions.

MatchNonXDTScenario
Los filtros de escenario no XDT equivalente se pueden usar para seleccionar los hechos que tengan el mismo escenario, después de excluir el contenido XBRL Dimension de la comparación.

Período
Los filtros de período se pueden usar para filtrar los hechos según un amplio abanico de criterios relacionados con el período de tiempo durante el que los hechos se miden.



Inicio de período
Los filtros de inicio de período se pueden usar para filtrar los hechos según el inicio de la duración durante la cual se miden los hechos.



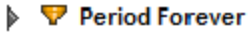
Fin de período
Los filtros de fin de período se pueden usar para filtrar los hechos según el fin de la duración durante la cual se miden los hechos.



Instante de período
Los filtros de instante de período se pueden usar para filtrar los hechos según el instante en el que se miden.



Período forever
Los filtros forever se pueden usar para filtrar los hechos que se comunican con un período forever.

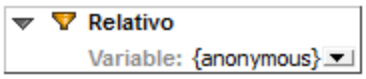


Duración de instante de período
Los filtros de instante-duración se pueden usar para filtrar los hechos que se comunican en un instante en el que dicho instante coincide con el inicio o el fin de la duración para la cual se comunicó otro hecho.



 © 2016-2022 Altova GmbH

Relativo
Los filtros relativos se pueden usar para seleccionar hechos para los cuales los aspectos que abarca el filtro relativo tienen valores que coinciden con los aspectos correspondientes de otro hecho. El hecho que filtra el filtro relativo debe ser el resultado de la evaluación de otra variable fact del conjunto de variables que se está evaluando.



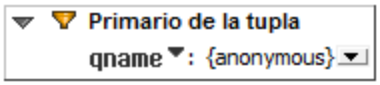
Segmento
Los filtros de segmento se pueden usar para filtrar hechos que tienen contenido no XDT que satisface las restricciones indicadas. El contenido no XDT es el contenido del segmento que no está basado en dimensiones explícitas o con tipo definidas en la especificación XBRL Dimensions.



Escenario
Los filtros de escenario se pueden usar para filtrar hechos que tienen contenido no XDT que satisface las restricciones indicadas. El contenido no XDT es el contenido del escenario que no está basado en dimensiones explícitas o con tipo definidas en la especificación XBRL Dimensions.



Primario de la tupla
Los filtros de elemento primario se pueden usar para seleccionar los hechos que tienen el elemento primario indicado.



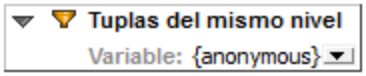
Clase: `qname` o `exp`

Antecesor de la tupla
Los filtros de elemento antecesor se pueden usar para seleccionar los hechos que tienen el elemento antecesor indicado.

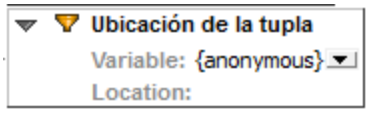


Clase: `qname` o `exp`

Tuplas del mismo nivel
Los filtros de elementos del mismo nivel se pueden usar para seleccionar los hechos que son elementos del mismo nivel que otro hecho.



Ubicación de la tupla
Los filtros de ubicación se pueden usar para seleccionar los hechos que tienen la ubicación indicada relativa a la ubicación de otro hecho.

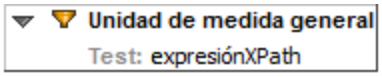


Unidad de medida simple
Los filtros de unidad de medida simple se pueden usar para filtrar hechos que se comunican con una unidad que se indica por medio de una medida simple.

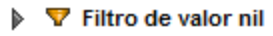


Clase: `qname` o `exp`

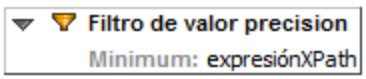
Unidad de medida general
Los filtros de unidad de media general se pueden usar para seleccionar hechos según los criterios que afectan a varias medidas de unidad.



Filtro de valor nil
Los filtros de valor nil se pueden usar para filtrar hechos que se comunican con el valor nil.



Filtro de valor precision
Los filtros de valor precision se pueden usar para filtrar los hechos que tengan una precisión mínima real o inferida, teniendo en cuenta que la precisión se puede inferir del valor del atributo `@decimal`. Recuerde que el filtro de valor precision no seleccionará hechos si el filtro implica una precisión mínima necesaria infinita. El filtro tampoco seleccionará hechos no numéricos ni hechos que se comuniquen con un valor nil.



 © 2016-2022 Altova GmbH

17.4.2.6 Precondiciones

Las precondiciones sirven para determinar si un conjunto de variables enlazadas pueden activar un valor de la fórmula y un hecho de salida o una prueba de valor de aserción o recuento de existencia.

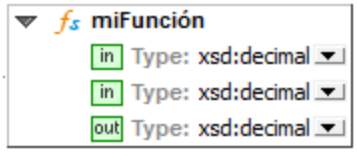


17.4.2.7 Funciones

Una función personal es una función XPath que no está definida en la especificación XPath Functions ni XQuery Functions ni en el registro de XBRL Functions. Las funciones personales se pueden usar dentro de expresiones XPath.

Firma de las funciones

La firma de una función es como la que aparece en esta imagen:

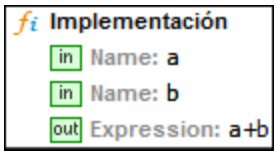


Los elementos secundarios, si los hay, de una firma de función personal indican los tipos de datos de los parámetros de entrada de la función personal. El orden de los parámetros de entrada de la función personal coincide con el orden de documento de los elementos secundarios de la firma de la función personal.

Los datos de entrada aparecen en el ayudante de entrada Detalles en secciones independientes.

Implementación de las funciones

La implementación de una función es como la que aparece en esta imagen:



La implementación de una función personal (CFI) contiene una secuencia de elementos secundarios que sirven para definir nombres para las entradas de la función, para expresar expresiones XPath que comprenden la implementación de la función personal y par definir el resultado de la función personal.

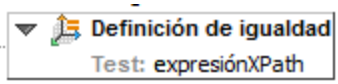
La relación función-implementación es la relación que existe entre la firma de una función personal y la implementación de una función personal. Como la implementación debe ser el destino de la relación función-implementación, siempre se muestra debajo de la firma correspondiente. Si la relación falta (o si la firma se

define bajo una función de vínculo distinta), la implementación se muestra bajo la sección *Funciones* directamente.

Las entradas y los pasos de la función se muestran en el ayudante Detalles en secciones independientes.

17.4.2.8 Definiciones de igualdad

Una definición de igualdad es una definición de igualdad entre dos valores cualquiera de una definición de dominio de dimensión con tipo. Esta es el elemento de un esquema XML que define el modelo de contenido para una dimensión con tipo y que se define como tal por medio del atributo `@xbrldt:typedDomainRef` del elemento XML Schema que declara una dimensión con tipo. Una relación definición-igualdad es la relación entre una definición de dominio de dimensión con tipo y una definición de igualdad. La relación definición-igualdad se muestra como una relación inversa entre la definición de igualdad y la dimensión con tipo correspondiente.



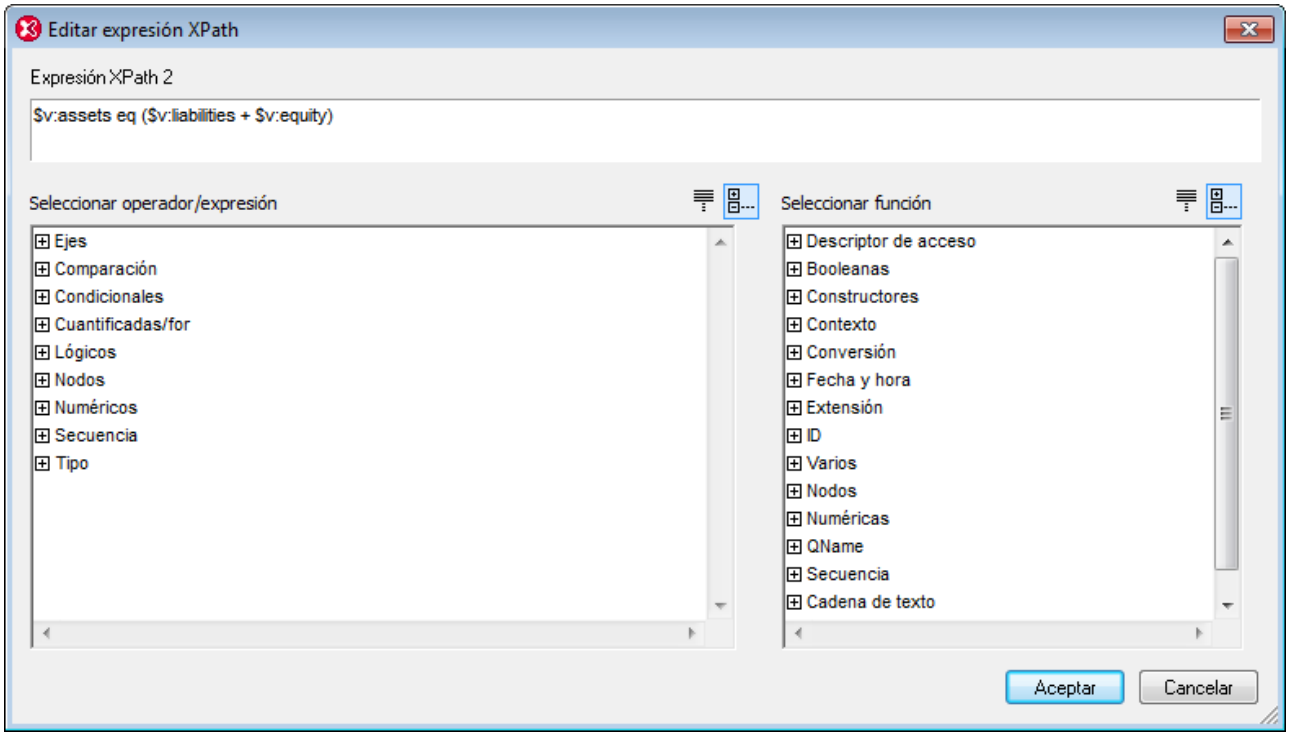
Puede crear una relación igualdad-definición arrastrando una dimensión con tipo desde el ayudante Elementos globales hasta el componente **`Definición de igualdad`**. Recuerde que ni la definición de igualdad ni la dimensión con tipo pueden participar todavía en una relación igualdad-definición.

17.4.3 Editar el contenido y las propiedades de los componentes

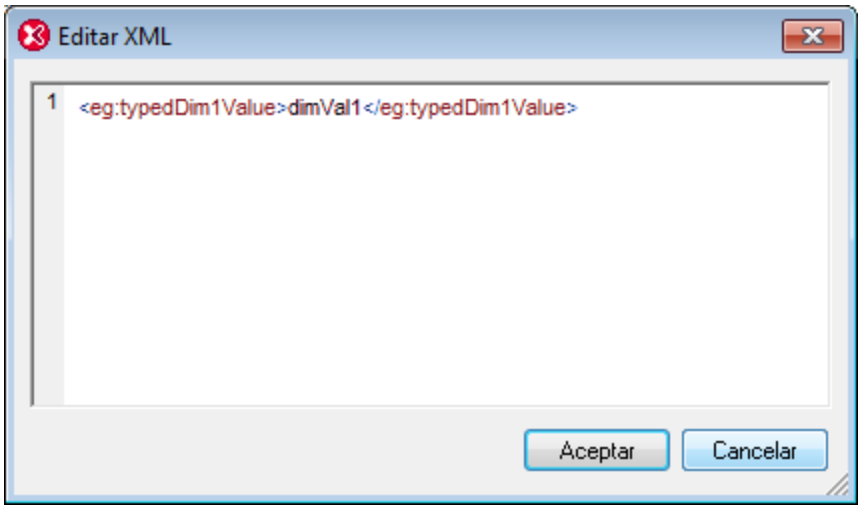
Las propiedades de los componentes de una fórmula se pueden editar en el diagrama directamente o en el ayudante de entrada Detalles.

En el diagrama, cuando un componente está contraído, su nombre (si lo tiene) o el valor de la propiedad predeterminada adecuada aparecen en gris junto al texto descriptivo del componente. Haga doble clic en el componente para expandirlo. Haga doble clic en una propiedad para cambiar al modo de edición. Si una propiedad o el contenido del componente tiene una expresión XPath, se abre el cuadro de diálogo "Editar la expresión XPath" (*imagen siguiente*).

© 2016-2022 Altova GmbH



Si edita contenido XML, como el contenido de la propiedad `Value` de una *Regla de aspecto dimensión con tipo* o el contenido de `Fragments` de una *Regla de aspecto OCC,* se abre el cuadro de diálogo "Editar XML" (*imagen siguiente*). Recuerde que para añadir una regla de aspecto a una fórmula debe hacer clic con el botón derecho en la fórmula y elegir el comando correspondiente.



Introduzca el texto XML en el cuadro de diálogo "Editar XML" y pulse **Aceptar**. El texto XML se introduce como contenido de la propiedad. Si el formato XML del texto no es correcto, se le notifica por medio de un mensaje y el texto no se admite como valor de la propiedad.

17.4.4 Relaciones entre los componentes

Para crear una relación entre dos componentes de la fórmula, puede crear un vínculo entre ambas mediante una operación de arrastrar y colocar. Las relaciones se representan en forma de arcos en el diagrama (*imagen siguiente*).



Las operaciones de edición y visualización disponibles son las siguientes:

- El orden de los secundarios de un componente depende de los valores de la propiedad del arco `Order`, que se puede modificar moviendo los secundarios mediante operaciones de arrastrar y colocar (*imagen anterior*).
- Un componente secundario se puede arrastrar encima o debajo de un componente primario distinto para copiar o mover la relación (y sus propiedades).
- Cuando un componente nuevo se crea desde el menú contextual de un componente (primario) ya existente, la relación (el arco) también se genera automáticamente.
- Los comandos **Invalidar arco** y **Eliminar arco** del menú contextual de un componente secundario sirven respectivamente para invalidar y quitar la relación que existe entre el componente y su primario.
- Al igual que con las relaciones entre conceptos, los arcos de las relaciones reemplazadas se muestran en sublíneas.

Nota: la función de arco de las relaciones de componentes de fórmulas no se puede modificar.

Relaciones variable-set

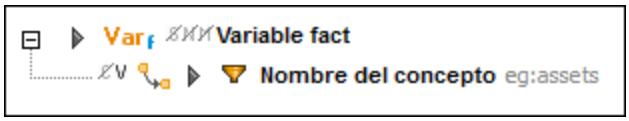
Una relación variable-set es la relación que existe entre (i) un recurso variable-set (una aserción de valores, una aserción de existencia o una fórmula) y (ii) una variable (variable `fact` o general) o un parámetro. El nombre `Name` de una variable o parámetro aparece delante del icono del arco (*imagen siguiente*).



Relaciones variable-filtro

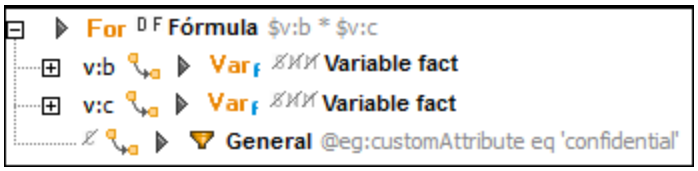
Una relación variable-filtro es la relación que existe entre una variable `fact` y un filtro. Si se estableció la marca binaria `Complement` (icono `C`), la relación es una relación variable-filtro con `complement`. Si se estableció la marca binaria `Cover` (icono `V`), la relación es una relación variable-filtro con `cover` (*imagen siguiente*). Este caso el filtro abarca aspectos de los hechos que se están filtrando.

© 2016-2022 Altova GmbH



Relaciones variable-set-filtro

Una relación variable-set-filtro (*imagen siguiente*) es la relación que existe entre un recurso variable-set y un filtro. El filtro que participa en una relación variable-set-filtro está asociado por definición con cada una de las variables `fact` del conjunto de variables definido por el recurso con el que está relacionado. La marca binaria `Complement` indica si las variables usan el complemento de filtro o no. Todos los filtros que están asociados a variables `fact` mediante relaciones variable-set-filtro no abarcan ningún aspecto, por definición.

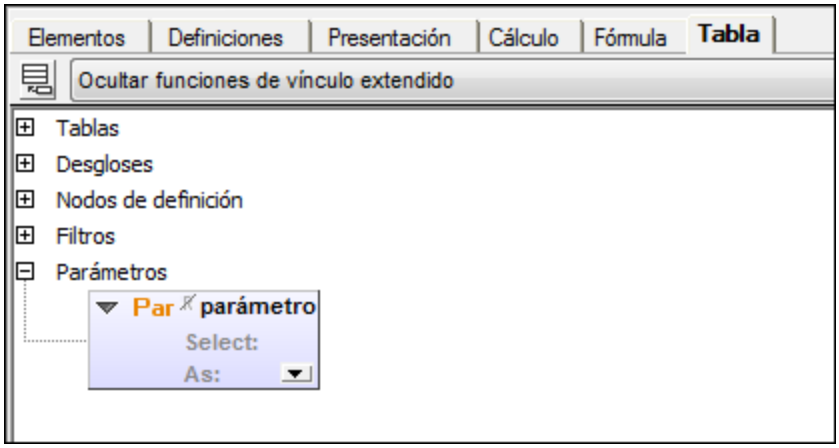


Generar fórmulas de forma visual en la vista previa del diseño de la tabla

XMLSpy también ofrece una práctica función para obtener una vista previa del diseño de la tabla XBRL que le ayudará a desarrollar sus taxonomías. Para más información consulte el apartado Generar fórmulas en la vista previa del diseño de la tabla 918.

17.4.5 Parámetros

Puede usar parámetros XBRL en expresiones XPath de fórmulas y de definiciones de tabla. Los parámetros que se usarán como parámetros de fórmula (residentes en la base de enlaces de fórmula) se crean en la pestaña **Fórmula**, mientras que los parámetros de tabla (residentes en la base de enlaces de tabla) se crean en la pestaña **Tabla**. Tanto los parámetros de fórmula como los de tabla pueden ser locales o globales. Los parámetros locales son parámetros globales vinculados al componente correspondiente (fórmula o tabla) en el momento de su creación. Los parámetros locales se crean haciendo clic con el botón derecho en el componente (fórmula o tabla) y seleccionado **Agregar parámetro nuevo**. Los parámetros globales se crean haciendo clic con el botón derecho en el fondo en blanco de la pestaña y seleccionando **Agregar parámetro nuevo**. Como resultado se añade un parámetro nuevo llamado `parámetro` en la vista gráfica. Por ejemplo, la imagen siguiente muestra un parámetro global. Para cambiar el nombre del parámetro, haga doble clic en su nombre y edítelo.

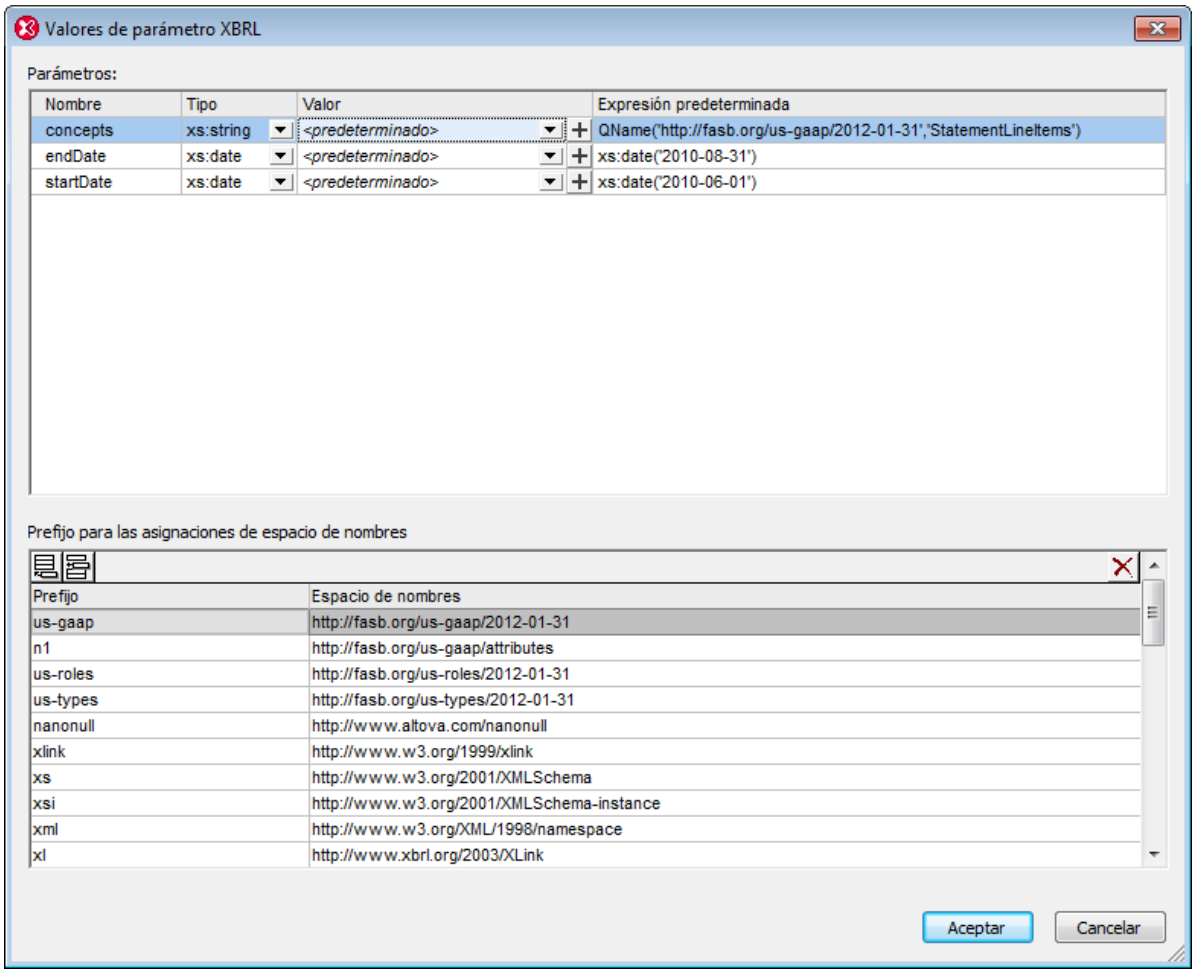


Todos los parámetros tienen la marca `Required` (*Obligatorio*). Si tiene establecida esta marca, el parámetro es obligatorio, es decir, su valor debe ser aportado por la aplicación de procesamiento. Si el parámetro no es obligatorio y la aplicación de procesamiento no aporta ningún valor, entonces el valor dado se puede calcular usando la expresión XPath dada en la propiedad `Select`. Haga doble clic en el campo `Select` para escribir una expresión XPath. Este valor será el valor predeterminado del parámetro. La propiedad opcional `As` indica el tipo de datos que exige el parámetro. Haga clic en este campo y elija un tipo de datos en la lista desplegable.

En el caso de los parámetros que se usarán como parámetros de tabla, puede editar el tipo de datos del parámetro y aportar un valor de parámetro que reemplace el valor predeterminado. Esto se hace de la siguiente manera:

1. Haga clic en **XBRL | Valores de parámetro** para abrir el cuadro de diálogo "Valores de parámetro" (*imagen siguiente*).

© 2016-2022 Altova GmbH



2. Ahora introduzca un valor de parámetro. Este valor reemplazará el valor predeterminado.

   Como los parámetros que se utilizan como parámetros de tabla pueden tomar varios valores, puede añadir varios valores para un mismo parámetro. Esto se hace con el icono **+** de la columna *Valor*.

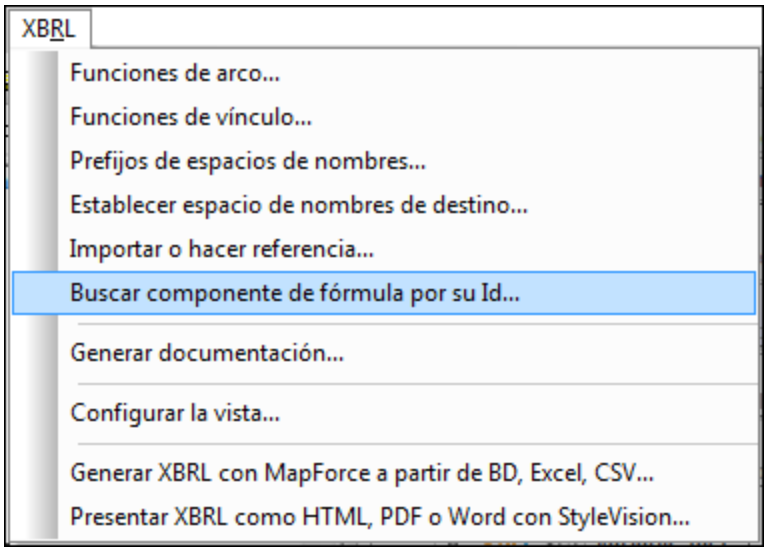
Los valores de los parámetros globales asignados en este cuadro de diálogo solamente se evalúan para los parámetros de tabla. Los valores de parámetro utilizados en las fórmulas no se pueden editar en este cuadro de diálogo.

17.4.6 Buscar componentes de fórmulas

Puede buscar componentes de fórmulas por su identificador o recorriendo el documento.

Buscar componentes de fórmula por su identificador

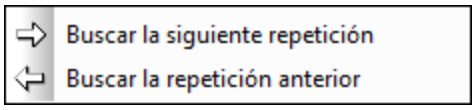
En las taxonomías con bases de enlace de fórmula de gran tamaño que contienen varios componentes del mismo tipo, quizás la mejor opción sea buscar los componentes por su identificador. Para ello puede utilizar el comando de menú **XBRL | Buscar componente por su Id**.



Al hacer clic en este comando se abre un cuadro de diálogo donde puede insertar el Id. que desea buscar.

Buscar repeticiones del componente

La mayoría de los componentes de fórmula se presentan varias veces dentro del diagrama de la base de enlaces de fórmula: (i) en la definición, que está justo debajo del nodo de sección correspondiente y (ii) en todas las referencias al componente (por medio de relaciones). Los comandos **Buscar la siguiente repetición** y **Buscar la repetición anterior** del menú contextual del componente (*imagen siguiente*) sirven para navegar hasta dónde se haga referencia al componente.



Estos comandos también están disponibles en la barra de herramientas (*imagen siguiente*).



Al llegar a la definición del componente aparece un mensaje a tal efecto.

© 2016-2022 Altova GmbH

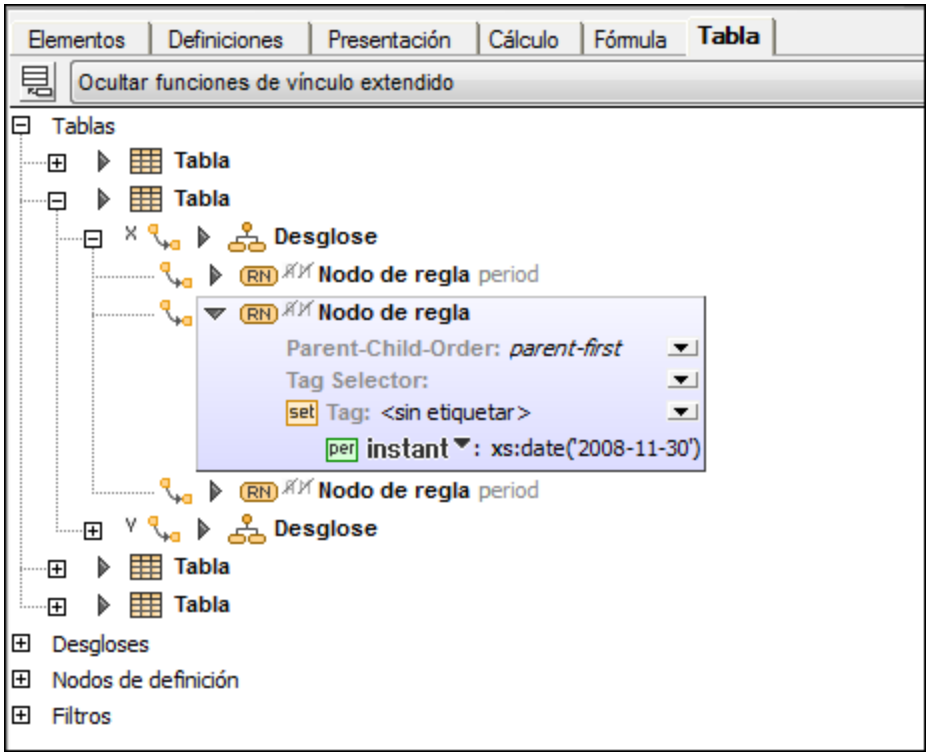
17.5 Editor de definiciones de tabla XBRL

Las especificaciones XBRL permiten usar una base de enlaces de tabla como complemento de la base de enlaces de presentación. Las tablas permiten definir de otra manera las vistas de los conceptos definidos en taxonomías XBRL. En lugar de mostrar los conceptos de forma jerárquica (como hace la base de enlaces de presentación), la base de enlaces de tabla permite definir tablas con varios ejes. Los componentes de los ejes no tienen por qué ser elementos individuales, sino que se pueden definir combinando dimensiones, referencias a períodos de tiempo, unidades, entidades o cualquier otra propiedad que se pueda utilizar para identificar los hechos financieros representados por las taxonomías. En XBRL Table Linkbase Overview 1.0 y en la recomendación Table Linkbase 1.0 del 18 de marzo de 2014 puede encontrar un resumen de la sintaxis y la semántica de las bases de enlaces de tabla XBRL.

XMLSpy sigue la recomendación Table Linkbase 1.0 del 18 de marzo de 2014 y utiliza el espacio de nombres http://xbrl.org/2014/table.

Las bases de enlaces XBRL estándar (presentación, cálculo, definición) definen relaciones entre los conceptos mediante localizadores y arcos estándar en vínculos extendidos estándar. Por el contrario, las bases de enlaces de tabla contienen componentes (tablas, desgloses, nodos de definición, etc.) y sus relaciones a través de recursos y arcos genéricos en vínculos extendidos genéricos. La especificación XBRL Table Linkbase define una secuencia de tres modelos y procesos para transformar cada modelo en el siguiente. Los tres modelos son: el modelo de definición, el modelo estructural y el modelo de diseño (o presentación). El modelo de definición es el modelo del contenido semántico de la base de enlaces de tabla. Las tablas se definen por sus ejes y las definiciones de eje están compuestas a su vez por árboles de nodos de definición.

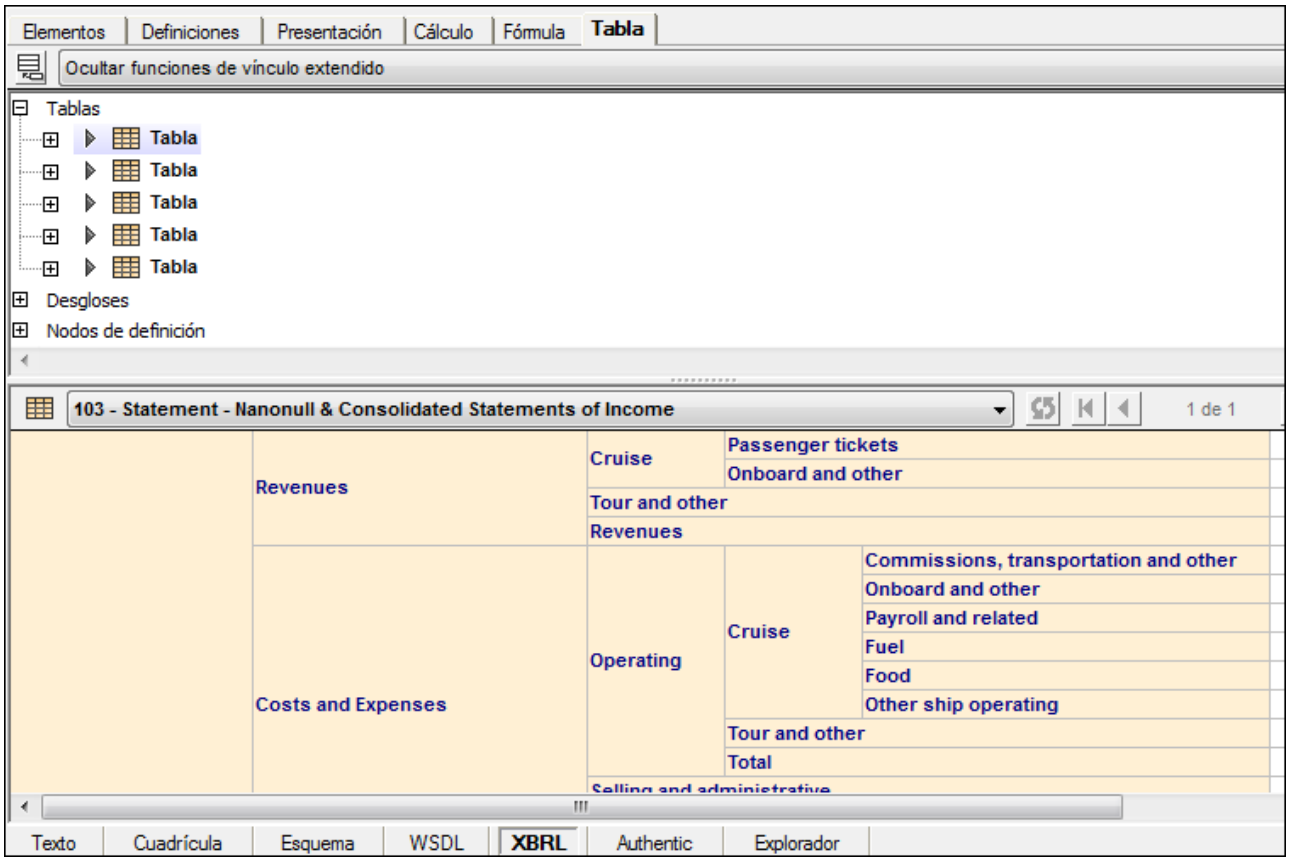
El editor de definiciones de tabla XBRL de XMLSpy está implementado en el editor de taxonomías XBRL y está disponible en la pestaña **Tabla** de la vista **XBRL** (*imagen siguiente*).



Puede crear y editar definiciones de tabla en la pestaña **Tabla** y con ayuda de los ayudantes de entrada Vista general y Detalles. En el ayudante de entrada Vista general puede definir la base de enlaces predeterminada para las tablas XBRL (es decir, el archivo predeterminado donde se guardan las definiciones de tabla), mientras que en el ayudante de entrada Detalles puede editar las propiedades y el contenido de los componentes de tabla. En la pestaña **Tabla** propiamente dicha puede editar las definiciones de tabla.

Vista previa del diseño de la tabla XBRL

El editor de taxonomías XBRL ofrece una práctica función que genera una vista previa del diseño de una definición de tabla. Se trata del panel de vista previa del diseño de la tabla XBRL de la pestaña **Tabla** de la vista **XBRL** (*imagen siguiente*). Cuando seleccione una tabla o componente de tabla en la vista gráfica de la pestaña, el panel inferior genera una vista previa del diseño de la tabla. Si lo prefiere, también puede seleccionar una tabla en la lista del cuadro combinado del panel de vista previa. Esta lista incluye todas las tablas de la base de enlaces de tabla.



Para más información sobre esta función de vista previa consulte estos apartados:

- Estructura de las tablas 892
- Parámetros de tablas 912
- Vista previa del diseño de las tablas 916

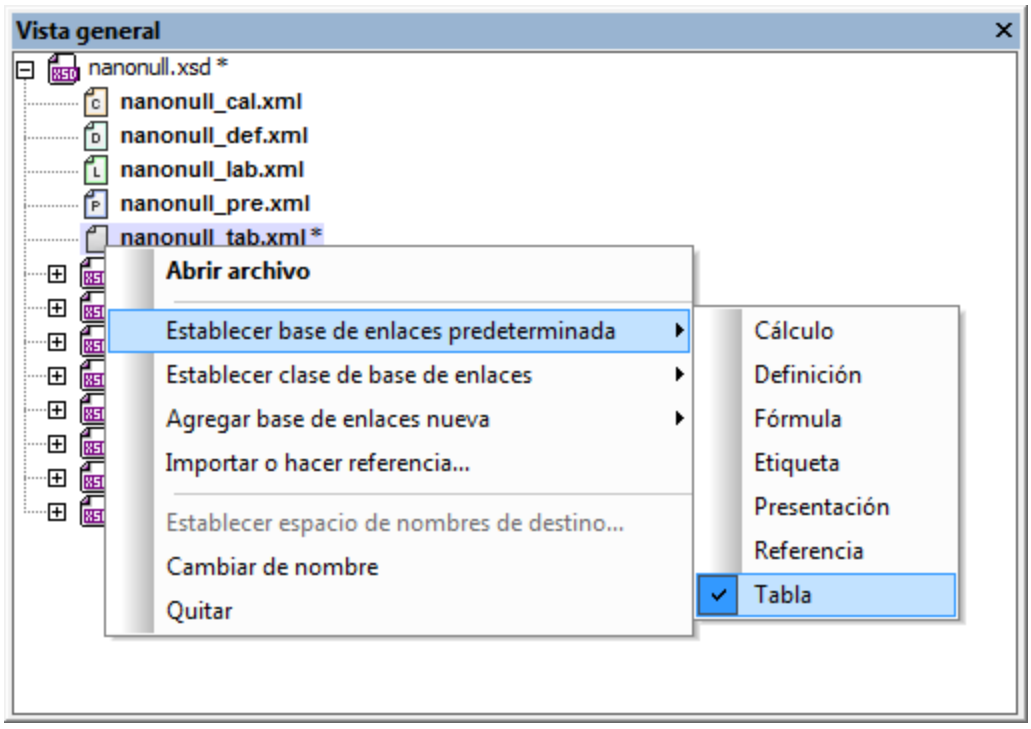
© 2016-2022 Altova GmbH

17.5.1 Bases de enlaces y funciones de vínculo de tablas

Las bases de enlaces XBRL estándar (definición, presentación, cálculo) definen relaciones entre los conceptos mediante localizadores y arcos estándar en vínculos extendidos estándar. Por el contrario, las bases de enlaces de tabla contienen componentes (tablas, desgloses, nodos de definición, etc.) y sus relaciones. Estas definiciones se especifican mediante recursos y arcos genéricos en vínculos extendidos genéricos.

Agregar una base de enlaces de tabla

En el ayudante de entrada Vista general (*imagen siguiente*), haga clic con el botón derecho en el archivo de taxonomía o en una de las base de enlaces disponibles y elija **Agregar base de enlaces nueva | Tabla**. La base de enlaces recién añadida se convierte en la base de enlaces de tabla predeterminada. El archivo de base de enlaces de tabla predeterminado es el archivo en el que se guardan las definiciones de tabla nuevas cuando se guarda el archivo de taxonomía. Si quiere usar otro archivo de base de enlaces de tabla como archivo predeterminado, haga clic con el botón derecho en ese archivo y elija **Establecer base de enlaces predeterminada | Tabla** (*imagen siguiente*).



Recuerde que las bases de enlaces predeterminadas se resaltan en negrita y que las bases de enlaces modificadas y sin guardar se marcan con un asterisco.

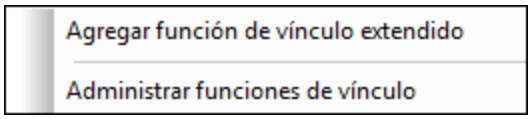
La base de enlaces de tabla se abre en la pestaña **Tabla**.

Nota: si no existe una base de enlaces de tabla todavía y añade un componente de tabla a la taxonomía 905, XMLSpy crea una base de enlaces de tabla automáticamente.

Funciones de vínculo

Al igual que los vínculos extendidos estándar (para definiciones, presentaciones y cálculos), los vínculos genéricos deben definir un valor de función de vínculo extendido que divida las relaciones del mismo tipo en redes inconexas. Todos los vínculos extendidos genéricos que tengan la misma función de vínculo se consolidan bajo un nodo de función de vínculo en el diagrama de la pestaña **Tabla**, incluso si residen en archivos de base de enlaces distintos.

Las funciones de vínculo genéricos se pueden crear en la pestaña **Tabla** con el menú contextual (*imagen siguiente*). No obstante, recuerde que este menú contextual aparece solamente si la pestaña **Tabla** está en el modo **Mostrar todas las funciones de vínculo extendido**.



Este menú también aparece al hacer clic en el icono  de la barra de herramientas de la pestaña **Tabla**. Como las redes de relaciones no son tan importantes para una base de enlaces de tabla, el modo predeterminado de la pestaña **Tabla** es **Ocultar funciones de vínculo extendido**. Este modo oculta las funciones de vínculo y muestra los componentes de tabla sin sus funciones de vínculo.

Si al crear la función de vínculo extendido no existe un archivo de base de enlaces de tabla predeterminado, XMLSpy crea uno automáticamente. Si al crear la función de vínculo extendido no existe ninguna función de vínculo extendido en el archivo de base de enlaces de tabla predeterminado, XMLSpy crea una automáticamente.

17.5.2 Estructura de las tablas

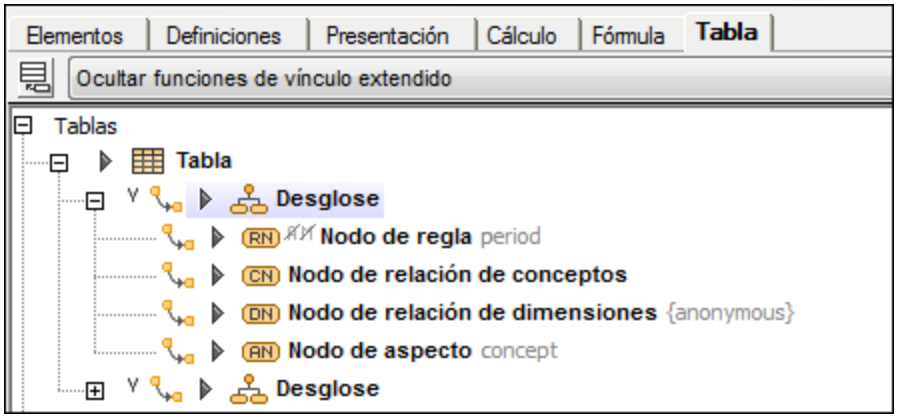
La estructura de una tabla en la definición de tabla viene dada por los ejes de la tabla (X,Y,Z) y cada uno de estos ejes corresponde a uno o varios componentes de tipo desglose (*imagen siguiente*).

- Los ejes X e Y[894] corresponden respectivamente a las columnas y a las filas de la tabla generada.
- Si se definió un eje Z[903], se presenta como una tabla distinta.
- Cada componente de tipo desglose puede incluir varios nodos de definición de tabla (*imagen siguiente*). Hay varios tipos de nodos de definición:

  - nodos de regla (icono RN en la imagen)
  - nodos de relación de conceptos (CN)
  - nodos de relación de dimensiones (DN) y
  - nodos de aspecto (AN).

Para más información sobre las propiedades estructurales de estos nodos de definición, consulte el apartado *Nodos de definición*[896].

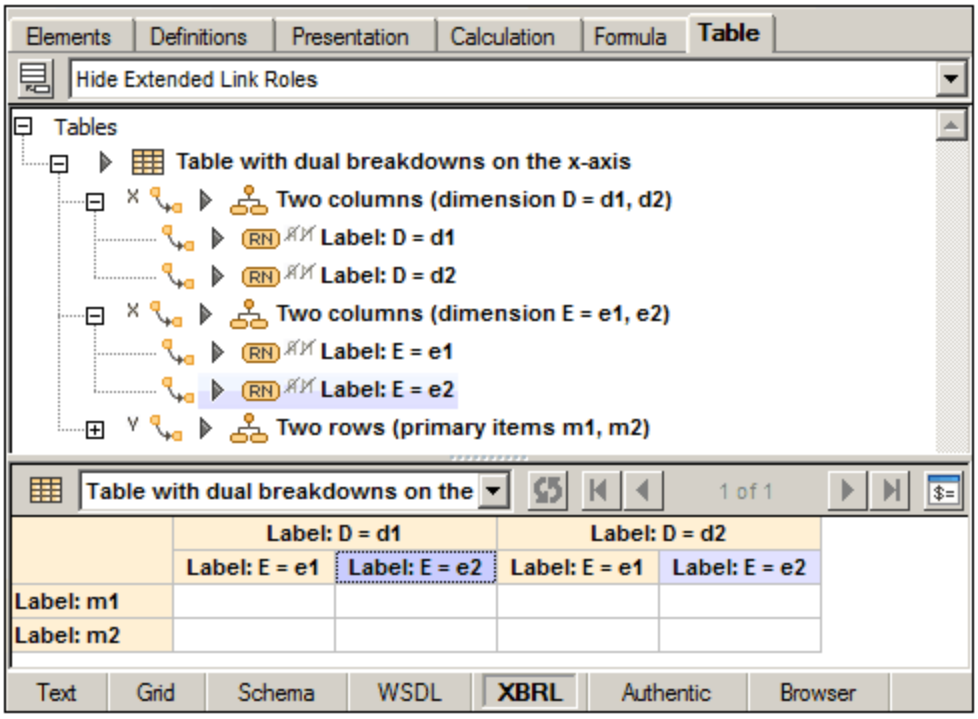
© 2016-2022 Altova GmbH



Proyecciones para varios desgloses

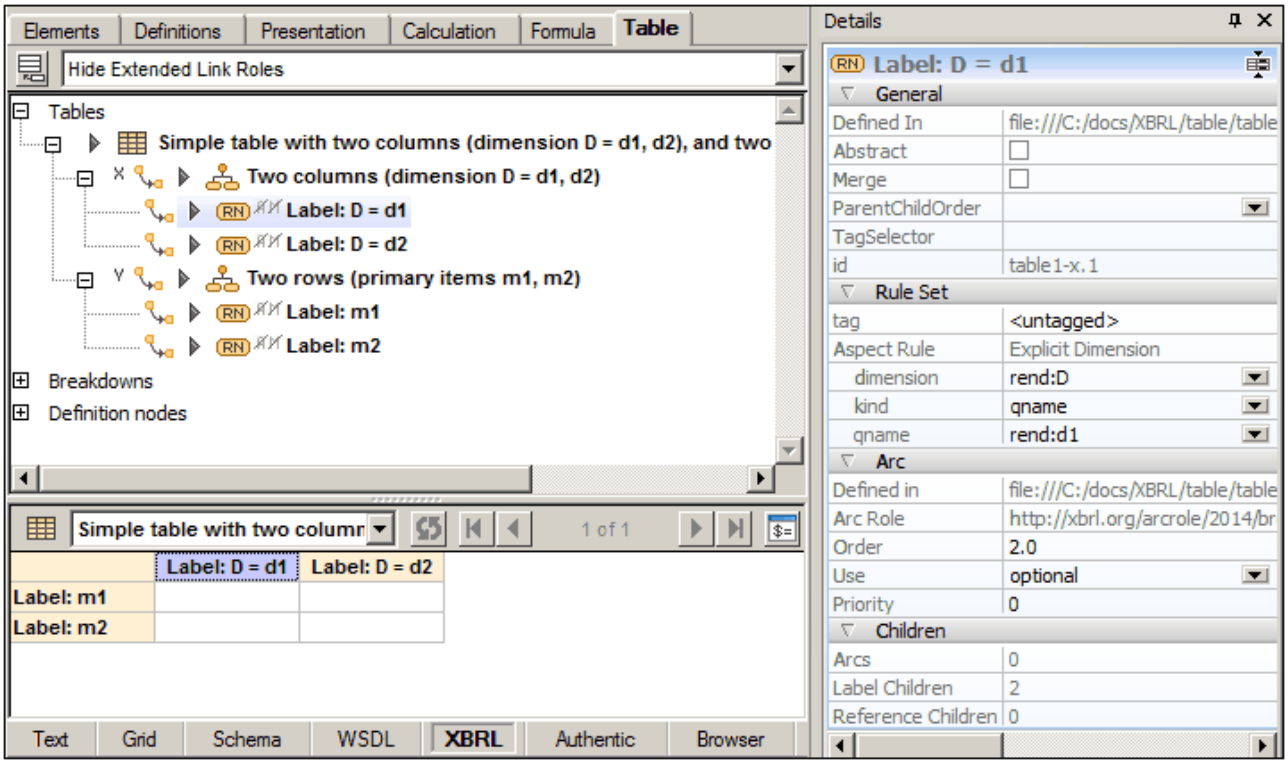
Existe la posibilidad de asociar varios desgloses independientes con un solo eje de tabla. El mecanismo para resolver cómo se combinan varios desgloses en un solo desglose "real" se denomina *proyección*. La prioridad relativa de varios desgloses asociados a un solo eje viene determinada por el atributo `@order` de cada desglose. Los desgloses se presentan en la vista gráfica en forma de árbol. Por cada rama del primer desglose, se anexa recursivamente el segundo desglose entero, y así sucesivamente.

En la imagen siguiente, por ejemplo, hay dos desgloses para el eje X: **`dimension D`** tiene mayor prioridad que **`dimension E`**. Así que, por cada rama de **`dimension D`** (**`d1`** y **`d2`**) se anexa todo el árbol de **`dimension E`**. Como el eje X genera columnas, estos desgloses crean una proyección para la estructura de columnas de la tabla. Esto puede observarse en la vista previa de la imagen siguiente.



17.5.2.1 Eje X y eje Y

El eje X y el eje Y determinan las columnas y las filas de la tabla respectivamente. Por cada eje se define un desglose jerárquico como mínimo (*imagen siguiente*). Los desgloses que corresponden a un solo eje dan como resultado un solo desglose "real". Si solamente hay un desglose para un eje, ese desglose será el eje "real". Si se definieron varios desgloses para un eje, el método de resolución se describe más abajo 894.



Tenga en cuenta estas propiedades y características de edición propias de los ejes:

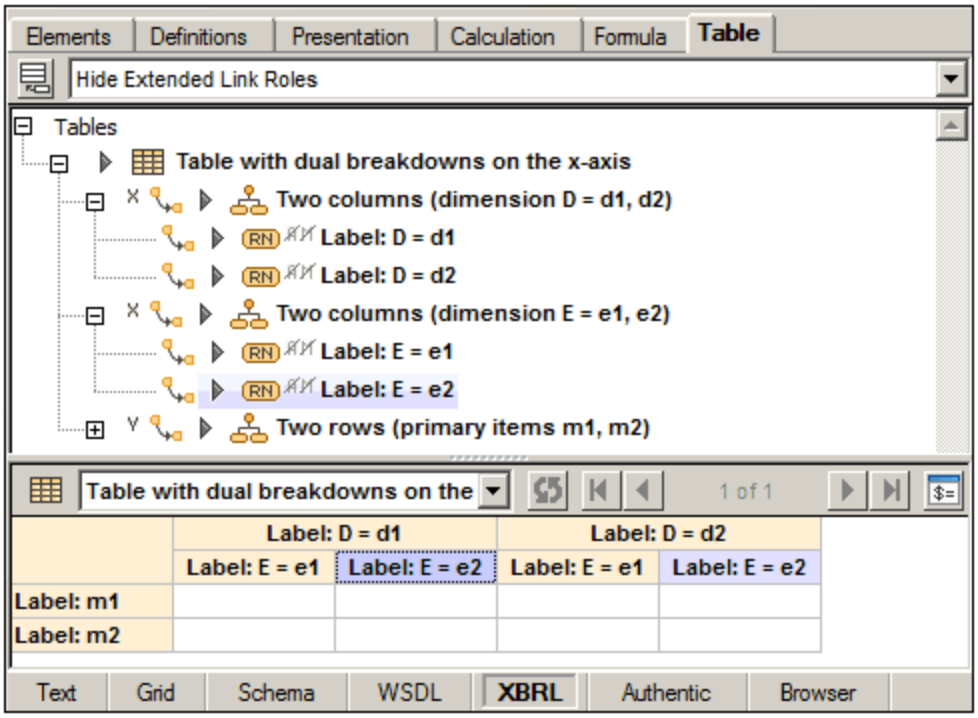
- En las definiciones de tablas el eje X corresponde a las columnas de la tabla generada, mientras que el eje Y corresponde a las filas de la tabla (*imagen siguiente*).
- Cada eje puede tener un desglose como mínimo (consulte el último subapartado 894).
- El color de fondo de cada celda de la tabla generada es naranja/amarillo. En la definición de tabla, una celda corresponde a un nodo de definición de un desglose del eje.
- Cuando se selecciona una celda, su nodo de definición también se selecciona y viceversa. El color de fondo de las celdas que están seleccionadas es violeta.
- Cuando se selecciona un componente, sus propiedades aparecen en el ayudante de entrada Detalles, donde se pueden editar (*imagen anterior*).
- Las celdas de datos no tienen ningún color de fondo. Siempre están vacías porque la taxonomía propiamente dicha no contiene hechos.
- Las restricciones de celdas se calculan a partir de los ejes (usando selectores de etiqueta, si los hay) y aparecen en la pestaña *Restricciones* del ayudante de entrada Detalles. En el apartado Eje Z 903 puede ver un ejemplo.

© 2016-2022 Altova GmbH

Proyecciones para varios desgloses

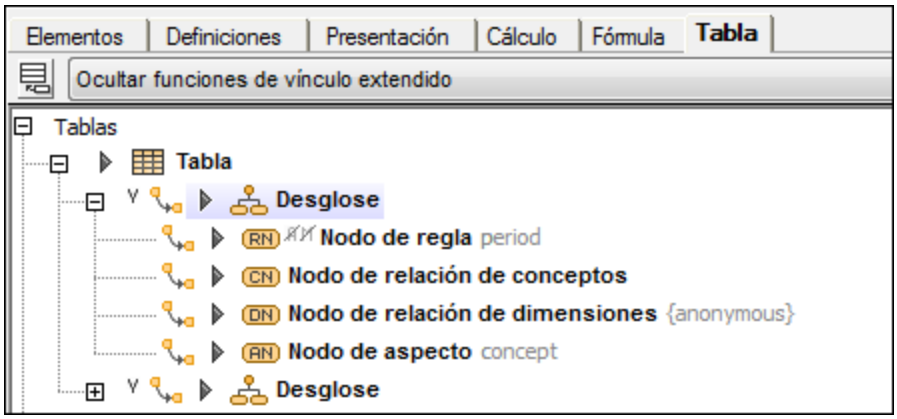
Existe la posibilidad de asociar varios desgloses independientes con un solo eje de tabla. El mecanismo para resolver cómo se combinan varios desgloses en un solo desglose "real" se denomina *proyección*. La prioridad relativa de varios desgloses asociados a un solo eje viene determinada por el atributo `@order` de cada desglose. Los desgloses se presentan en la vista gráfica en forma de árbol. Por cada rama del primer desglose, se anexa recursivamente el segundo desglose entero, y así sucesivamente.

En la imagen siguiente, por ejemplo, hay dos desgloses para el eje X: **`dimension D`** tiene mayor prioridad que **`dimension E`**. Así que, por cada rama de **`dimension D`** (**`d1`** y **`d2`**) se anexa todo el árbol de **`dimension E`**. Como el eje X genera columnas, estos desgloses crean una proyección para la estructura de columnas de la tabla. Esto puede observarse en la vista previa de la imagen siguiente.



17.5.2.2 Nodos de definición

Cada componente de tipo desglose puede incluir varios nodos de definición de tabla (*imagen siguiente*).



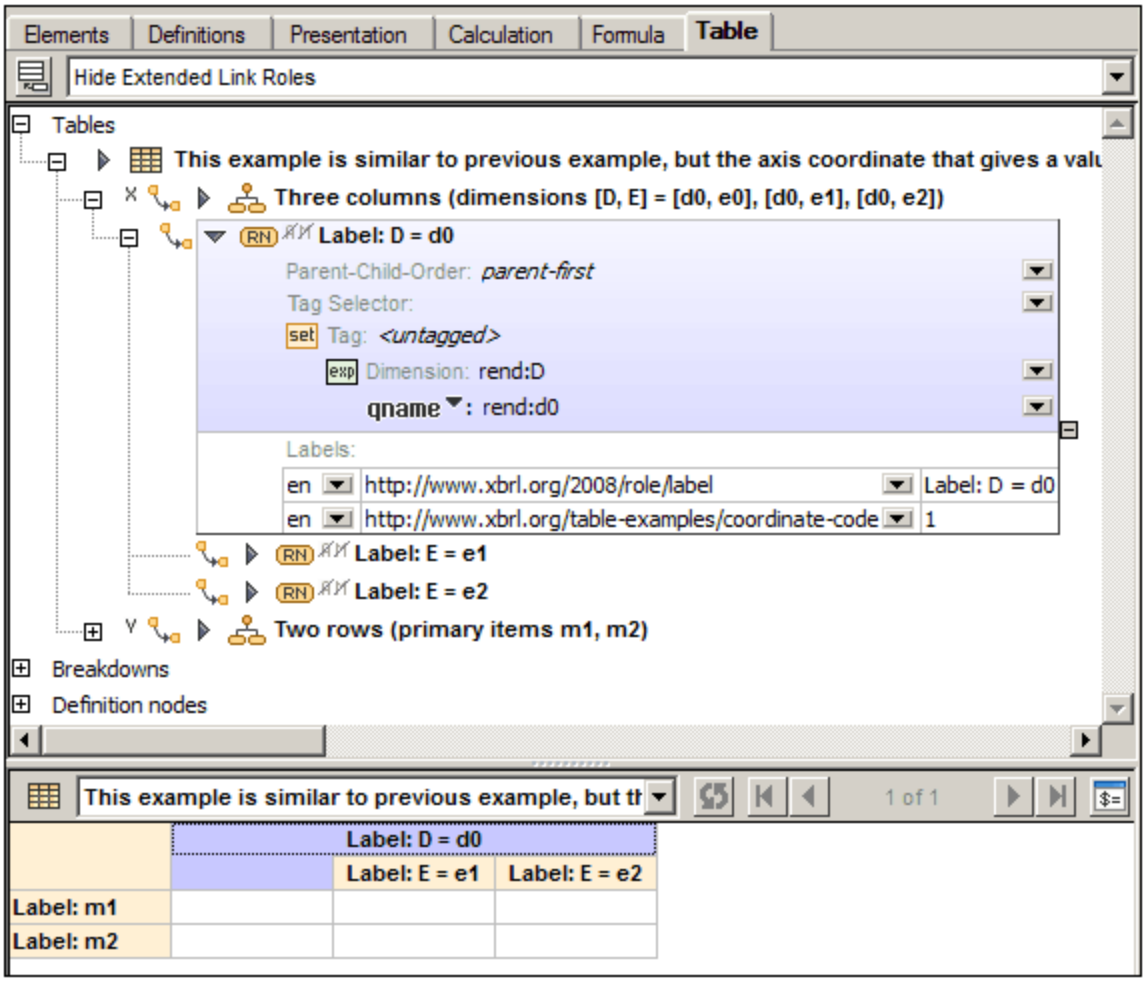
Hay varios tipos de nodos de definición:

- Nodos de regla[896] (icono **RN** en la imagen anterior)
- Nodos de relación de conceptos[900] (**CN**)
- Nodos de relación de dimensiones[900] (**DN**)
- Nodos de aspecto[903] (**AN**)

17.5.2.2.1 Nodos de regla

Un nodo de regla define reglas de aspecto para uno o varios aspectos: `concept`, `period`, `unit`, `entity identifier`, `dimension` u `open content`. El componente de la estructura de la definición se corresponde con una celda del diseño si el nodo de regla es de tipo abstracto y no tiene secundarios. De lo contrario, el diseño contiene una celda "acumulada" más cuya posición viene dada por el valor real de la propiedad `parentChildOrder` del nodo de regla.

© 2016-2022 Altova GmbH

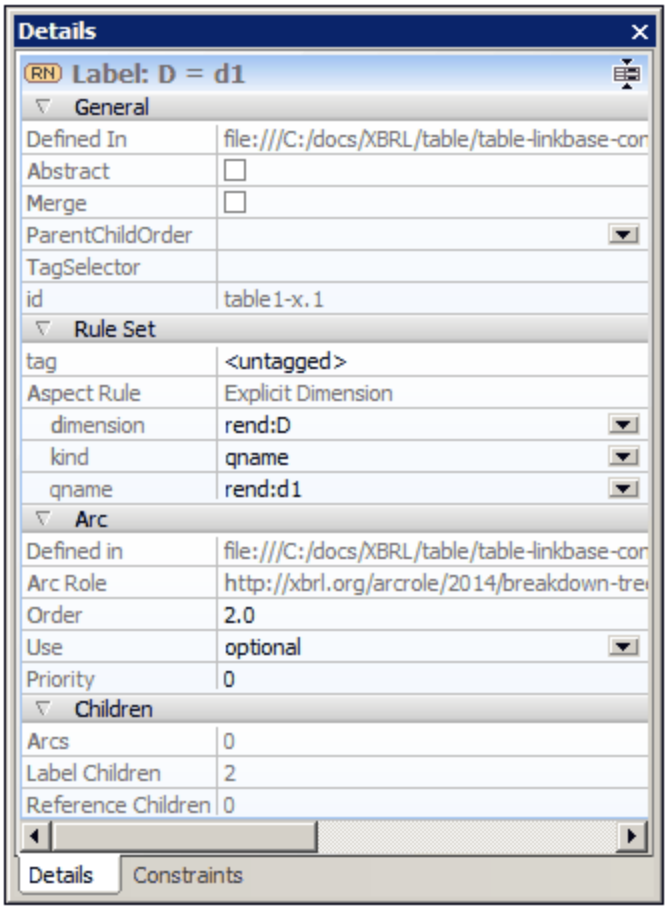


El encabezado de la celda del diseño se calcula a partir del nodo de regla de la siguiente manera:

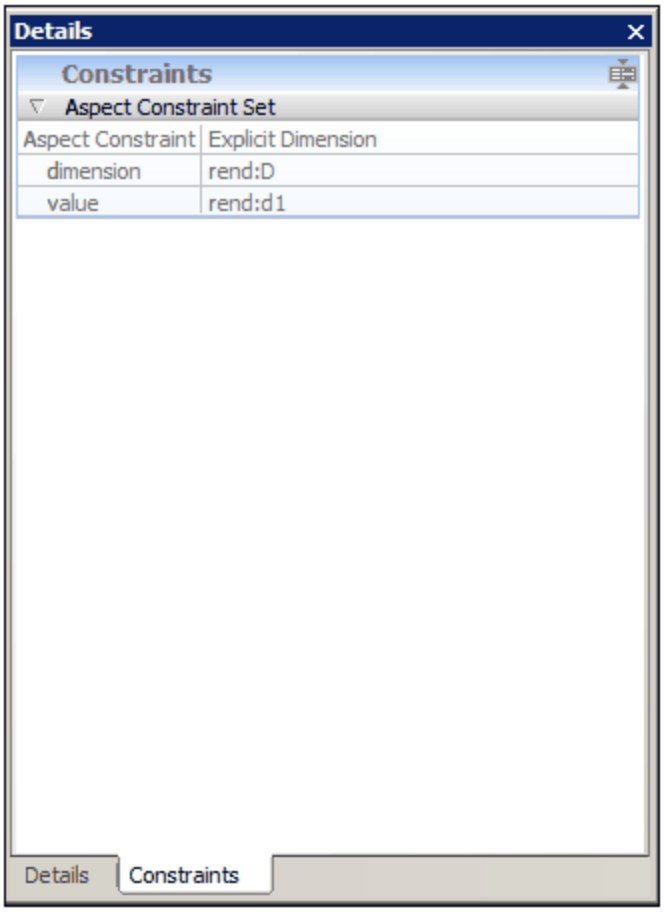
- Si el nodo está asociado con una etiqueta definida por el usuario, se muestra el texto de esta etiqueta.
- Si falta la etiqueta pero el nodo define una sola restricción de aspecto (`concept`, `dimension`, `unit`, `entity-identifier` o `period`), su valor aparece (por ejemplo, el nombre completo del concepto).
- De lo contrario se utiliza el texto estático `Nodo de regla`.

Ayudante de entrada Detalles

Las propiedades del nodo de definición aparece en la pestaña *Detalles* del ayudante de entrada Detalles (*imagen izquierda*). La pestaña *Restricciones* (*imagen derecha*) ofrece una vista de solo lectura de los conjuntos de restricciones de aspecto que se calculan a partir de las reglas de aspecto del nodo de regla.

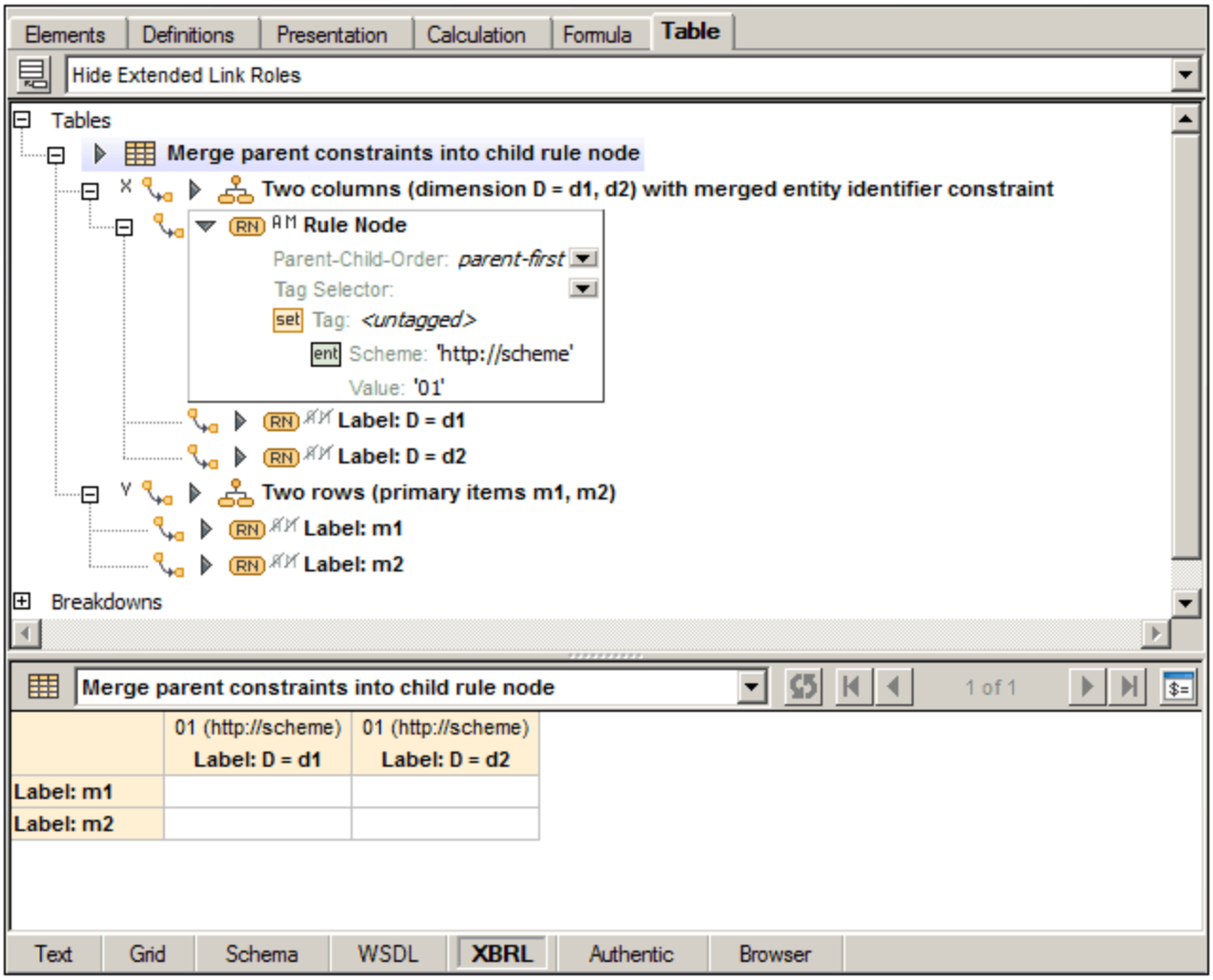


© 2016-2022 Altova GmbH



Nodos de regla combinados

Un nodo de regla combinado indica propiedades adicionales que afectan a todos sus secundarios, es decir, contribuye todas sus restricciones a cada conjunto de restricciones que producen sus secundarios (*imagen siguiente*).

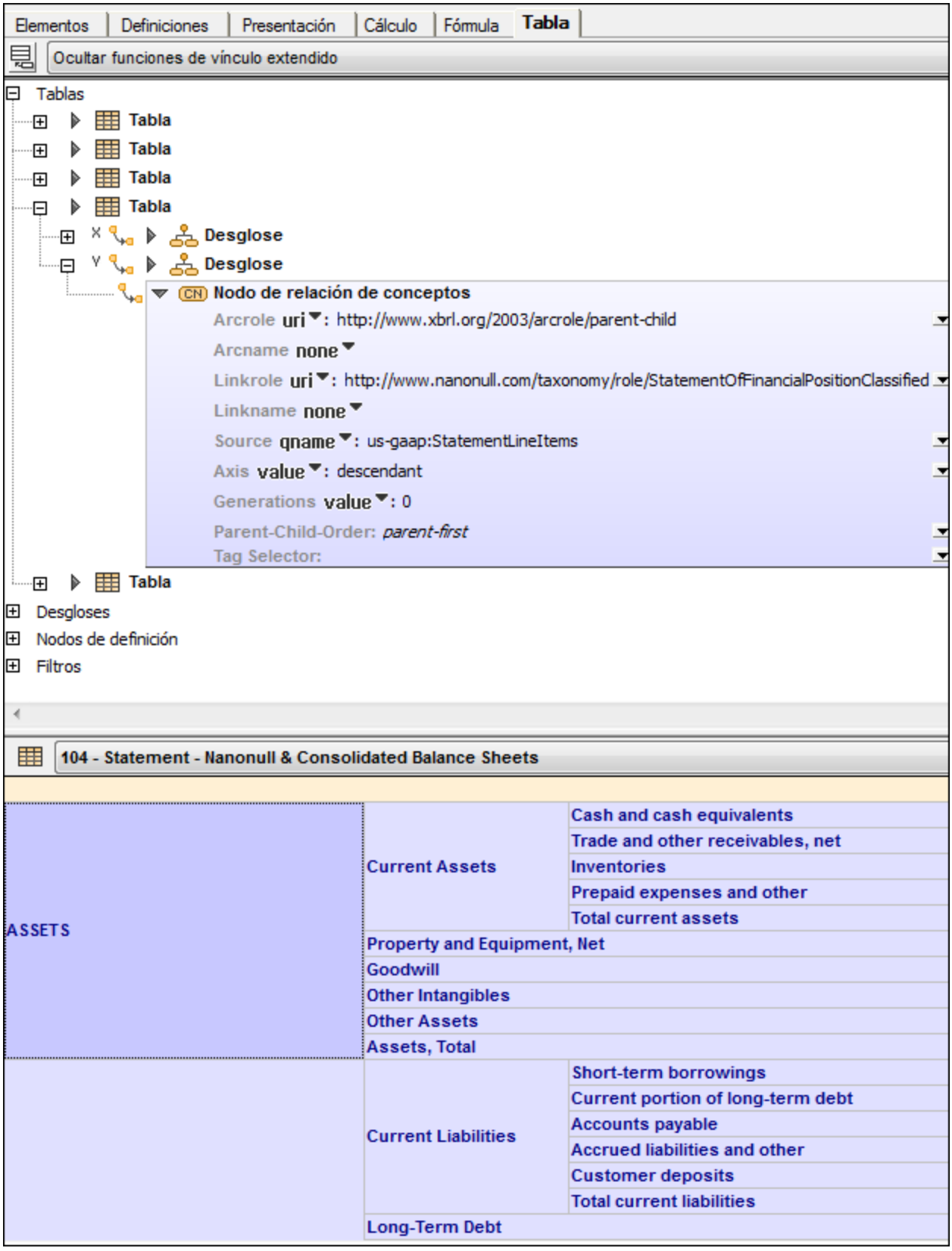


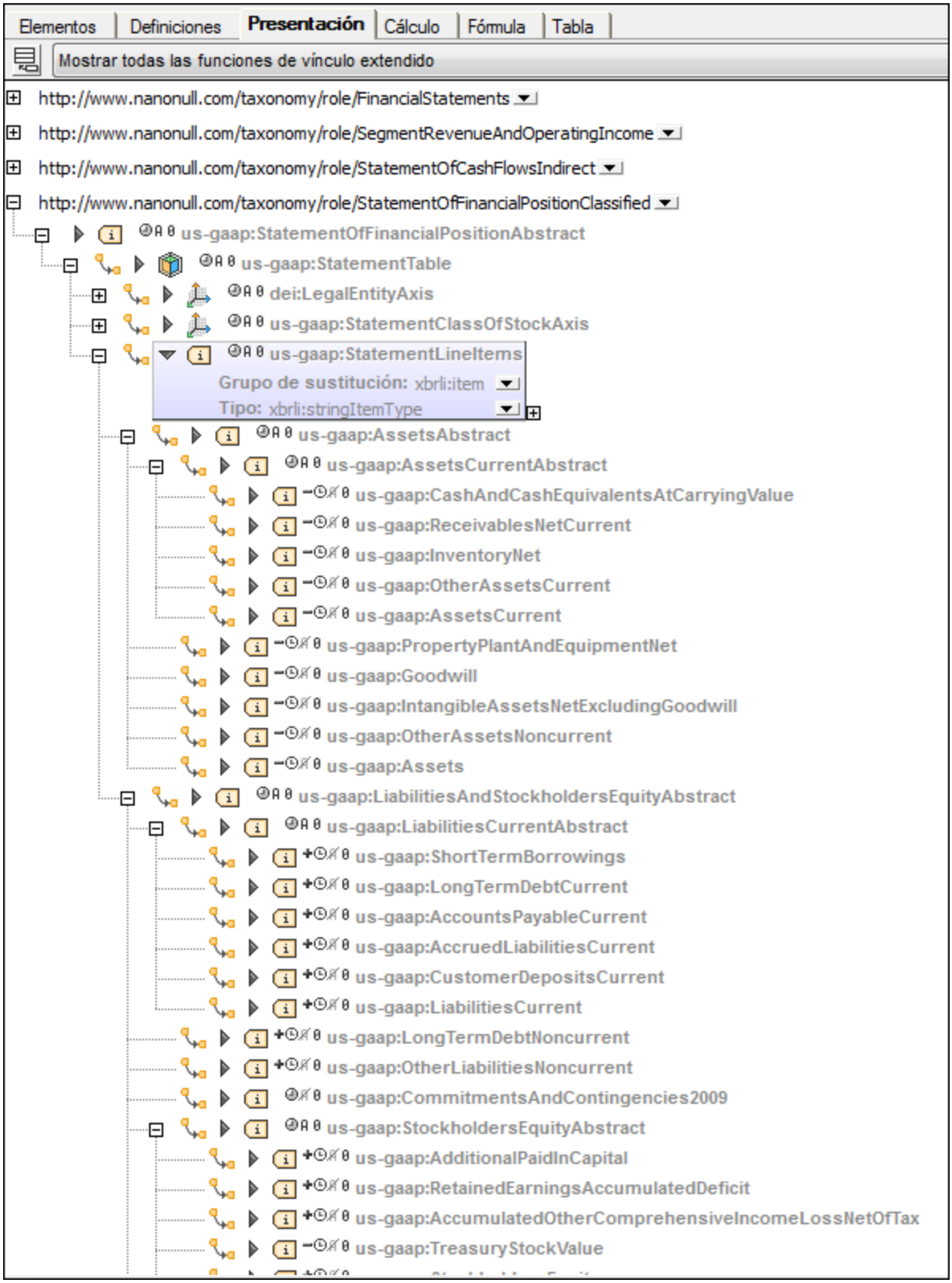
17.5.2.2.2 Nodos de relación

Un nodo de relación de dimensiones o de relación de conceptos da como resultado un árbol de nodos estructurales, definidos por las redes de conceptos o miembros de dimensión explícita de la DTS. Por tanto, el componente del árbol de definiciones corresponde a un bloque de celdas en el diseño.

La imagen izquierda muestra una definición de tabla con un nodo de relación de conceptos. La de la derecha muestra la red correspondiente de conceptos, en este caso definidos en la base de enlaces de presentación.

© 2016-2022 Altova GmbH


Elementos
Definiciones
Presentación
Cálculo
Fórmula
Tabla
Ocultar funciones de vínculo extendido
Tablas
Tabla
Tabla
Tabla
Tabla
Desglose
Desglose
Nodo de relación de conceptos
Arcrole uri: http://www.xbrl.org/2003/arcrole/parent-child
Arcname none
Linkrole uri: http://www.nanonull.com/taxonomy/role/StatementOfFinancialPositionClassified
Linkname none
Source qname: us-gaap:StatementLineItems
Axis value: descendant
Generations value: 0
Parent-Child-Order: parent-first
Tag Selector:
Tabla
Desgloses
Nodos de definición
Filtros
104 - Statement - Nanonull & Consolidated Balance Sheets
ASSETS
Current Assets
Cash and cash equivalents
Trade and other receivables, net
Inventories
Prepaid expenses and other
Total current assets
Property and Equipment, Net
Goodwill
Other Intangibles
Other Assets
Assets, Total
Current Liabilities
Short-term borrowings
Current portion of long-term debt
Accounts payable
Accrued liabilities and other
Customer deposits
Total current liabilities
Long-Term Debt

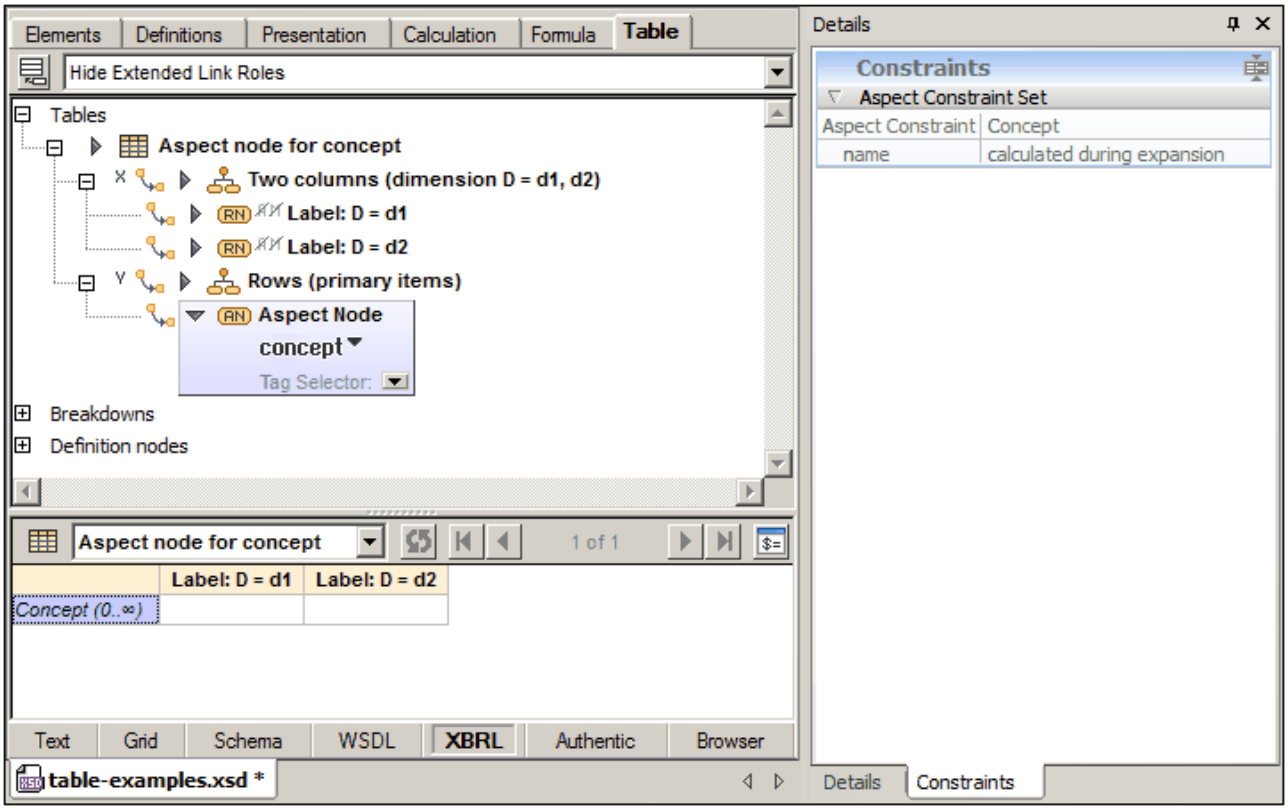


© 2016-2022 Altova GmbH

Un nodo de relación tiene exactamente una restricción de aspecto: concepto o dimensión explícita. Por tanto, el encabezado de cada celda del diseño es la etiqueta del concepto (si existe) o su nombre completo. La pestaña *Detalles* del ayudante de entrada Detalles muestra las propiendas del nodo de relación, mientras que la pestaña *Restricciones* muestra la restricción de aspecto que viene definida por la celda resaltada en el diseño.

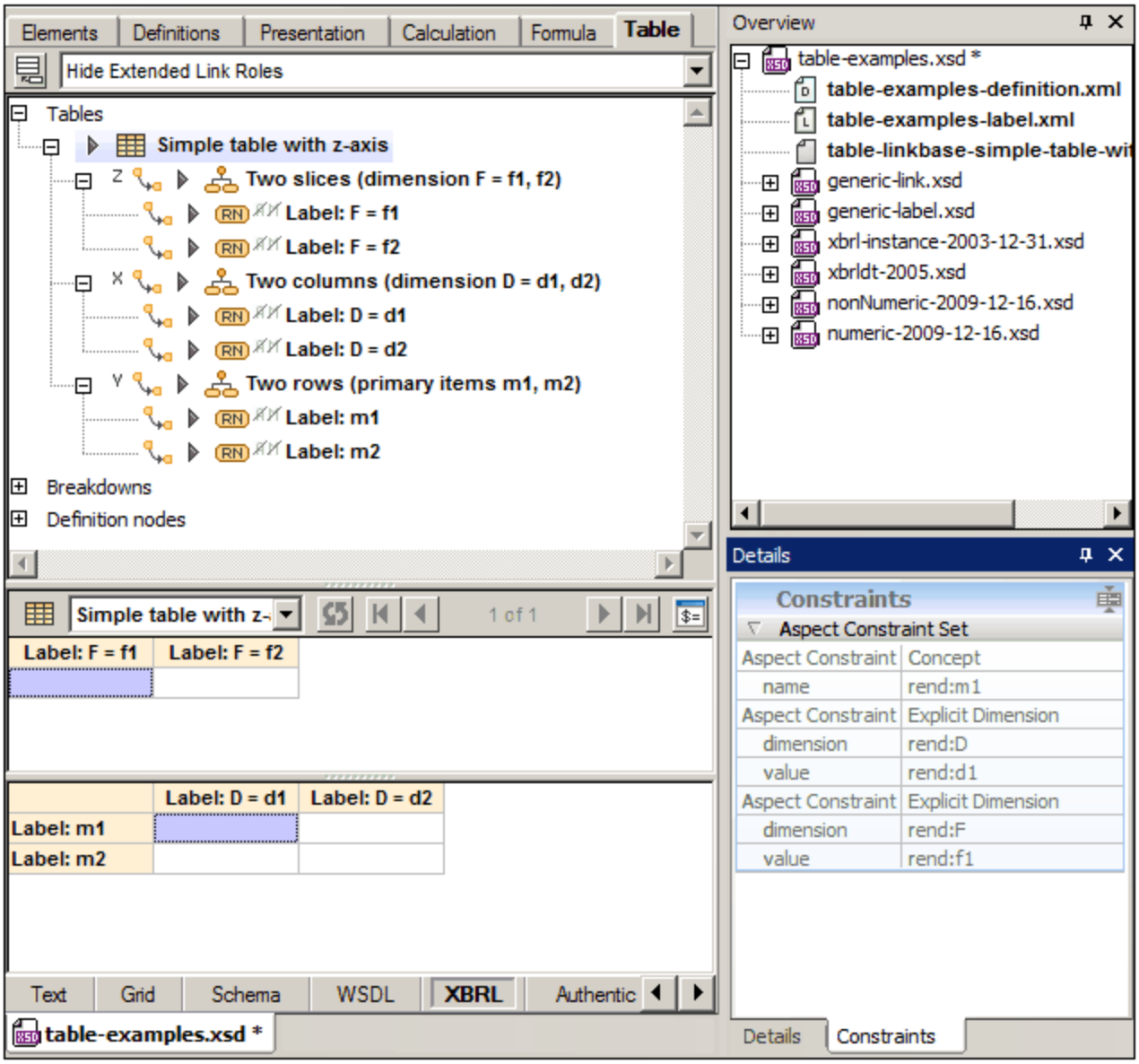
17.5.2.2.3 Nodos de aspecto

Un nodo de aspecto es un nodo de definición abierta que especifica directamente un solo aspecto activo. Durante el proceso de diseño el nodo de aspecto se expande a una celda por cada valor único de su aspecto activo que esté presente entre los hechos del archivo XBRL de instancia. Como la restricción de valor de aspecto no viene totalmente determinada por la definición del nodo y la DTS, la vista previa del diseño muestra un marcador de posición (*imagen siguiente*).



17.5.2.3 Eje Z

Si una definición de tabla contiene un eje Z, este eje se interpretará como una tabla de dos dimensiones. En el panel de vista previa del diseño de la tabla, el eje Z se representa como una tabla distinta, situada encima de la tabla XY (*imagen siguiente*).



Si una definición de tabla contiene un eje Z, en la vista previa del diseño de la tabla siempre se resaltan **dos** celdas de datos (*imagen anterior*). Así es como se especifican las coordinadas de los tres ejes: las coordinadas de los ejes X e Y en la tabla XY y la coordenada del eje Z en la tabla Z. Esto se puede observar claramente en la imagen anterior.

Las propiedades y características de edición propias del eje Z son las mismas que las de los ejes X e Y 894. Las restricciones de las celdas se calculan a partir de los ejes (por medio de selectores de etiqueta, si los hay) y se presentan en la pestaña *Restricciones* del ayudante de entrada Detalles (*imagen anterior*).

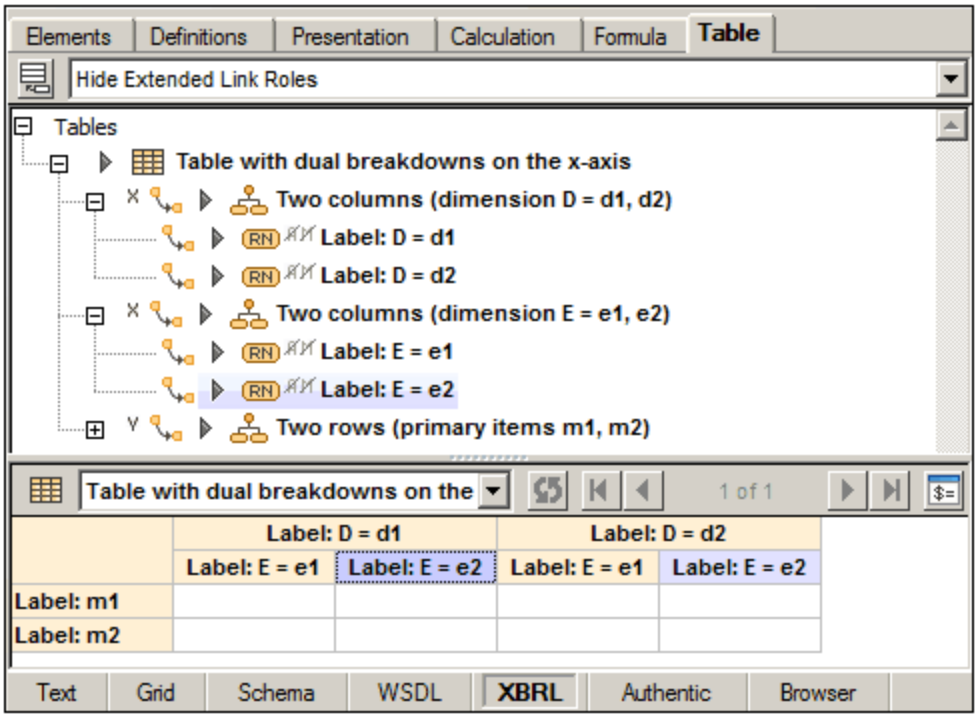
Proyecciones para varios desgloses

Existe la posibilidad de asociar varios desgloses independientes con un solo eje de tabla. El mecanismo para

 © 2016-2022 Altova GmbH

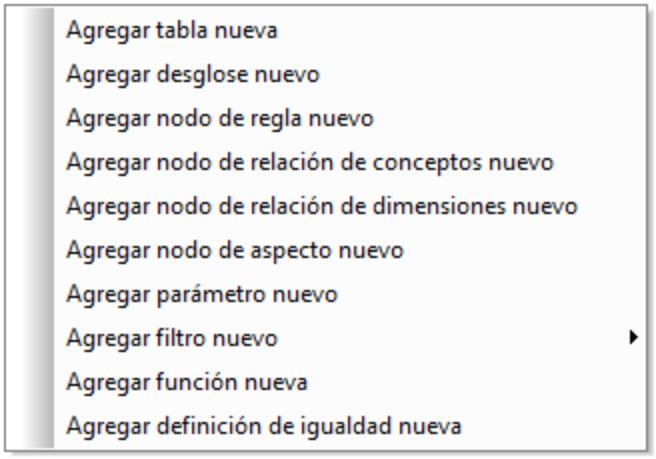
resolver cómo se combinan varios desgloses en un solo desglose "real" se denomina *proyección*. La prioridad relativa de varios desgloses asociados a un solo eje viene determinada por el atributo `@order` de cada desglose. Los desgloses se presentan en la vista gráfica en forma de árbol. Por cada rama del primer desglose, se anexa recursivamente el segundo desglose entero, y así sucesivamente.

En la imagen siguiente, por ejemplo, hay dos desgloses para el eje X: **`dimension D`** tiene mayor prioridad que **`dimension E`**. Así que, por cada rama de **`dimension D`** (**`d1`** y **`d2`**) se anexa todo el árbol de **`dimension E`**. Como el eje X genera columnas, estos desgloses crean una proyección para la estructura de columnas de la tabla. Esto puede observarse en la vista previa de la imagen siguiente.

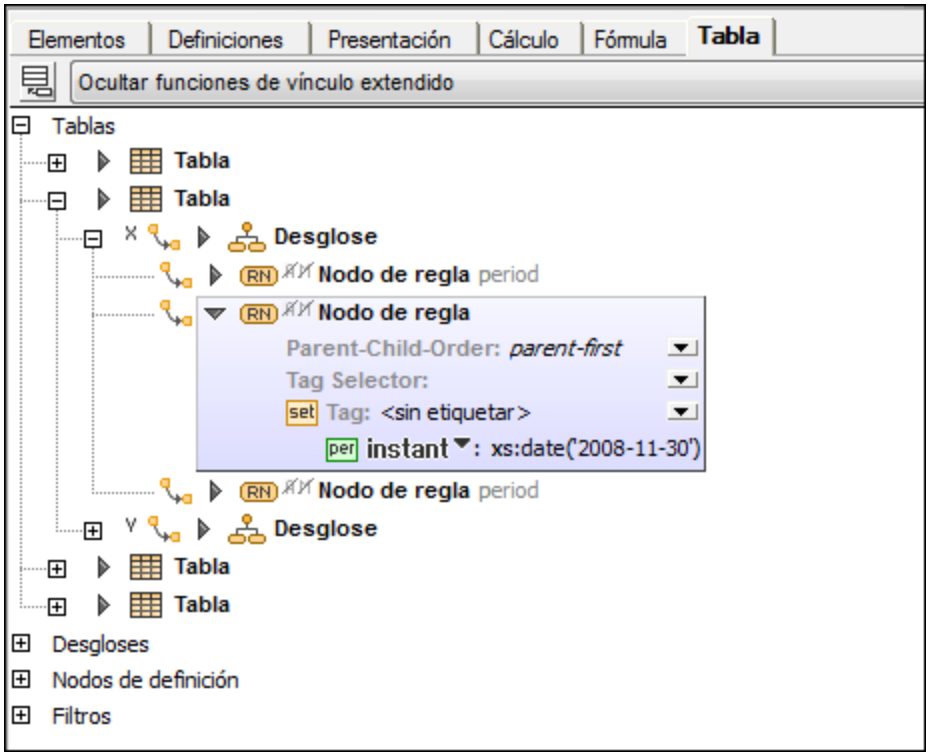


17.5.3 Componentes de tablas

Los componentes de tabla nuevos se crean con los comandos del menú contextual que aparece al hacer clic con el botón derecho en un nodo de función de vínculo (*imagen siguiente*) o con el icono de la barra de herramientas si el modo de la pestaña **Tabla** es **Ocultar funciones de vínculo extendido**.



Los apartados de esta sección incluyen instrucciones para insertar cada tipo de componente. Una vez añadido, el componente de tabla aparece en el diagrama de la pestaña **Tabla** (*imagen siguiente*).

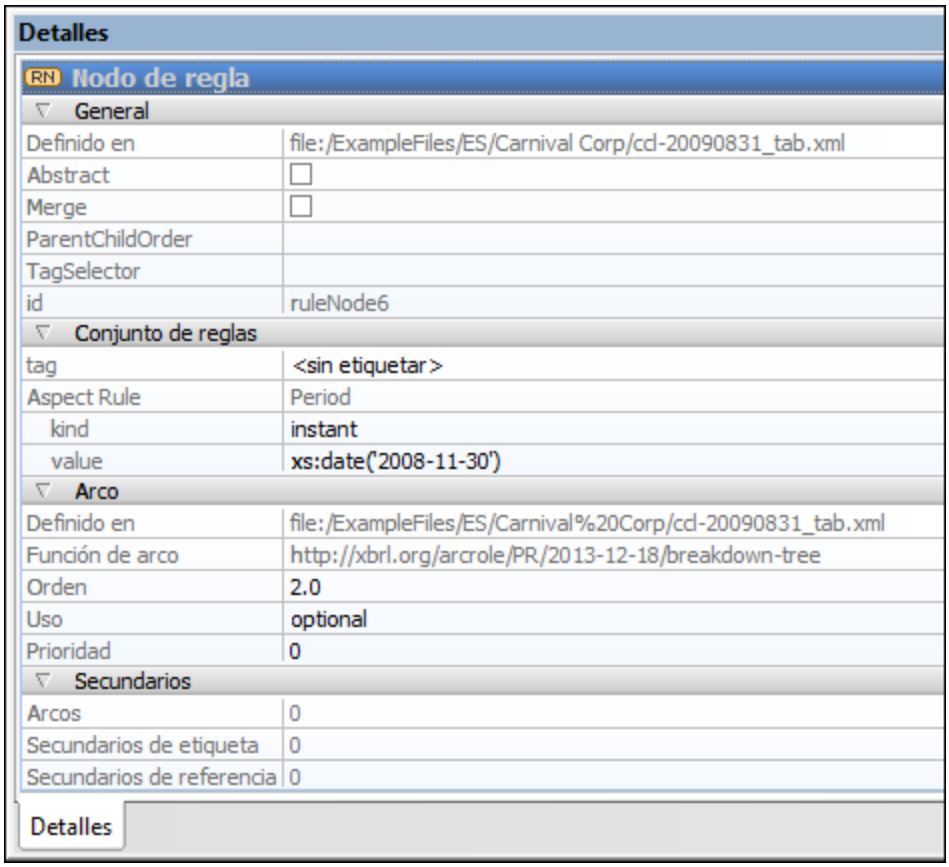


Por motivos de claridad, los componentes de tabla se dividen en secciones y sus relaciones con otros componentes (los arcos) se presentan dentro de una estructura jerárquica (*imagen anterior*). Las propiedades de los componentes y de las relaciones (arcos) se representan como iconos a la izquierda del componente o arco respectivamente (*imagen anterior*).

En la imagen siguiente puede ver el ayudante de entrada Detalles del nodo de regla que está resaltado en la imagen anterior. Las propiedades del nodo aparecen debajo de la sección *General*. Los valores de las

© 2016-2022 Altova GmbH

propiedades booleanas se definen con una casilla de verificación (si está marcada = `true`, si no = `false`). Las demás secciones del ayudante de entrada incluyen otros detalles del nodo.



Para ver las propiedades de un archivo en el ayudante de entrada Detalles, seleccione el componente `to` (destino) en el diagrama y las propiedades del arco aparecerán en la sección *Arco*.

Menús contextuales del editor de tablas

Dependiendo del tipo de componente, el editor de tablas XBRL ofrece distintos menús contextuales. Los comandos del menú contextual se organizan por grupos según su función:

- Modificar la relación (solo para subelementos): **Invalidar/Eliminar arco**
- Modificar el contenido (para nodos de regla y de relación). P. ej. **Anexar/Insertar regla de aspecto**
- **Agregar etiquetas/Referencias**
- Crear componentes secundarios nuevos (incluir relaciones). P. ej. **Agregar desglose nuevo**
- Eliminar un componente (incluidas las relaciones)
- **Buscar la siguiente repetición/la repetición anterior** (del componente)

Nota: los elementos de contenido que se pueden crear o eliminar desde el menú contextual aparecen en el ayudante Detalles en secciones adicionales (p. ej. *Conjunto de reglas*).

17.5.3.1 Tabla

Las tablas tienen la propiedad `parent-child-order` (`parent-first/children-first`). Define la posición predeterminada de los nodos de acumulación aportados por todos los nodos de definición cerrada de la tabla para los que no se reemplaza.



17.5.3.2 Desglose

Los desgloses tienen la propiedad `parent-child-order` (`parent-first/children-first`). Define la posición predeterminada de los nodos de acumulación aportados por todos los nodos de definición cerrada del desglose y reemplaza el valor heredado de la tabla.



17.5.3.3 Nodo de definición: regla

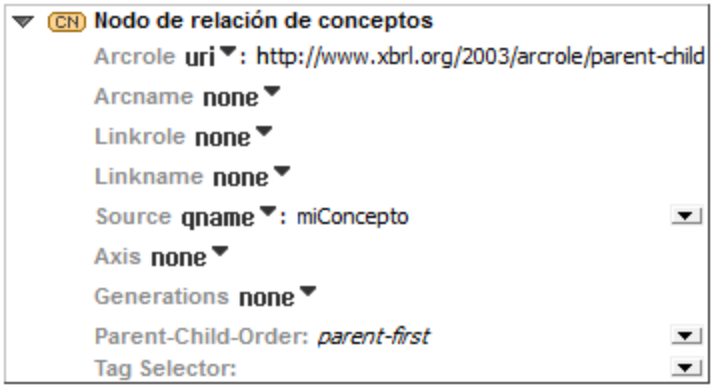
Los nodos de regla son nodos de definición cerrada que definen restricciones mediante reglas de aspecto (ver componente de fórmula). Un nodo de regla define cero o más conjuntos de reglas, es decir, conjuntos de reglas de aspectos. Cada conjunto de reglas puede especificar una etiqueta. Como máximo uno de los conjuntos de reglas puede omitir la etiqueta. Esta regla sin etiquetar aparece siempre antes de los conjuntos de reglas etiquetados. Si hay al menos un conjunto de reglas etiquetado, los conjuntos de reglas sin etiquetar vacíos no se muestran en el diagrama. Los nodos de regla tienen dos propiedades booleanas: `abstract` y `merge`, que se representan con iconos distintos en el diagrama. La imagen siguiente, por ejemplo, muestra un nodo de regla sin reglas de aspecto.



© 2016-2022 Altova GmbH

17.5.3.4 Nodo de definición: relación de conceptos

Los nodos de relación de conceptos detectan conceptos realizando un examen del árbol de una red XBRL 2.1. El examen del árbol los identifica de forma exclusiva por la red y por los orígenes de las relaciones. Un nodo de relación de conceptos debe identificar una sola red. En la mayoría de los casos, la combinación de la función de vínculo y de la función de arco basta para identificar la red de forma exclusiva. Sin embargo, a veces es necesario indicar más datos como el nombre del arco o el nombre del vínculo extendido.

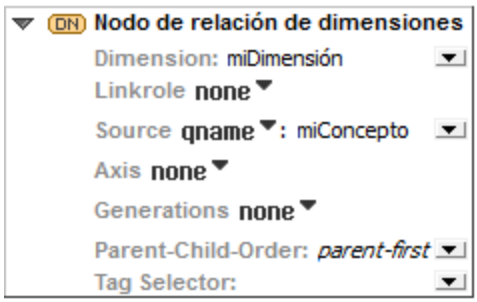



```
Arcrole:          clase = uri | exp
Arcname:          clase = none | qname | exp
Linkrole:         clase = none | uri | exp
Linkname:         clase = none | qname | exp
Source:                   clase = qname | exp
Axis:                     clase = none | value | exp
Generations:  clase = none | value | exp
```

Los nodos de relación de conceptos no pueden tener árboles subordinados.

17.5.3.5 Nodo de definición: relación de dimensiones

Los nodos de relación de dimensiones describen árboles de miembros de dimensión explícita realizando un examen del árbol de un conjunto de relaciones dimensional (DRS). Este examen del árbol lo identifican de forma exclusiva los orígenes de las relaciones.




```
Linkrole:         clase = none | uri | exp
```

```
Source:                    clase = qname | exp
Axis:                      clase = none | value | exp
Generations:  clase = none | value | exp
```

Los nodos de relación de dimensiones no pueden tener árboles subordinados.

17.5.3.6 Nodo de definición: aspecto

Un nodo de aspecto especifica un aspecto exactamente.



Las especificaciones de aspecto dimensionales tienen una propiedad booleana más: `include unreported value`, que se representa con un icono en el diagrama.

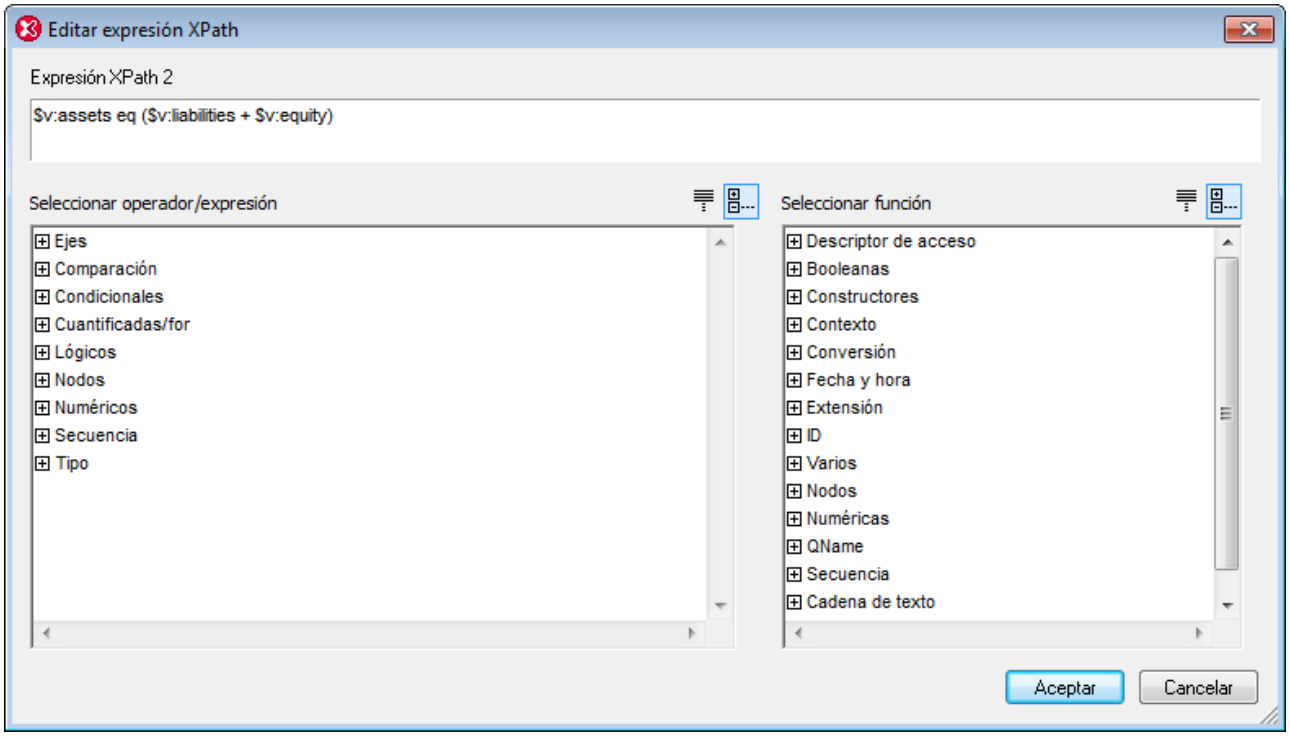


17.5.4 Editar el contenido y las propiedades de los componentes

Las propiedades de los componentes de tabla se pueden editar directamente en el diagrama o en el ayudante de entrada Detalles.

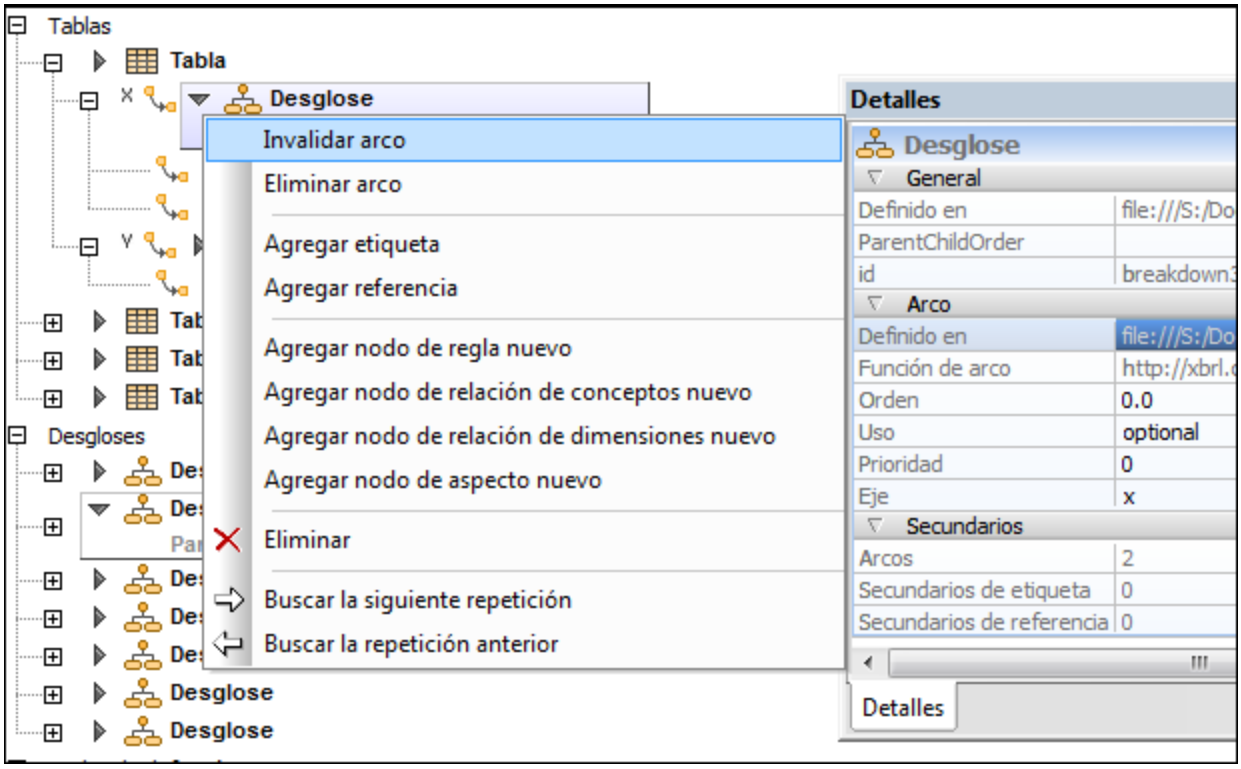
En el diagrama, cuando un componente está contraído, su nombre (si lo tiene) o el valor de la propiedad predeterminada correspondiente aparece en gris junto a la descripción del componente. Haga doble clic en el componente para expandirlo. Haga doble clic en una propiedad para activar el modo de edición. Si la propiedad o el contenido incluye una expresión XPath, aparece el cuadro de diálogo "Editar expresión XPath".

© 2016-2022 Altova GmbH



17.5.5 Relaciones entre los componentes

Puede crear una relación entre dos componentes de tabla enlazando un componente de tabla con otro mediante una operación de arrastrar y colocar. El orden de los secundarios del componente primario depende de los valores de la propiedad `order` del arco. Este orden se puede modificar arrastrando los secundarios a otra posición. Los componentes secundarios también se pueden colocar encima o debajo de un componente primario distinto con el objetivo de copiar o mover la relación (incluidas sus propiedades).

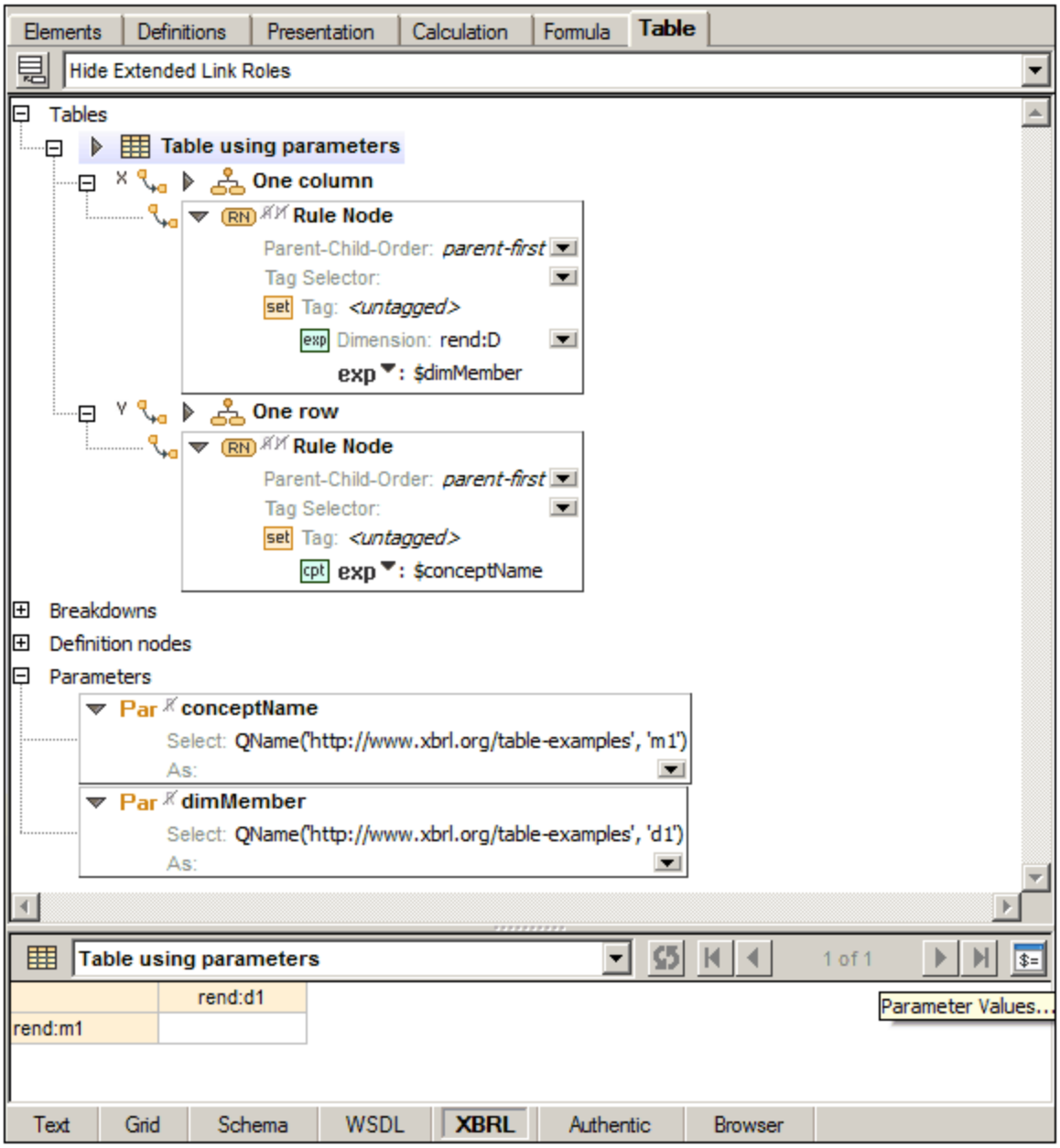


Cuando se crea un componente nuevo desde el menú contextual de un componente (primario) ya disponible, la relación (es decir, el arco) se genera automáticamente. Los comandos **Invalidar arco** y **Eliminar arco** del menú contextual de un componente secundario sirven para invalidar o eliminar la relación entre el componente y su primario. Los arcos de las relaciones invalidadas aparecen en sublíneas. La función de arco de relaciones del componente de tabla no se puede modificar.

17.5.6 Parámetros de tablas

Puede usar parámetros de tabla para definir los ejes de una tabla. Por ejemplo, en la imagen siguiente, el eje X de la tabla seleccionada viene definido por el parámetro `$dimMember` y el eje Y está definido por el parámetro `$conceptName`. Las definiciones de los dos parámetros propiamente dichos aparece en la lista de parámetros globales bajo las definiciones de tabla. En el panel inferior de vista previa del diseño de tabla aparece la tabla que se generará. Las columnas y las filas de la tabla se crean a partir de los ejes X e Y.

© 2016-2022 Altova GmbH

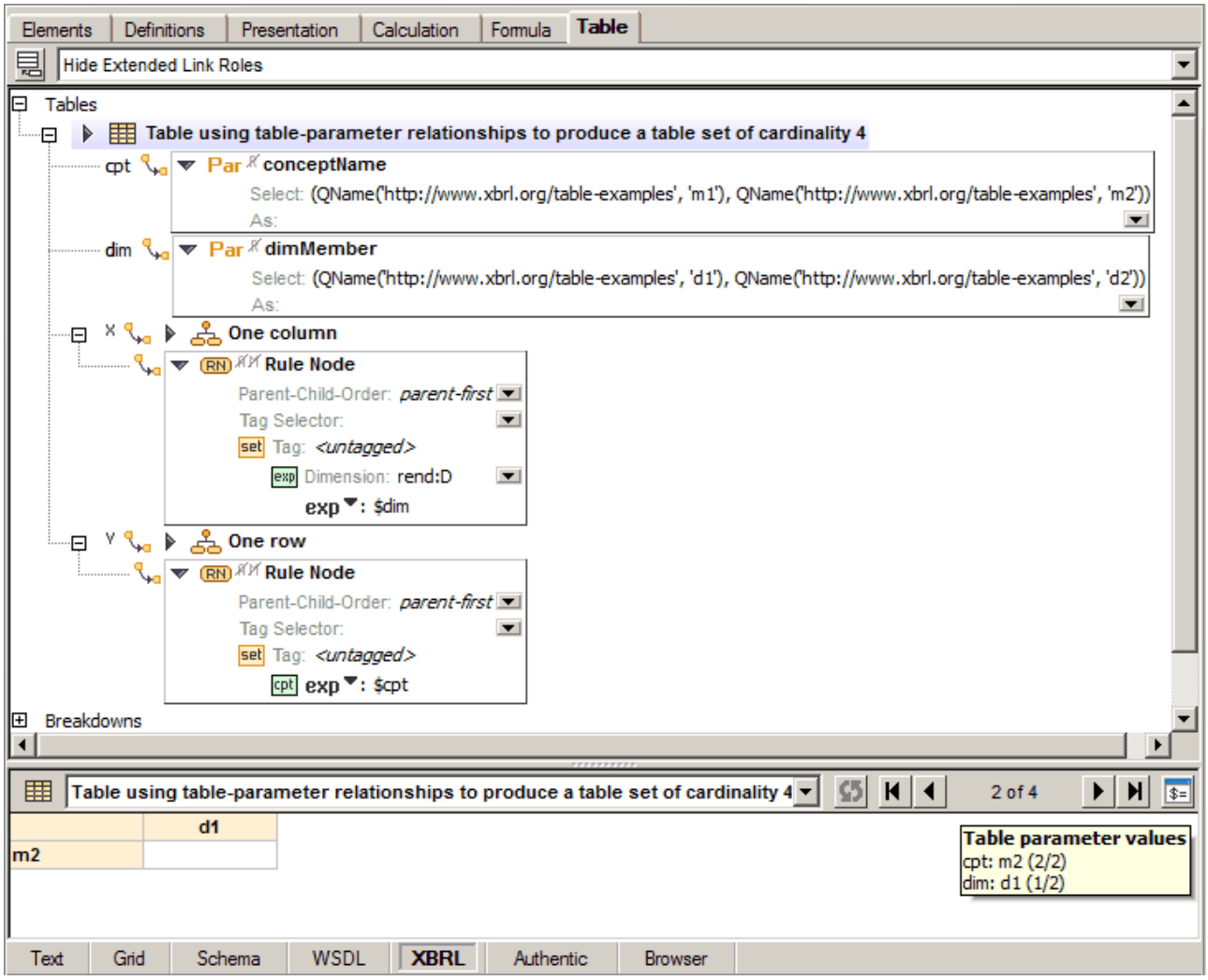


Los parámetros de tabla permiten generar varias tablas relacionadas a partir de una sola definición de tabla, dando lugar a lo que se conoce como *conjunto de tablas*.

- Si un solo parámetro da como resultado una secuencia de valores, entonces el conjunto de tablas contiene una tabla por cada elemento de la secuencia resultante.
- Si la definición de tabla tiene varios parámetros, entonces el conjunto de tablas corresponde a un producto cartesiano ordenado de las secuencias obtenidas al evaluar los parámetros. Ejemplos de producto cartesiano ordenado:

```
A × B = {1,2} × {3,4} = {(1,3), (1,4), (2,3), (2,4)}
B × A = {3,4} × {1,2} = {(3,1), (3,2), (4,1), (4,2)}
```

A continuación puede ver una definición de tabla con dos parámetros de tabla (`conceptName` y `dimMember`), cada uno de los cuales da como resultado una secuencia de dos QName (vea las expresiones XPath de la propiedad `Select`).



Debe tener en cuenta que:

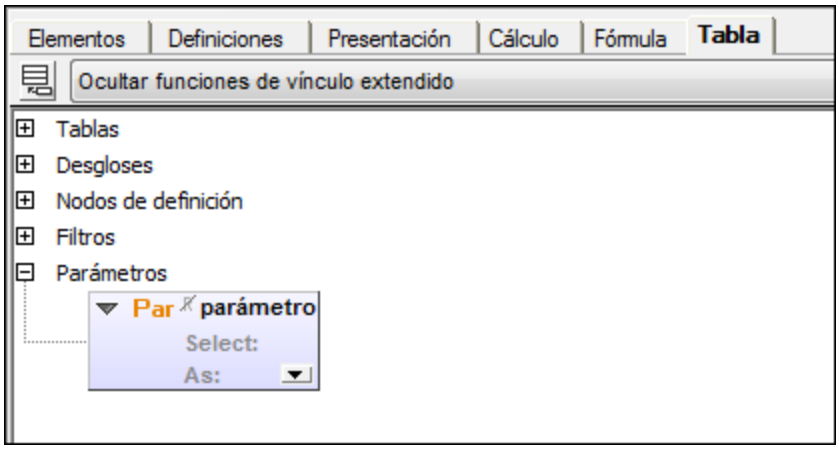
- Los parámetros son parámetros locales, creados para esta tabla concreta haciendo clic con el botón derecho en el componente de tabla y seleccionando μ. No son parámetros globales como los de la primera imagen de este apartado.
- El producto cartesiano ordenado de las dos secuencias de dos QName produce cuatro tablas: `dimMember × conceptName = {d1,d2} × {m1,m2} = {(d1,m1), (d1,m2), (d2,m1), (d2,m2)}`
- Cuando seleccione en el diagrama una definición de tabla que describe un conjunto de tablas, los iconos de navegación del panel de vista previa del diseño de tabla se habilitan y puede recorrer la vista previa de las tablas. En la barra de herramientas del panel de vista previa se indica qué tabla está actualmente en la vista previa. En la imagen anterior, por ejemplo, la tabla actual de la vista previa es la tabla número 2 de 4. También aparece información rápida con los valores de parámetro de la tabla que está en la vista previa (*imagen anterior*).

© 2016-2022 Altova GmbH

- El icono **Actualizar** de la barra de herramientas del panel de vista previa se habilita cuando la vista previa no está sincronizada con las definiciones del editor (p. ej. después de que se añada un concepto nuevo).
- El botón **Valores de parámetro...** de la barra de herramientas del panel de vista previa abre el cuadro de diálogo "Valores de parámetro XBRL", donde puede editar los valores y los tipos de datos de todos los parámetros de tabla (globales y locales).

Definir parámetros XBRL

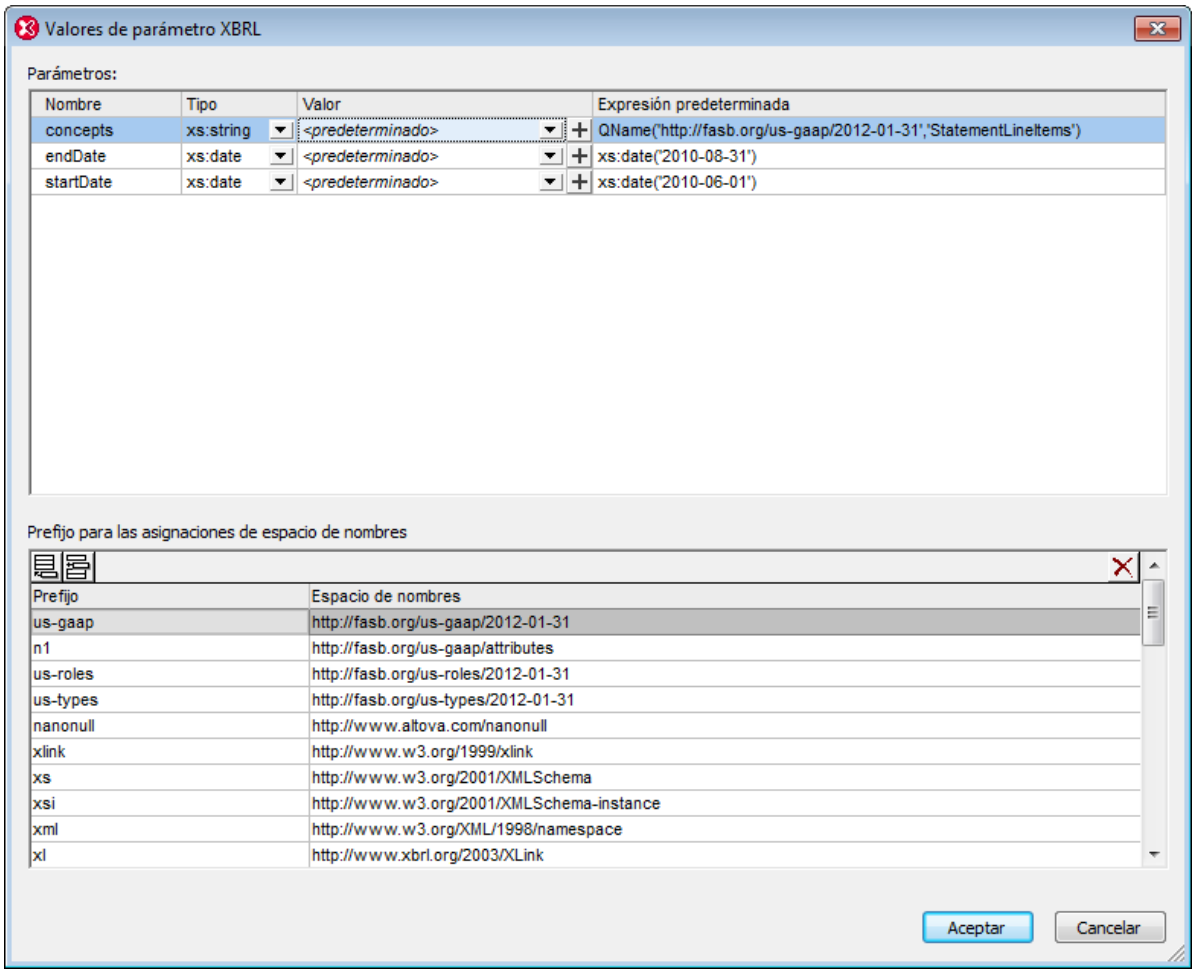
Puede usar parámetros XBRL en expresiones XPath de fórmulas y de definiciones de tabla. Los parámetros que se usarán como parámetros de fórmula (residentes en la base de enlaces de fórmula) se crean en la pestaña **Fórmula**, mientras que los parámetros de tabla (residentes en la base de enlaces de tabla) se crean en la pestaña **Tabla**. Tanto los parámetros de fórmula como los de tabla pueden ser locales o globales. Los parámetros locales son parámetros globales vinculados al componente correspondiente (fórmula o tabla) en el momento de su creación. Los parámetros locales se crean haciendo clic con el botón derecho en el componente (fórmula o tabla) y seleccionado **Agregar parámetro nuevo**. Los parámetros globales se crean haciendo clic con el botón derecho en el fondo en blanco de la pestaña y seleccionando **Agregar parámetro nuevo**. Como resultado se añade un parámetro nuevo llamado `parámetro` en la vista gráfica. Por ejemplo, la imagen siguiente muestra un parámetro global. Para cambiar el nombre del parámetro, haga doble clic en su nombre y edítelo.



Todos los parámetros tienen la marca `Required` (*Obligatorio*). Si tiene establecida esta marca, el parámetro es obligatorio, es decir, su valor debe ser aportado por la aplicación de procesamiento. Si el parámetro no es obligatorio y la aplicación de procesamiento no aporta ningún valor, entonces el valor dado se puede calcular usando la expresión XPath dada en la propiedad `Select`. Haga doble clic en el campo `Select` para escribir una expresión XPath. Este valor será el valor predeterminado del parámetro. La propiedad opcional `As` indica el tipo de datos que exige el parámetro. Haga clic en este campo y elija un tipo de datos en la lista desplegable.

En el caso de los parámetros que se usarán como parámetros de tabla, puede editar el tipo de datos del parámetro y aportar un valor de parámetro que reemplace el valor predeterminado. Esto se hace de la siguiente manera:

1. Haga clic en **XBRL | Valores de parámetro** para abrir el cuadro de diálogo "Valores de parámetro" (*imagen siguiente*).



2. Ahora introduzca un valor de parámetro. Este valor reemplazará el valor predeterminado.

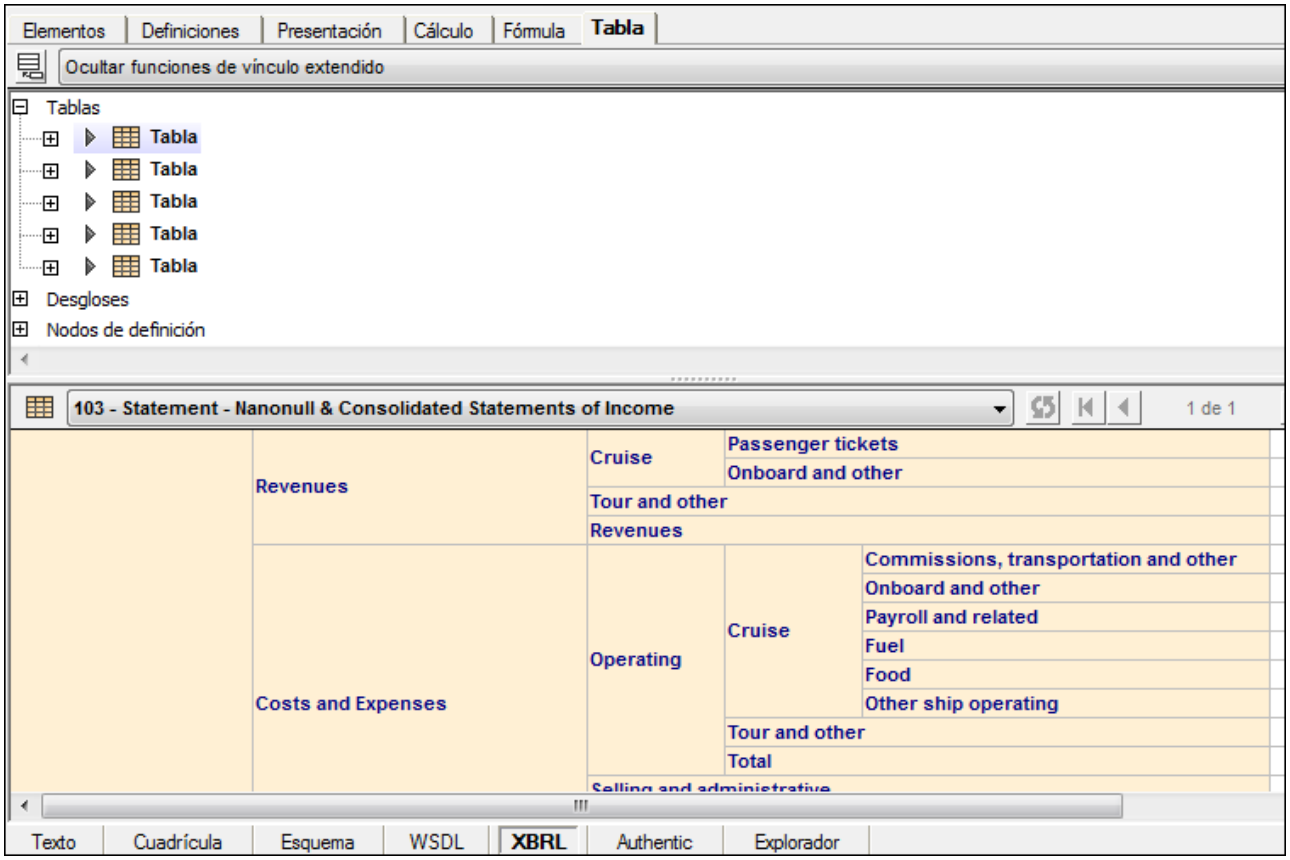
   Como los parámetros que se utilizan como parámetros de tabla pueden tomar varios valores, puede añadir varios valores para un mismo parámetro. Esto se hace con el icono **+** de la columna *Valor*.

Los valores de los parámetros globales asignados en este cuadro de diálogo solamente se evalúan para los parámetros de tabla. Los valores de parámetro utilizados en las fórmulas no se pueden editar en este cuadro de diálogo.

17.5.7 Vista previa del diseño de tabla

El panel *Vista previa del diseño de tabla* está situado en la parte inferior de la pestaña ***Tabla***, bajo el árbol de definiciones de tabla (*imagen siguiente*). En este panel hay un cuadro combinado que enumera todas las tablas de la base de enlaces de tabla de la taxonomía activa. Para ver una vista previa del diseño de una tabla, selecciónela en este cuadro combinado. Recuerde que la vista previa solamente muestra el diseño de la tabla y que las celdas de la tabla están vacías. Esto se debe a que la taxonomía XBRL no contiene datos.

© 2016-2022 Altova GmbH



El panel de vista previa del diseño de tabla sirve para:

- ver diseños de tabla: la vista previa se actualiza automáticamente cuando se modifican las definiciones de tabla.
- ir directamente a una definición del componente haciendo clic en una celda de la tabla (y viceversa, haciendo clic en un componente en el árbol de definiciones para ir a la celda correspondiente).
- abrir el cuadro de diálogo "Parámetros de tabla XBRL" para gestionar los parámetros de la tabla (haciendo clic en el icono **Valores de parámetro...** de la barra de herramientas).

Edición

A continuación explicamos cómo se gestionan las modificaciones en las definiciones de tabla y en la taxonomía:

- *Cambios en la tabla:* si modifica la estructura de una definición de tabla (en la pestaña ***Tabla*** o en el ayudante de entrada Detalles), la vista previa del diseño de tabla se actualiza automáticamente. Los cambios en las definiciones de parámetro o en los valores de parámetro también provoca la actualización de la vista previa.

- *Cambios en la DTS:* el panel de vista previa usa el motor XPath de Altova para evaluar las expresiones XPath utilizadas en los nodos de definición. El modelo XPath se crea cuando se carga un esquema de taxonomía en la vista XBRL y se actualiza durante la validación. Si la DTS subyacente se modifica (p. ej. si se edita un concepto o una base de enlaces), la vista previa de la tabla dejará de estar

sincronizada con la DTS modificada. Entonces aparece el icono que indica que la tabla no está sincronizada con la DTS. El icono **Actualizar** sirve para volver a detectar la DTS y, por tanto, para volver a validar la taxonomía.

Gestión de errores

Así se gestionan los errores relacionados con la vista previa del diseño de tabla:

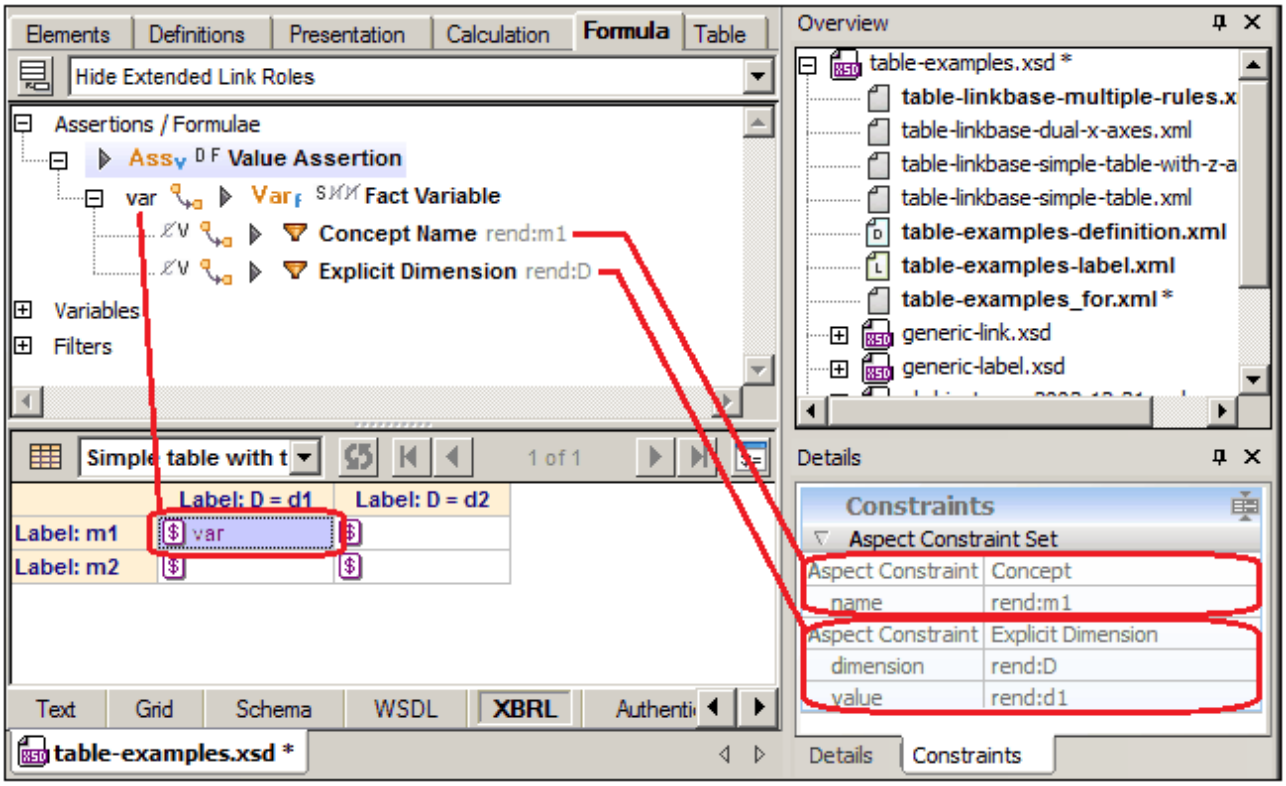
- *Expresiones no válidas en nodos de definición de tabla:* si un nodo de definición de tabla incluye una expresión XPath que no se puede resolver, el encabezado de la correspondiente celda aparece en color rojo. En casos así, la restricción de aspecto no válida se resalta en la pestaña *Restricciones* del ayudante de entrada Detalles.
- *Nodos de relación que no se pueden resolver:* si no se puede resolver un nodo de relación[896] debido a propiedades no válidas o a una DTS no válida, el diseño muestra una celda marcador de posición con un mensaje de error en rojo.
- *Nodos de regla combinados sin nodos secundarios:* si un nodo de regla combinado[896] no tiene nodos secundarios, el diseño muestra una celda marcador de posición con un mensaje de error en rojo.
- *DTS no válida:* si al cargarse en el editor XBRL la taxonomía no es válida, el modelo XPath no está disponible. La vista previa del diseño de tabla dará error (lo cual se indica con el icono). A pesar de ello, se puede crear un diseño limitado. Sin embargo, las expresiones XPath no se podrán evaluar. El texto de información rápida del icono de la barra de herramientas indicará cómo resolver el problema (es decir, corrigiendo el error de validación y volviendo a validar la taxonomía).

17.5.7.1 Generar fórmulas en la vista previa del diseño de tabla

El panel de vista previa del diseño de tabla también está disponible desde la pestaña ***Fórmula*** de la vista XBRL (*imagen siguiente*) y sirve para crear variables fact bajo conjuntos de variables (es decir, bajo fórmulas o aserciones de valor/existencia). En estos casos, las celdas de los ejes de la tabla no se pueden seleccionar porque los nodos de definición de tabla correspondientes no se pueden ver en la base de enlaces de fórmula. Las celdas de datos, por el contrario, muestran el icono **Agregar una variable fact**, que se habilita en cuanto se selecciona un conjunto de variables en la fórmula (*imagen siguiente*).

Para agregar una variable fact a un conjunto de variables seleccione el conjunto de variables en la pestaña ***Fórmula***. Un conjunto de variables es una fórmula o una aserción de valor/existencia. En la imagen siguiente, por ejemplo, el conjunto de variables seleccionado es una aserción de valor. En las celdas de la vista previa del diseño de tabla, se habilitará el icono **Agregar una variable fact**. Haga clic en el icono para agregar una variable al conjunto de variables. Durante la ejecución se creará bajo el conjunto de variables una variable fact nueva que contiene un filtro adecuado por cada restricción de aspecto definida por la celda de datos.

© 2016-2022 Altova GmbH



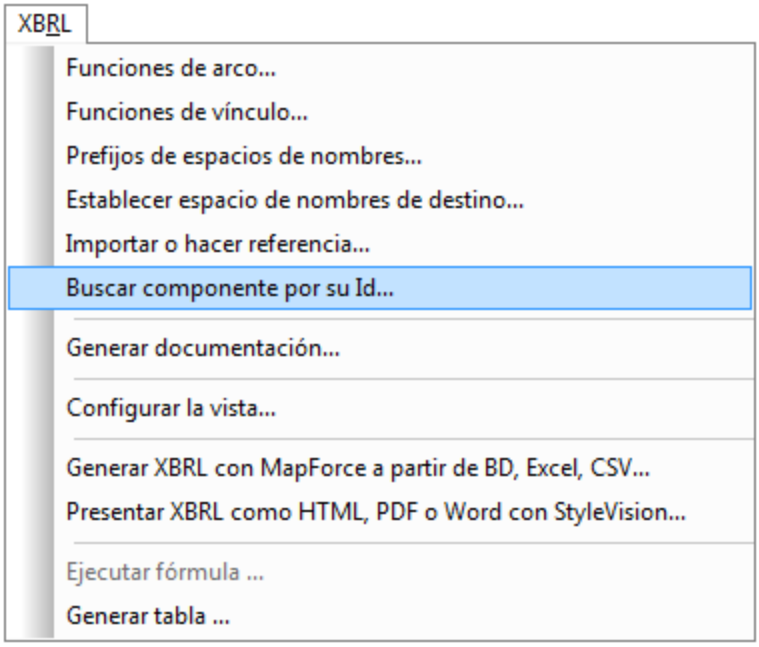
Si la aserción o fórmula seleccionada tiene una variable que contiene filtros que cumplan las restricciones de aspecto de una celda de datos de la vista previa, entonces el nombre de variable aparece en la celda de datos. Esto suele ocurrir después de crear una variable fact con el icono **Agregar una variable fact** .

17.5.8 Buscar componentes de tabla

Puede buscar componentes de tabla por su identificador o recorriendo el documento.

Buscar componentes de tabla por su identificador

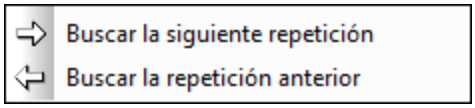
En las taxonomías con bases de enlace de tabla o de fórmula de gran tamaño que contienen varios componentes del mismo tipo (p. ej. aserciones, filtros, tablas, etc.), quizás la mejor opción sea buscar los componentes por su identificador. Para ello puede utilizar el comando de menú **XBRL | Buscar componente por su Id**.



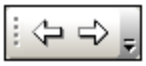
Al hacer clic en este comando se abre un cuadro de diálogo donde puede insertar el Id. que desea buscar.

Buscar repeticiones del componente

La mayoría de los componentes de tabla se presentan varias veces dentro del diagrama de la base de enlaces de tabla: (i) en la definición, que está justo debajo del nodo de sección correspondiente y (ii) en todas las referencias al componente (por medio de relaciones). Los comandos **Buscar la siguiente repetición** y **Buscar la repetición anterior** del menú contextual del componente (*imagen siguiente*) sirven para navegar hasta dónde se haga referencia al componente.



Estos comandos también están disponibles en la barra de herramientas (*imagen siguiente*).



Al llegar a la definición del componente aparece un mensaje a tal efecto.

© 2016-2022 Altova GmbH

17.6 XULE

XULE (abreviación en inglés de XBRL Rule) es un lenguaje de consulta que se usa en informes y taxonomías XBRL. Su objetivo principal es permitir hacer consultar y comprobar informes antes de enviarlos para garantizar la calidad de los datos. XULE permite comprobar informes de dos maneras:

- *Extraer resultados de los datos de los informes:* para ello consulta datos en los informes y computa resultados a partir de esos datos. Se puede evaluar la calidad de los resultados.
- *Crear aserciones:* con ellas se pueden comprobar los datos de los informes y después tomar medidas en función de los resultados

Enlaces de Internet

- Página web de XULE (contiene un resumen de XULE)
- Documentación de la sintaxis del lenguaje XULE

Características en XMLSpy

XMLSpy cuenta con varias características que permiten trabajar con XULE:

- **Procesador XULE integrado** que procesa documentos y expresiones XULE con respecto a un documento de instancia XBRL
- Creación de documentos conforme a XULE
- Validación de la correcta sintaxis de documentos XULE con respecto a la especificación XULE
- Coloración de sintaxis en documentos XULE
- Finalización automática de construcciones en lenguaje XULE al editar documentos XULE
- Una **ventana XULE** en la que puede realizar consultas en documentos de instancia XBRL
- Procesamiento de una instancia XBRL con respecto a un único documento XULE o un conjunto de documentos XULE almacenados en un archivo zip; para ello se pueden usar el motor de XMLSpy o el motor de servidor RaptorXML(+XBRL) Server 1068
- Integración en proyectos de XMLSpy 1060 de la ejecución de documentos y conjuntos de documentos XULE

Estas características se describen con más detalle en las subsecciones de esta sección.

Altova RaptorXML+XBRL Server

Altova's RaptorXML+XBRL Server permite procesar XULE de forma personalizada y rápida, lo que permite procesar documentos XULE desde la línea de comandos con scripts y a través de varias APIs de servidor y motor, lo que incluye una potente API de Python. Para más información consulte:

- La página de RaptorXML+XBRL en el sitio web de Altova
- La certificación DQC de Altova RaptorXML+XBRL
- La documentación del producto RaptorXML+XBRL
- La documentación de la API de Python de RaptorXML

17.6.1 Documentos XULE

Con XMLSpy puede crear, validar y ejecutar documentos XULE. Las características relacionadas se describen a continuación.

Archivos conforme a XULE (archivos .xule)

El tipo de archivos **.xule** viene predefinido en XMLSpy conforme a XULE. Esto significa que cuando abre un archivo **.xule** en XMLSpy se habilitan la coloración de sintaxis y la finalización automática. Puede indicar qué otras extensiones son conforme a XULE en la sección Tipos de archivos del cuadro de diálogo "Opciones"[1587].

Conjuntos de documentos XULE

Se pueden guardar varios documentos XULE en un archivo zip (por lo general con extensión **.zip**). A este archivo zip se le considera un conjunto de documentos XULE (o conjunto de reglas XULE). Puede ejecutar todo el conjunto de documentos XULE de una instancia XBRL definiendo el archivo zip como Archivo XULE que ejecutar[929].

Un archivo zip puede tener cualquier estructura. Los archivos XULE de los distintos niveles del archivo se usan durante la ejecución; los archivos que no sean XULE se omiten.

Coloración de sintaxis

Los documentos XULE se pueden editar en la vista Texto de XMLSpy. La imagen siguiente muestra la coloración de sintaxis predeterminada de un documento XULE de ejemplo. Puede personalizar esta coloración en la sección Fuentes y colores | Vista Taxto[1603] del cuadro de diálogo "Opciones".

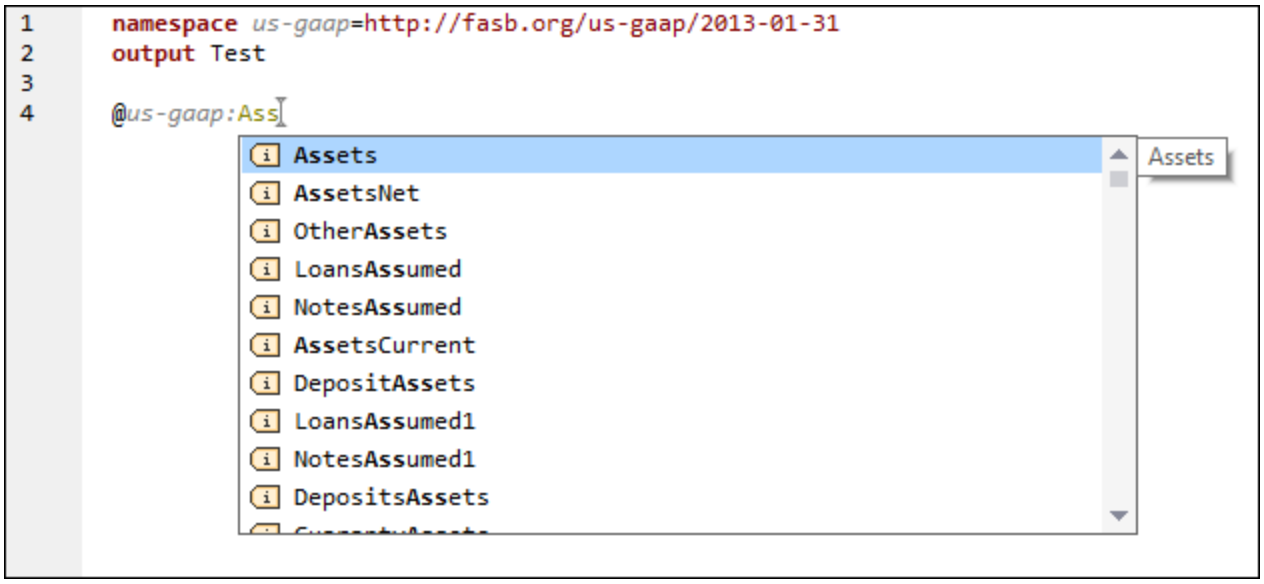
```
1   namespace us-gaap=http://fasb.org/us-gaap/2013-01-31
2
3   // Balance sheet test
4   assert balance-test unsatisfied
5
6   @us-gaap:Assets#assets == @us-gaap:LiabilitiesAndStockholdersEquity#liabilities
7
8   message
9
10  "Balance sheet is unbalanced in period {$assets.period}, {$assets} != {$liabilities}"
```

Finalización automática

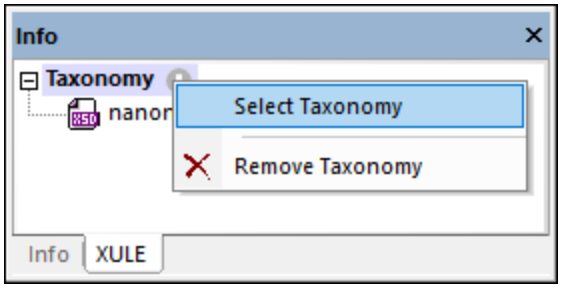
Al introducir las reglas en el documento XULE aparecen dos tipos de asistentes de finalización automática:

- relacionados con la sintaxis del lenguaje XULE
- relacionados con la estructura de la taxonomía XBRL seleccionada

© 2016-2022 Altova GmbH



Puede seleccionar en la ventana *Información* la taxonomía que quiere usar (*imagen siguiente*). En la pestaña *XULE* haga clic en el icono que hay a la derecha del elemento Taxonomía y seleccione en el menú que aparece **Seleccionar taxonomía**. Después navegue hasta la taxonomía que necesite y selecciónela. Solo se puede añadir una taxonomía cada vez. Si añade una taxonomía nueva, esta sobrescribirá la que hubiera antes. También puede eliminar la taxonomía que estuviera seleccionada antes de añadir una nueva.



Nota: para que funcione función de finalización automática de los componentes de la taxonomía debe añadir al documento XULE las declaraciones de espacio de nombres de todos los componentes de la taxonomía que necesite (*véanse las imágenes más arriba*). No es necesario que el prefijo del espacio de nombres sea igual al de la taxonomía, pero se recomienda que use los mismos prefijos de espacio de nombres para evitar confusiones.

Validar XULE

Puede comprobar si la sintaxis de un documento XULE es correcta con respecto a la especificación del lenguaje XULE con el comando **XML | Validar (F8)**.

Integración en proyectos de XMLSpy

Para integrar documentos XULE en un proyecto de XMLSpy:

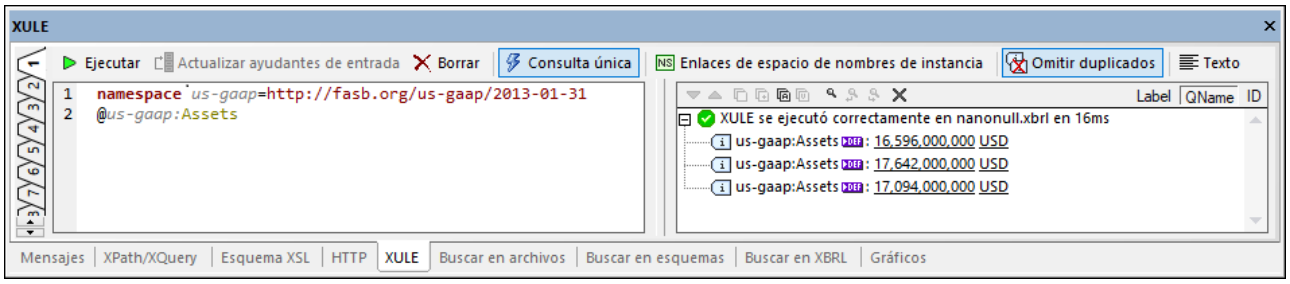
1. Añadir el documento XULE a la carpeta apropiada en un proyecto de XMLSpy[1060].
2. En las propiedades del proyecto de esa carpeta[1328] introduzca el archivo de instancia XBRL en el que quiere ejecutar el archivo XULE.

3. Haga clic con el botón derecho en el documento XULE del proyecto y seleccione el comando **XBRL | Executar XULE**. El documento XULE se ejecuta en la instancia XBRL que se haya indicado para esa carpeta y los resultados se muestran en la ventana Mensajes[125] o en un documento nuevo (*véanse las opciones de ejecución XULE*[1624]).

17.6.2 Ventana XULE

La ventana XULE (*imagen siguiente*) es una ventana de resultados[134]. En ella puede hacer consultas en el documento de instancia XBRL activo y ver los resultados de esas consultas.

La ubicación predeterminada de la ventana XULE es bajo la ventana principal, en la parte inferior de la interfaz de XMLSpy[119]. Esta ventana tiene nueve pestañas, cada una de las cuales está dividida en dos paneles: (i) un panel de expresiones XULE donde se introduce la expresión XULE (o regla XULE) que quiere ejecutar en el documento activo y (ii) el panel *Resultados*, donde aparecen los resultados de la ejecución.



Para ejecutar una expresión XULE en el documento de instancia XBRL activo siga estos pasos:

1. Asegúrese de que el documento de instancia XBRL en el que quiere hacer la consulta es el documento que está activo en la ventana principal.
2. Introduzca la expresión XULE en el panel izquierdo, que cuenta con coloración de sintaxis y finalización automática.
3. Haga clic en **Ejecutar** en la barra de herramientas de la ventana para ejecutar la expresión.
4. Los resultados de la ejecución se muestran en el panel derecho (*panel Resultados*). Si hace clic en uno de los enlaces de los resultados irá directamente al nodo correspondiente del documento de instancia XBRL.

Nota: la coloración de sintaxis para XULE se puede personalizar en el cuadro de diálogo "Opciones"[1583] (en la sección Fuentes y colores | Vista texto[1603]). Para más información sobre la finalización automática consulte más abajo la descripción del comando **Actualizar ayudantes de entrada** de la barra de herramientas.

Barra de herramientas: comandos y opciones

La barra de herramientas de la ventana XULE contiene comandos y opciones que permiten crear y ejecutar expresiones XULE. Los describimos a continuación.

Ejecutar
Este comando ejecuta la expresión XULE.

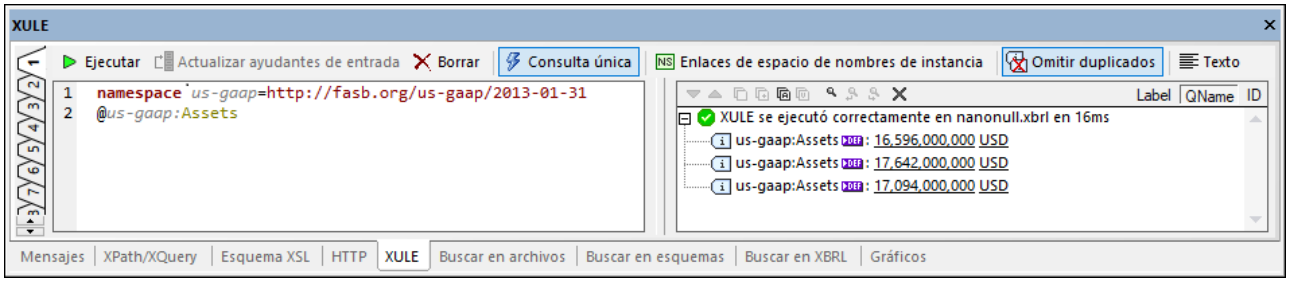
© 2016-2022 Altova GmbH

Actualizar ayudantes de entrada
El panel de expresiones XULE ofrece dos tipos de finalización automática según teclea: (i) una relacionada con la sintaxis XULE y (ii) otra relacionada con la estructura del documento de instancia XBRL activo. Para ello la ventana XULE debe leer la taxonomía XBRL a la que hace referencia la instancia XBRL. Haga clic en **Actualizar ayudantes de entrada** para cargar la taxonomía asociada a la instancia XBRL. Una vez la haya cargado se habilita la finalización automática relacionada con la estructura y este botón de la barra de herramientas se deshabilita (ya que ya no es necesario para este documento de instancia XBRL). Tenga en cuenta que la taxonomía se lee también cada vez que hace clic en **Ejecutar**.

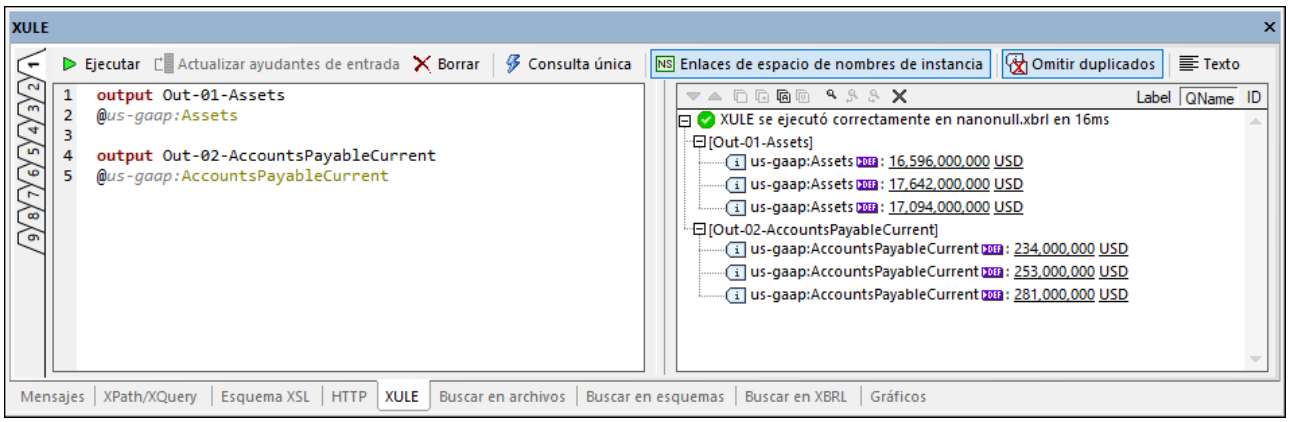
Borrar
Haga clic en **Borrar** para borrar la expresión.

Consulta única
La ventana XULE de XMLSpy contiene el modo Consulta única, que permite: (i) introducir una expresión sin la palabra clave `output` y como consulta única y (ii) generar el resultado como una única salida. La sintaxis XULE válida requiere la palabra `output`, pero esta opción permite consultar el documento XBRL de forma rápida sin ella.

Las dos imágenes siguientes muestran, respectivamente, cómo usar la consulta única y varias consultas.



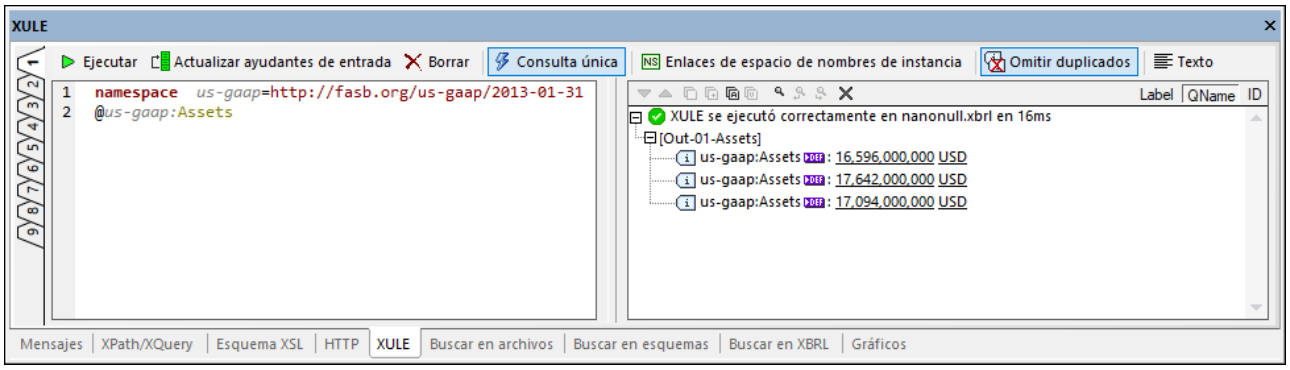
Modo Consulta única habilitado: no hace falta la palabra clave 'output'.



Modo Consulta única deshabilitado: varias consultas con la palabra clave "output" y varios resultados.

Enlaces de espacio de nombres de instancia
Si marca la opción **Enlaces de espacio de nombres de instancia**, entonces no necesita declarar espacios de nombre en la consulta XULE; los prefijos de espacio de nombres están vinculados a los URIs de espacio de nombres declarados para ellos en la instancia XBRL. por ejemplo, en las imágenes anteriores el prefijo de espacio de nombres `us-gaap` está vinculado al espacio de nobres definido para él en el documento de

instancia. Si no marca esta opción debe declarar los espacios de nombres en la consulta XULE (con la palabra clave **namespaces**, como se ve en la imagen siguiente).



Gracias a esta opción no tiene que escribir las declaraciones de espacio de nombres en la consulta XULE.

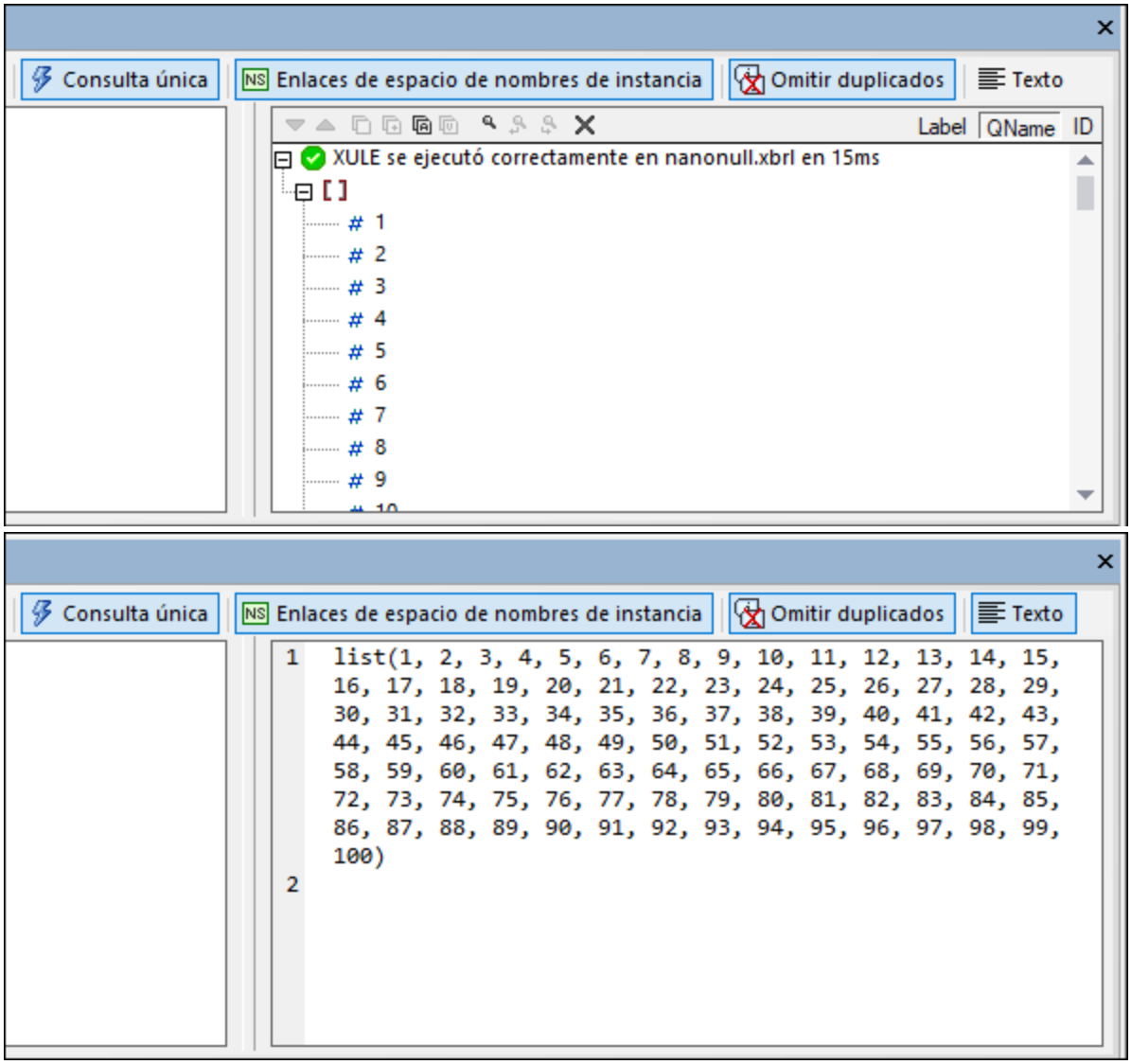
Ignorar duplicados

Un hecho duplicado ocurre (habitualmente en Inline XBRL) si se nota más de una vez el mismo hecho en el código HTML. La opción Ignorar duplicados indica que este tipo de hechos deben aparecer una sola vez en el resultado.

Texto

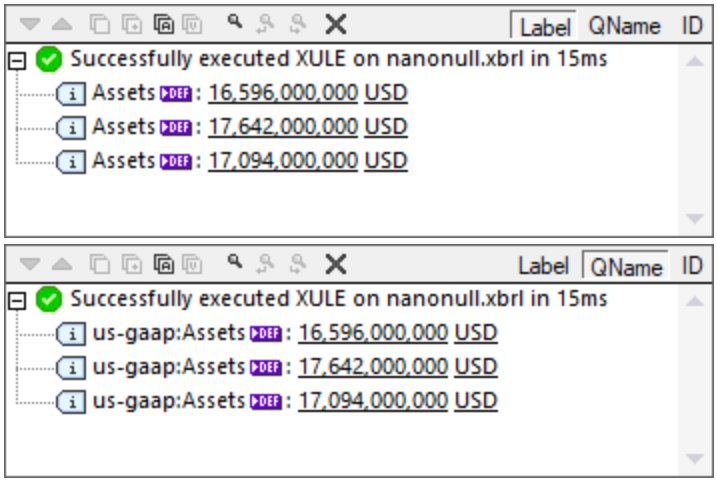
La opción **Texto** alterna entre mostrar los datos de salida en el panel *Resultados* en forma de texto o de árbol. Por ejemplo, en las imágenes siguientes se ve el resultado en forma de árbol (*izquierda*) y en forma de texto (*derecha*). La consulta es `list(for $i in range(100) $i)`.

© 2016-2022 Altova GmbH



El panel Resultados

Los resultados se pueden mostrar con sus etiquetas, nombres QName o IDs. Seleccione la opción que prefiera en la barra de herramientas del panel Resultados; puede hacerlo tanto antes como después de ejecutar la consulta. La imagen siguiente muestra los resultados con etiquetas (*izquierda*) y con nombres QName (*derecha*).



La barra de herramientas del panel *Resultados* contiene iconos para las funciones de navegación, búsqueda y copia. Estos iconos, empezando por la izquierda, se describen en la tabla siguiente. Los comandos correspondientes también están disponibles en el menú contextual de los elementos de la lista de resultados.

Icono	Función
Siguiente, Anterior	Selecciona, respectivamente, el elemento siguiente o anterior de la lista de resultados
Copiar mensaje seleccionado	Copia el elemento de resultado seleccionado en el portapapeles
Copiar mensaje seleccionado incluyendo elementos secundarios	Copia el elemento de resultado seleccionado en el portapapeles, así como sus elementos secundarios. Cada elemento se copia en una línea distinta
Copiar todos los mensajes	Copia todos los elementos de resultado en el portapapeles
Copiar el valor de la línea seleccionada en el portapapeles	Solo copia en el portapapeles el valor del elemento de resultado seleccionado
Buscar	Abre el cuadro de diálogo "Buscar" para buscar cadenas de texto en los resultados
Buscar anterior	Busca la ocurrencia anterior del último término introducido en el cuadro de diálogo "Buscar"
Buscar siguiente	Busca la siguiente ocurrencia del último término introducido en el cuadro de diálogo "Buscar"
Borrar	Borra la lista de resultados

© 2016-2022 Altova GmbH

17.6.3 Ejecutar XULE

Para ejecutar un documento XULE[922] en un documento de instancia XBRL seleccione el comando de menú **XBRL | Ejecutar XULE**[1534]. Este comando se puede usar en varios casos:

- si el documento activo es un documento XULE y selecciona este comando, la aplicación le pedirá que seleccione la instancia XBRL con respecto a la cual quiere que se ejecute el documento XULE.
- si el documento activo es un documento XULE y selecciona este comando, la aplicación le pedirá que seleccione el documento XULE o el conjunto de documentos XULE que quiere usar.
- si (i) el documento XULE (archivo `.xule`) o el conjunto de documentos XULE[922] (archivo zip) y el documento de instancia XBRL forman parte de un proyecto de XMLSpy[1060] y (ii) el archivo de instancia XBRL se ha definido como el archivo XBRL de destino en las propiedades del proyecto de XMLSpy[1328], entonces haga clic con el botón derecho en el archivo XULE del proyecto en la ventana del proyecto de XMLSpy y seleccione **Ejecutar XULE**. Los documentos XULE se ejecutan con respecto al documento XBRL que se haya definido en el proyecto como archivo de destino para la ejecución XULE.

Además, puede configurar XMLSpy para que ejecute comandos de RaptorXML(+XBRL) Server[1068], lo que incluye comandos de procesamiento XULE, desde la interfaz de XMLSpy.

Opciones de ejecución XULE

Existen varias opciones de ejecución XULE:

- los resultados se pueden enviar a: (i) la ventana Mensajes o (ii) un documento nuevo que aparece en una ventana nueva de XMLSpy y se almacena temporalmente en la memoria; este documento se puede guardar en un archivo con el comando **Archivo | Guardar como**[1269].
- los hechos duplicados son referencias múltiples al mismo hecho. (Un hecho duplicado suele ocurrir en Inline XBRL si el mismo hecho se nota más de una vez debido al código HTML) Puede elegir que se informe solo una vez de cada hecho.

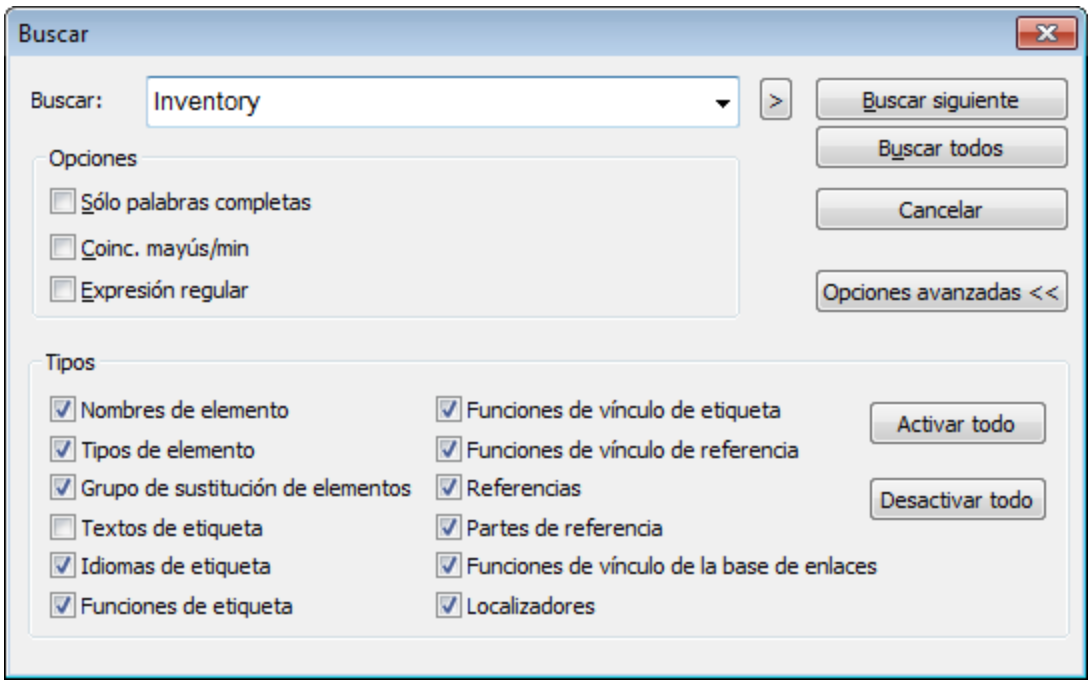
Puede seleccionar las opciones que prefiera en el cuadro de diálogo "Opciones" de la pestaña de XBRL *XULE*[1624] (**Herramientas | Opciones | XBRL | XULE**).

17.7 Búsqueda en XBRL

En la vista **XBRL** puede realizar búsquedas en la taxonomías XBRL con ayuda de la función de búsqueda en XBRL de XMLSpy. Esta función se habilita cuando se abre una taxonomía XBRL en la vista XBRL. Hay tres maneras de acceder a esta función:

- Seleccione el comando de menú **Edición | Buscar**.
- Haga clic en el botón **Buscar** de la ventana "Buscar en XBRL".
- Pulse las teclas **Ctrl+F**.

El resultado de estas tres acciones es el mismo: se abre el cuadro de diálogo Buscar (*ver imagen*).



Siga estos pasos para utilizar la función de búsqueda en XBRL:

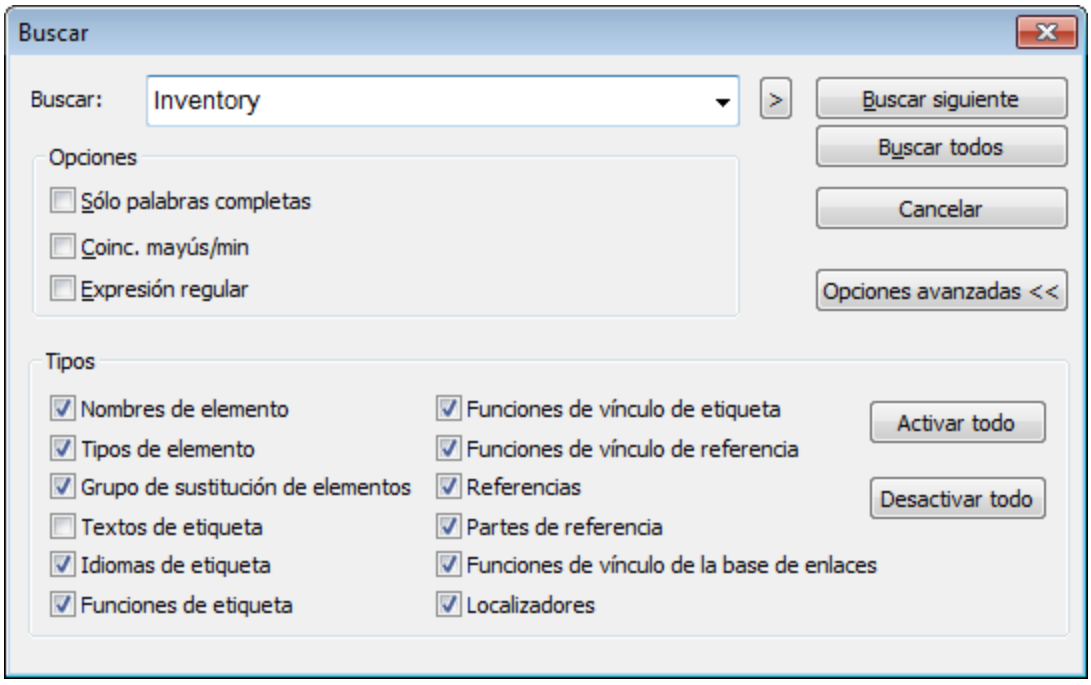
- Escriba el término de búsqueda 930 en el campo *Buscar* del cuadro de diálogo "Buscar" (*imagen anterior*) y marque las opciones correspondientes.
- Indique en qué componentes XBRL se debe realizar la búsqueda 930 en el grupo de opciones *Tipos*.
- Ejecute el comando 933 haciendo clic en el botón **Buscar siguiente** o **Buscar todos**.
- Use la ventana Buscar en XBRL 935 para ver los resultados de la búsqueda y navegar por ellos.

17.7.1 Término de búsqueda

En el cuadro de diálogo puede indicar si el término de búsqueda debe coincidir con una palabra completa y si debe coincidir en el uso de mayúsculas y minúsculas. Para ello marque las casillas pertinentes del grupo Opciones (*imagen siguiente*). Si desea buscar una expresión regular, marque la opción *Expresión regular* del

© 2016-2022 Altova GmbH

grupo Opciones antes de pulsar el botón **Buscar siguiente** o **Buscar todos**. Más abajo encontrará más información sobre cómo usar las expresiones regulares.

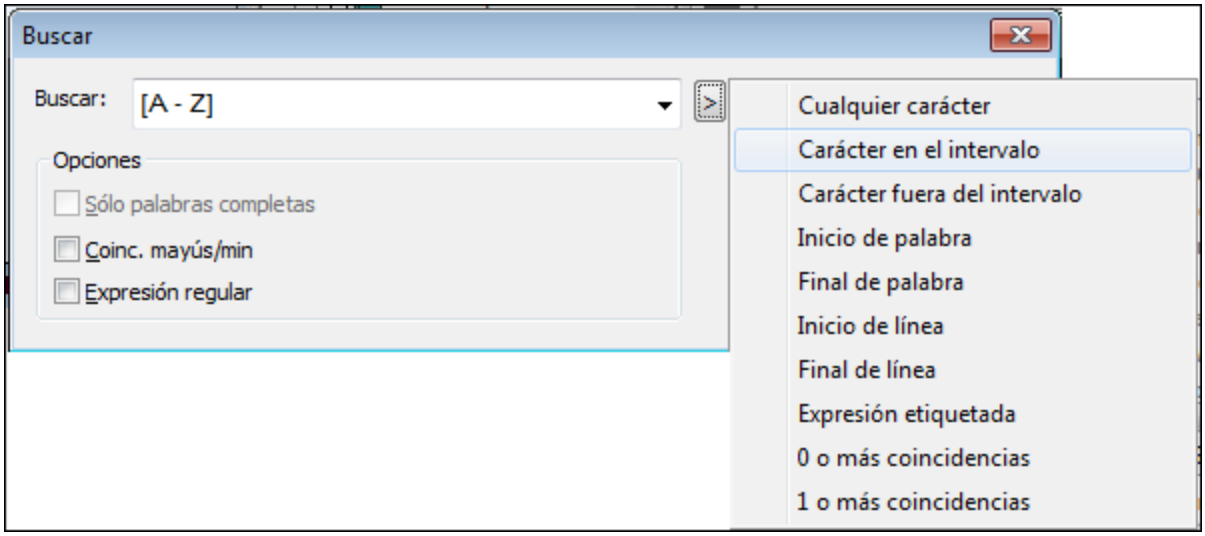


Nota: se entiende como palabra completa aquella que está delimitada por cualquier carácter no alfanumérico o por un guión bajo. Es decir, el término de búsqueda `asset` encontrará el texto `xbrl:asset` porque el carácter de dos puntos (:) se considera un delimitador de palabras.

En el grupo de opciones Tipos indique qué componentes se debe realizar la búsqueda.

Expresiones regulares

Puede usar expresiones regulares para limitar sus criterios de búsqueda. El cuadro de diálogo incluye una lista emergente con opciones para crear expresiones regulares. Para abrir la lista, haga clic en el botón **>** situado a la derecha del campo de entrada del término de búsqueda.



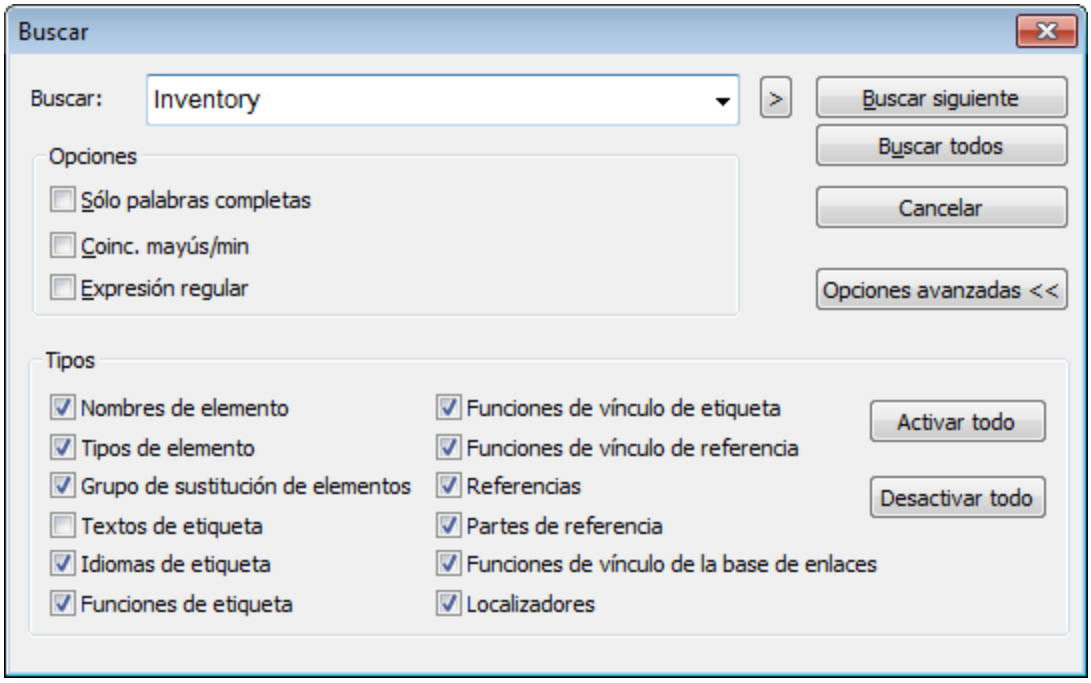
Haga clic en la descripción de la expresión que desea usar y se inserta el carácter de la expresión en el campo de entrada. Los caracteres utilizados en la sintaxis de expresiones regulares son:

`.`	Cualquier carácter. Es un comodín para un solo carácter.
`\(`	El inicio de una región en la que se realizará la búsqueda.
`\)`	El final de la región en la que se realizará la búsqueda.
`\<`	Inicio de palabra.
`\>`	Final de palabra.
`\x`	Encuentra el carácter x que sigue a la barra inversa, cuando el carácter `x` es un carácter que se utiliza en la sintaxis de expresiones regulares. Por ejemplo, `\[` encuentra el carácter `[` y no el inicio de un conjunto de caracteres.
`[...]`	Por ejemplo, `[abc]` encuentra los caracteres `a`, `b` o `c`. También puede usar intervalos como `[a-z]` para buscar cualquier carácter en minúsculas.
`[^...]`	Un carácter cualquiera no perteneciente al conjunto. Por ejemplo, `[^A-Za-z]` encuentra cualquier carácter excepto caracteres alfabéticos en mayúsculas o minúsculas.
`^`	Inicio de línea (a no ser que se use dentro de un conjunto de caracteres, ver fila anterior).
`$`	Final de línea. Por ejemplo, `A+$` encuentra una `A` o más de una `A` que estén al final de una línea.
`*`	0 veces o más. Por ejemplo, `Sa*m` encuentra `Sm`, `Sam`, `Saam`, `Saaam`, etc.
`+`	1 veces o más. Por ejemplo `Sa+m` encuentra `Sam`, `Saam`, `Saaam`, etc.

 © 2016-2022 Altova GmbH

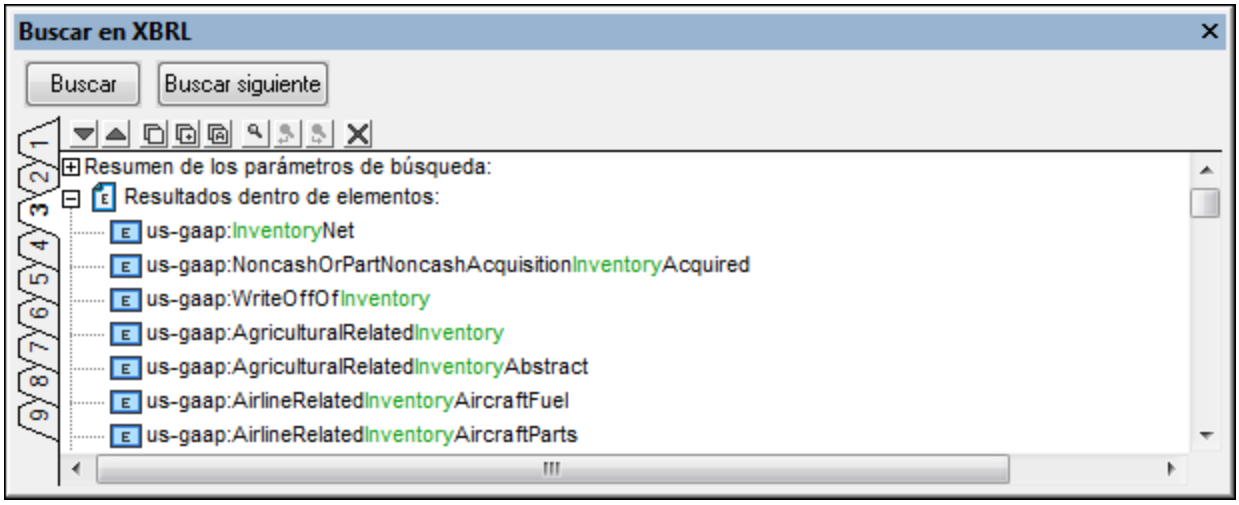
17.7.2 Ejecución del comando

Después de escribir el término de búsqueda, marcar opciones de búsqueda y tipos de componentes, hay dos maneras de ejecutar el comando de búsqueda: **Buscar siguiente** o **Buscar todos**. Estos comandos se ejecutan con los botones del cuadro de diálogo Buscar (*imagen siguiente*).



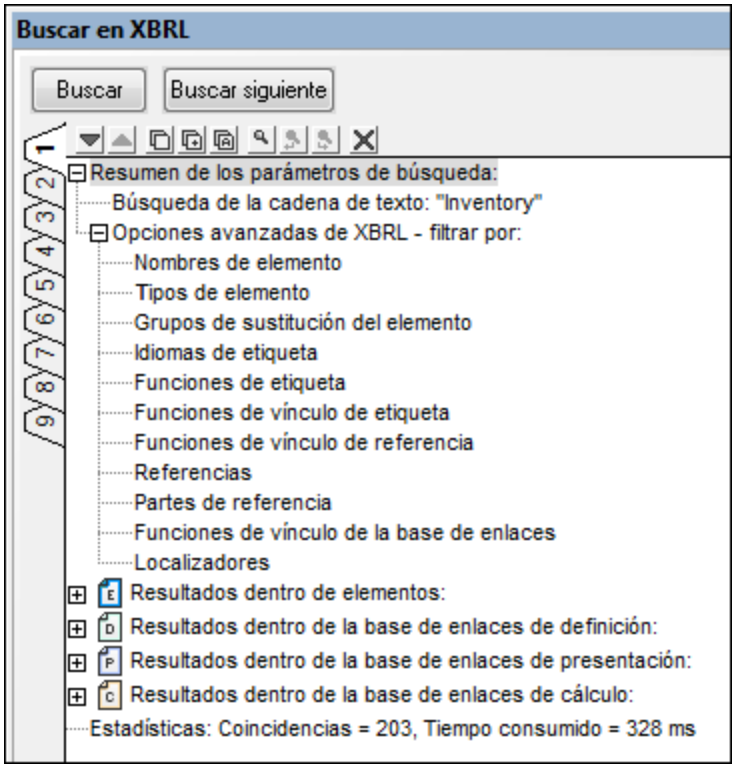
Buscar siguiente

El comando **Buscar siguiente** muestra la siguiente aparición del término de búsqueda en la ventana Buscar en XBRL (*ver imagen*). La búsqueda comienza a partir de la posición del cursor. Puede seguir ejecutando el comando **Buscar siguiente** hasta que se hayan mostrado todas las instancias del término.



Buscar todos

El comando **Buscar todos** muestra todas las instancias del término de búsqueda, así como un resumen de la búsqueda, en la ventana Buscar en XBRL (*ver imagen*).



La ventana de resultados ofrece: (i) un resumen de los tipos de componentes XBRL donde se buscó, (ii) las instancias encontradas del término de búsqueda ordenadas según la base de enlaces y (iii) estadísticas sobre la búsqueda, como el número de instancias encontradas y el tiempo que se necesitó para completar la

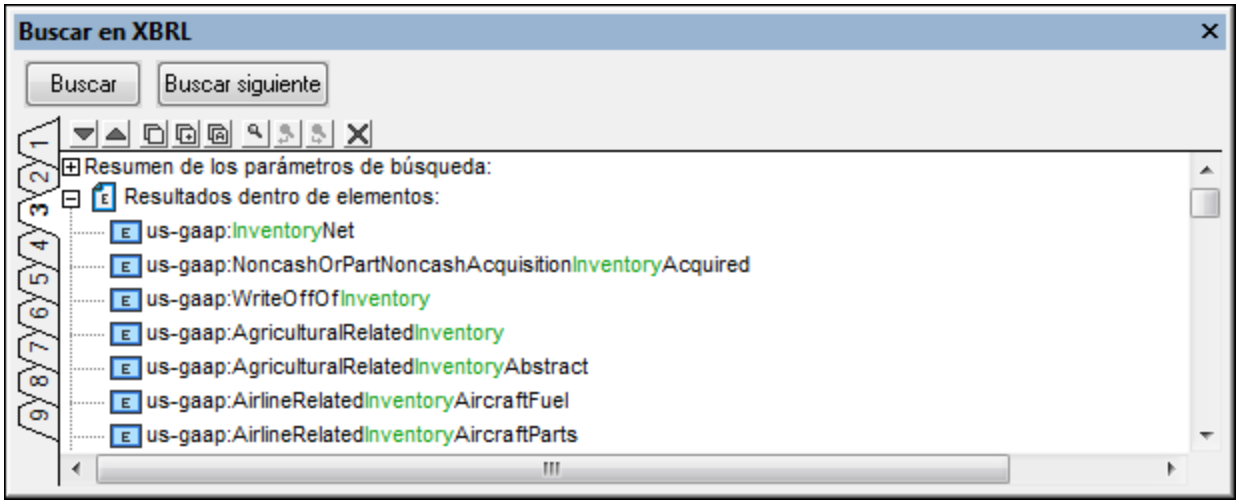
© 2016-2022 Altova GmbH

búsqueda. Puede expandir y contraer los resultados agrupados por base de enlaces. Al hacer clic en un resultado, el elemento correspondiente se resalta en el documento XBRL en la ventana principal.

Para más información sobre las características de la ventana Buscar en XBRL consulte el subapartado Resultados e información 935.

17.7.3 Resultados e información

Cada vez que se ejecuta el comando **Buscar** o **Buscar siguiente**, los resultados de la búsqueda aparecen en la ventana Buscar en XBRL (*ver imagen*). El término de búsqueda aparece en verde. (En la imagen siguiente puede ver que el término de búsqueda fue `Inventory`).



Características de la ventana Buscar en XBRL

Los resultados aparecen en nueve pestañas distintas (numeradas del 1 al 9) para que pueda guardar los resultados de una búsqueda en una pestaña, realizar otra búsqueda en otra pestaña y comparar los resultados. Al hacer clic en un resultado en la ventana Buscar en XBRL, el componente correspondiente se resalta en la ventana principal de la vista XBRL. Así puede usar la ventana Buscar XBRL para buscar un componente y navegar hasta él.

La ventana Buscar en XBRL incluye estos comandos en la barra de herramientas:

- Los iconos **Siguiente** y **Anterior** seleccionan el resultado siguiente o el resultado anterior al resultado que esté seleccionado.
- Los comandos **Copiar mensajes** copian en el portapapeles el mensaje seleccionado, el mensaje seleccionado y sus mensajes subordinados, o todos los mensajes.
- El comando **Buscar** busca cadenas de texto en la ventana Buscar en XBRL.
- El comando **Borrar** elimina todos los mensajes de la pestaña activa.

17.8 OIM

La especificación Open Information Model (OIM) 1.0 proporciona un modelo de sintaxis independiente para datos XBRL, lo que permite transformarlos en representaciones distintas (OIM-XML, OIM-JSON y OIM-CSV). XMLSpy cuenta con varias características que permiten trabajar con OIM:

- Convertir[1452] documentos de datos XBRL en OIM xBRL-JSON y OIM xBRL-CSV (con el menú Conversión[1452]).
- Convertir[1452] de cualquiera de los siguientes tres formatos en uno de los dos restantes: OIM xBRL-XML, OIM xBRL-JSON, and OIM xBRL-CSV (con el menú Conversión[1452]).
- Validar[937] documentos en cualquier formato OIM para que se reconozcan como documentos XBRL y se validen en consecuencia.

© 2016-2022 Altova GmbH

17.9 Validación de instancias y taxonomías XBRL

Para validar una instancia o taxonomía XBRL el documento activo debe ser el documento XBRL que desea validar. Después basta con seleccionar uno de estos métodos de validación:

- **XML | Validar XML (F8)**[1337]: la validación se lleva a cabo con el motor integrado de XMLSpy.
- **XML | Validar XML en el servidor (Ctrl+F8)**[1342]: la validación se lleva a cabo con el servidor remoto RaptorXML+XBRL Server (que puede configurarse previamente con el comando **Herramientas | Gestionar servidores Raptor**[1342])

Puede encontrar las opciones relevantes de validación en la pestaña [1619] *Validación XBRL* [1619] del cuadro de diálogo "Opciones" [1619] (**Herramientas | Opciones**).

18 Office Open XML, ZIP y EPUB

Los archivos Office Open XML (OOXML), ZIP y EPUB son similares, en el sentido de que todos ellos son paquetes que contienen otros archivos. La vista Archivo de XMLSpy incluye una interfaz en la que puede visualizar la estructura interna de estos paquetes, modificar su estructura y editar los archivos del paquete en XMLSpy. En el caso de los archivos EPUB, la vista Archivo también permite ver el libro EPUB directamente en la vista Explorador de XMLSpy.

Office Open XML (OOXML)

OOXML es un formato de archivo para describir documentos, hojas de cálculo y presentaciones. Inicialmente fue desarrollado por Microsoft para el conjunto de programas Microsoft Office, pero ahora es una especificación ECMA.

Estructura de un archivo OOXML

Un documento OOXML es un paquete de varios ficheros que sigue la convención de empaquetado abierto (Open Packaging Convention). Un paquete está formado por ficheros XML, otros ficheros de datos (como archivos de imágenes) y un archivo de relaciones que especifica las relaciones que existen entre los diferentes ficheros del paquete.

La estructura interna, las carpetas internas y el nombre de archivo de un archivo OOXML varía dependiendo del tipo de documento. No obstante, la estructura básica de un archivo OOXML consiste en un fichero XML llamado `[Content_Types].xml` en la raíz de la estructura de directorios y tres directorios: `_rels`, `docProps` y un directorio propio del tipo de documento (en el caso de los documentos `.docx`, por ejemplo, esta carpeta se llamaría `word`, en los documentos `.xslx` se llamaría `xl` y en los documentos `.pptx` se llamaría `ppt`).

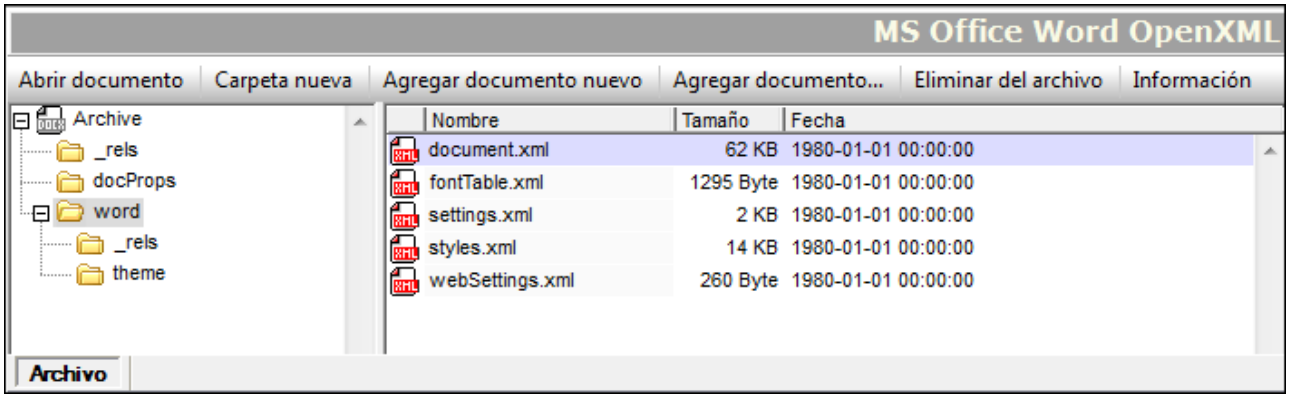
```
Archivo OOXML
|-- Fichero:         [Content_Types].xml
|-- Carpeta:  _rels
|-- Carpeta:  docProps
|-- Carpeta:  word/xl/ppt
```

- La carpeta `_rels` contiene un fichero llamado `rels.xml`, que especifica las relaciones que existen entre los ficheros del paquete.
- La carpeta `docProps` contiene los ficheros `app.xml` y `core.xml`, que describen propiedades clave del documento.
- Las carpetas `word`, `xl` y `ppt` contienen ficheros XML que almacenan el contenido del documento. Por ejemplo, en la carpeta `word` el fichero `document.xml` incluye el contenido principal del documento.

Archivos OOXML en la vista Archivo de XMLSpy

En la vista Archivo de XMLSpy (*imagen siguiente*) puede visualizar y editar el contenido de archivos OOXML.

© 2016-2022 Altova GmbH



La vista de carpetas situada a la izquierda de la vista muestra las carpetas del paquete, mientras que la ventana principal muestra los ficheros de la carpeta seleccionada en la vista de carpetas. En la vista Archivo puede agregar y eliminar carpetas y ficheros del archivo. También puede abrir los ficheros para editarlos en XMLSpy con sólo hacer doble clic en el fichero.

Edición inteligente de los ficheros internos de un archivo OOXML

Los documentos XML que forman parte de paquetes OOXML están basados en esquemas estándar. XMLSpy ofrece funciones de edición inteligente para trabajar con documentos OOXML, como ayudantes de entrada, una función de finalización automática y una función de validación.

Archivos ZIP

Los archivos ZIP archivan varios ficheros en un paquete de compresión sin pérdida de datos. Estos ficheros pueden ser de varios tipos. En la vista Archivo de XMLSpy puede crear archivos ZIP, modificar su estructura interna y editar los ficheros del archivo. Estas operaciones se describen en el apartado Archivos ZIP 943 de esta sección.

Archivos EPUB

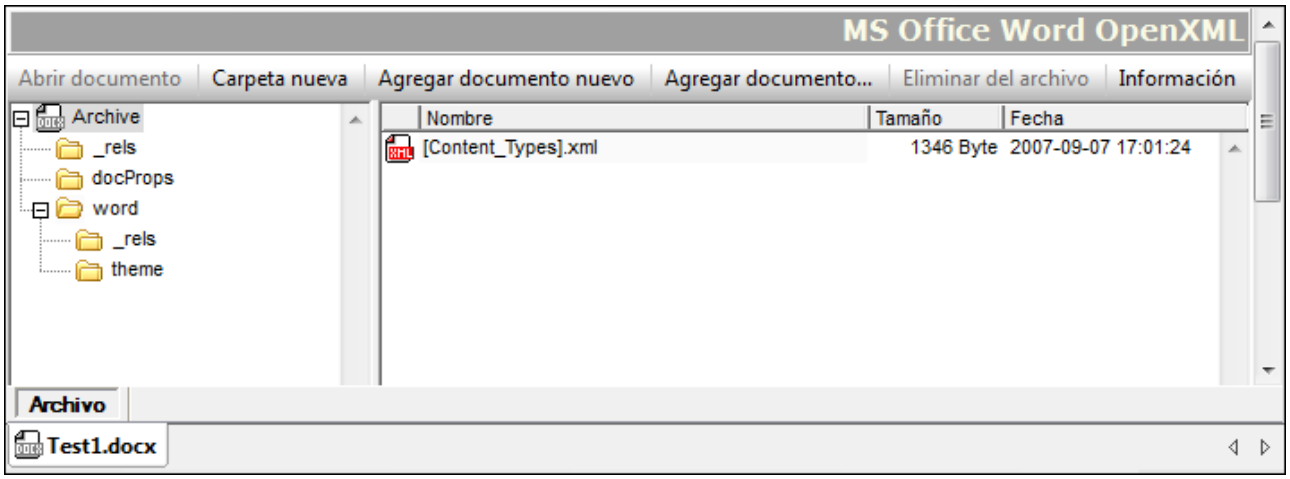
Un archivo EPUB es un grupo comprimido de ficheros utilizado para la distribución de publicaciones digitales (libros EPUB). En la vista Archivo 281 puede abrir, crear y editar archivos EPUB, ver una vista previa del libro EPUB, editar los ficheros que componen el archivo EPUB en XMLSpy, validar el archivo EPUB y guardar otra vez los ficheros en el archivo EPUB. Para más información consulte el apartado Archivos EPUB 945.

18.1 Trabajar con archivos OOXML

En este apartado describimos cómo se trabaja con documentos OOXML en la vista Archivo. En concreto, explicamos estos procedimientos:

- Crear, abrir y guardar archivos OOXML 940
- Editar la estructura de un archivo OOXML 940
- Abrir, editar y guardar documentos OOXML internos 941
- Edición inteligente de documentos OOXML internos 941
- Direccionar documentos en archivos OOXML 941
- Comparar archivos OOXML 940

Crear, abrir y guardar archivos OOXML



Para abrir un archivo OOXML en la vista Archivo use el cuadro de diálogo "Abrir" (**Archivo | Abrir**) de XMLSpy. Para guardar un archivo OOXML use el comando **Archivo | Guardar (Ctrl+S)**. Este comando guarda la estructura y las relaciones del archivo OOXML.

Editar la estructura de un archivo OOXML

El contenido de un archivo OOXML se puede modificar agregando y eliminando carpetas y documentos en el archivo OOXML desde la vista Archivo 281. Después de realizar estos cambios estructurales, es necesario guardar el archivo OOXML (**Archivo | Guardar**) para que los cambios surtan efecto. Es necesario tener en cuenta que:

- Cuando se añade una carpeta o un documento nuevo con los botones de la vista Archivo 281 es necesario dar un nombre a la carpeta o documento inmediatamente. El motivo es que en la vista Archivo no se pueden renombrar carpetas ni documentos.
- Después de añadir un documento nuevo a una carpeta del archivo, el documento se guarda en el archivo tanto si se guarda en su propia ventana como si se guarda el archivo OOXML.

© 2016-2022 Altova GmbH

Abrir, editar y guardar documentos OOXML internos

Para abrir un documento OOXML interno (es decir, un documento contenido en un paquete OOXML) haga doble clic en el documento en la vista Archivo.

Otra opción es seleccionar el documento en la ventana principal y pulsar el botón Abrir documento[281]. El documento se abre en una ventana nueva de XMLSpy. Cuando guarde el documento, este se guarda en el archivo OOXML y, por tanto, no hace falta guardar el archivo OOXML también.

Edición inteligente de documentos OOXML internos

XMLSpy ofrece funciones de edición inteligente para documentos OOXML internos, es decir, documentos contenidos en un paquete OOXML. Entre estas funciones destacan los ayudantes de entrada, la función de finalización automática y la de validación.

Direccionar documentos de archivos OOXML

Puede direccionar documentos de archivos OOXML con rutas de acceso de archivo normales más un carácter de barra vertical. Por ejemplo, la ruta de acceso de archivo

```
C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Office20XX\ExcelDemo.xlsx|zip\xl\tables\table1.xml
```

localiza el documento `table1.xml`, que está en la carpeta `xl\tables` del archivo OOXML `ExcelDemo.xlsx` ubicado en la carpeta `Examples\Office20XX` de la carpeta de ejemplos de XMLSpy.

Comparar archivos OOXML

Puede comparar un archivo OOXML que esté abierto en la vista Archivo con otro archivo OOXML. Para ello use el comando **Herramientas | Comparar directorios**[1551].

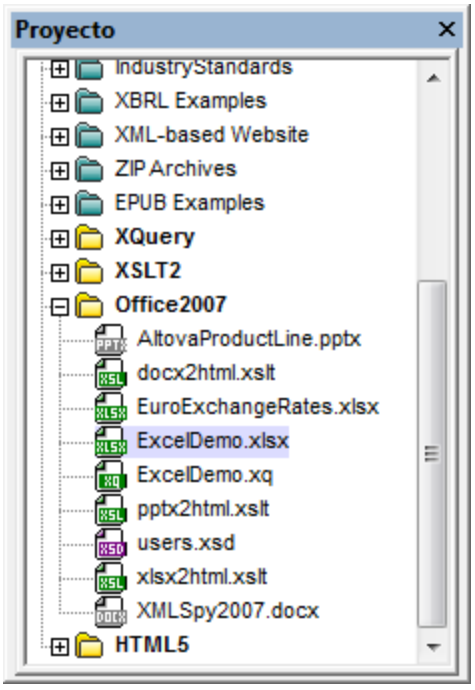
18.2 Archivos OOXML de ejemplo

En la carpeta `Examples\Office` de la carpeta de aplicación de XMLSpy encontrará estos archivos de ejemplo:

- 3 archivos OOXML: (i) un archivo Word Open XML (`.docx`), (ii) un archivo Excel Open XML (`.xlsx`) y (iii) un archivo PowerPoint Open XML (`.pptx`)
- 3 archivos XSLT: (i) `docx2html.xslt` (para convertir el archivo `.docx` de muestra en HTML), (ii) `xslx2html.xslt` (para convertir el archivo `.xslx` de muestra en HTML) y (iii) `pptx2html.xslt` (para convertir el archivo `.pptx` de muestra en HTML)
- 1 archivo XQuery: `ExcelDemo.xq` (para recuperar datos del archivo `.xslx`)

Los archivos XSLT y XQuery se ofrecen para que aprenda a usar archivos XSLT y XQ para acceder a datos de archivos OOXML y transformarlos. Para ejecutar los documentos XSLT y XQuery tiene varias opciones:

- Puede abrir el archivo OOXML en la vista Archivo. En la vista de carpetas seleccione `Archive` y después seleccione el comando de menú **XSL/XQuery | Transformación XSL** (para realizar una transformación XSLT) o **XSL/XQuery | Ejecutar XQuery** (para ejecutar una consulta XQuery). Después busque el archivo XSLT o XQuery y haga clic en **Aceptar**.
- Puede abrir el proyecto `Examples` en XMLSpy. En la ventana Proyecto haga clic con el botón derecho en el archivo `.xlsx` o `.docx` de la carpeta `Office2007` (*ver imagen*) y seleccione el comando de transformación correspondiente. Busque el archivo de transformación y haga clic en **Aceptar**.



- Puede abrir el archivo XSLT o XQuery en XMLSpy y seleccionar el comando de menú **XSL/XQuery | Transformación XSL** y **XSL/XQuery | Ejecutar XQuery** respectivamente. Cuando se le pida el archivo XML que desea transformar, busque el archivo `.docx`, `.xlsx` o `.pptx` (dependiendo de si el documento XSLT/XQ es para Word, Excel o PowerPoint).

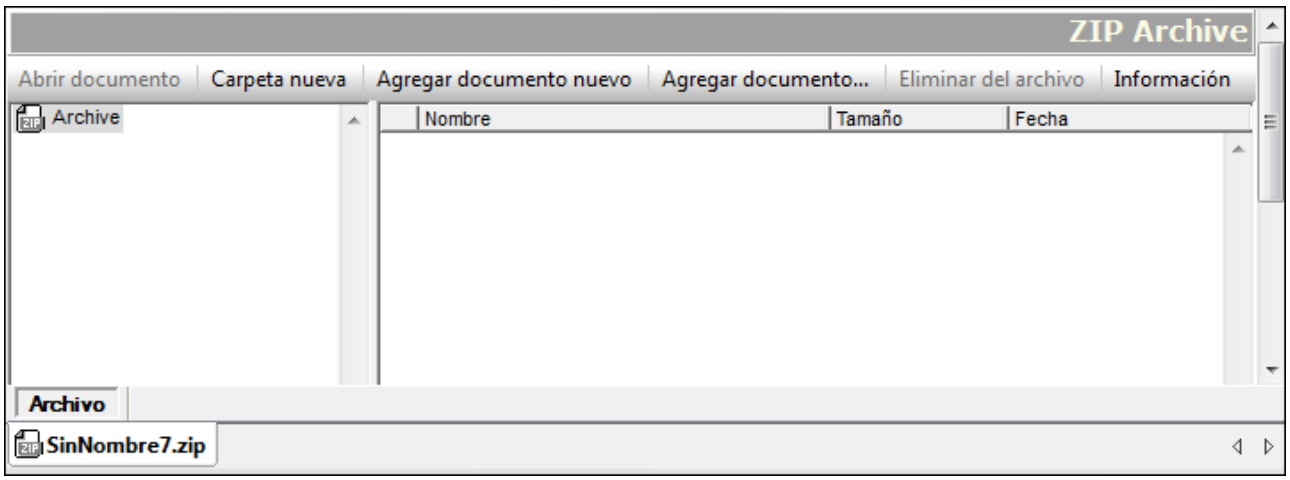
© 2016-2022 Altova GmbH

18.3 Archivos ZIP

En la vista Archivo [281] puede crear archivos WinZip, modificar la estructura interna de archivos ZIP (WinZip, WinRAR, etc.), editar ficheros de paquetes ZIP en XMLSpy y guardarlos otra vez en el archivo ZIP.

Crear y guardar un archivo WinZip

Para crear un archivo WinZip abra el cuadro de diálogo "Crear documento nuevo" (comando **Archivo | Nuevo**) y seleccione el tipo de archivo .zip. Esto crea un archivo WinZip nuevo en una ventana nueva de XMLSpy (*imagen siguiente*). Ahora guarde el archivo ZIP con el comando **Archivo | Guardar (Ctrl+S)**. Añada carpetas y archivos como se describe a continuación y después guarde el archivo ZIP para conservar esos cambios.



Para abrir un archivo ZIP que en la vista Archivo use el cuadro de diálogo "Abrir" de XMLSpy (**Archivo | Abrir**).

Nota: hay una diferencia entre crear un archivo ZIP nuevo y crear un archivo OOXML nuevo. Cuando se crea un archivo ZIP no se le pide que lo guarde antes de abrirlo en la vista Archivo. Para guardar un archivo ZIP recién creado es necesario usar el comando **Archivo | Guardar (Ctrl+S)**.

Agregar carpetas y ficheros y modificar la estructura del archivo

En la vista Archivo puede agregar carpetas (pulse el botón **Carpeta nueva**) y agregar ficheros ya existentes (**Agregar documento...**) o ficheros nuevos (**Agregar documento nuevo** a una carpeta del archivo Recuerde que si quiere renombrar la carpeta o el documento nuevo recién agregado es necesario hacerlo inmediatamente.

Direccionar documentos de archivos ZIP

Puede direccionar documentos de archivos ZIP con rutas de acceso de archivo normales más el carácter de barra vertical. Por ejemplo la ruta de acceso de archivo

```
C:\Documents and Settings\<usuario>\Mis Documentos\Altova\XMLSpy2023\Examples\Test.zip|zip\TestFolder\MyFile.xml
```

localiza el documento `MyFile.xml`, que está en la carpeta `TestFolder` del archivo ZIP `Test.zip` ubicado en la carpeta `Examples` de la carpeta de aplicación de XMLSpy.

Comparar archivos ZIP

Puede comparar un archivo ZIP que esté abierto en la vista Archivo con otro archivo ZIP. Para ello use el comando **Herramientas | Comparar directorios**[1551].

© 2016-2022 Altova GmbH

18.4 Archivos EPUB

Un archivo EPUB es un grupo comprimido de ficheros que se ajusta al estándar EPUB del International Digital Publishing Forum (IDPF). Se trata de un estándar de distribución e intercambio de publicaciones web digitales. En la vista Archivo [281] puede abrir archivos EPUB, ver la publicación digital del archivo EPUB en una pestaña de vista previa, editar los ficheros que forman parte del archivo EPUB en XMLSpy, validar el archivo EPUB y guardar otra vez los ficheros en el archivo EPUB.

Nota: (i) XMLSpy es compatible con EPUB 2.0.1. (ii) en el proyecto de ejemplo `Examples` que viene con XMLSpy y en la carpeta de ejemplos `(Mis) Documentos/Altova/XMLSpy2023/Examples` encontrará un archivo EPUB de muestra.

Terminología

En este apartado se usan estos términos:

- ***Archivo EPUB*** se usa en referencia al archivo EPUB que tiene la extensión de archivo `.epub`. Se trata de un archivo ZIP que contiene todo el archivo y es el archivo que se abrirá en la vista Archivo
- ***Archivo de almacenamiento*** se refiere a cada uno de los ficheros que contiene el archivo EPUB
- ***Libro EPUB*** es el término que denomina la publicación digital generada a partir del archivo EPUB comprimido

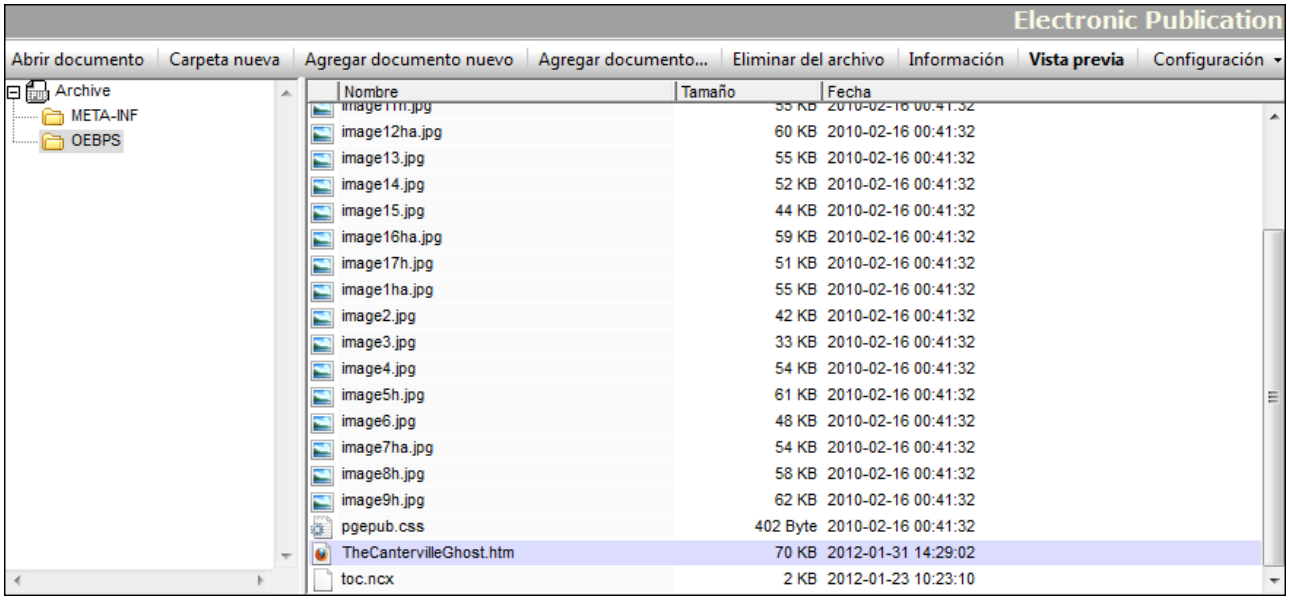
Contenido de esta sección

Este apartado describe las funciones de XMLSpy compatibles con EPUB y se divide en varias partes:

- Abrir archivos EPUB en la vista Archivo [945]
- Crear un archivo EPUB nuevo [946]
- Ver una vista previa de un libro EPUB [946]
- Modificar el contenido y la estructura de un archivo EPUB [947]
- Información y configuración [947]
- Editar archivos de almacenamiento en XMLSpy [948]
- Ayudantes de entrada para archivos de almacenamiento [948]
- Validar un archivo EPUB [948]

Abrir archivos EPUB en la vista Archivo

Seleccione el comando de menú **Archivo | Abrir**, navegue hasta el archivo EPUB y haga clic en **Abrir**. El archivo EPUB se abre en la vista Archivo (*ver imagen*). Otra opción es hacer clic con el botón derecho en el archivo EPUB en el explorador de Windows y abrir el archivo con XMLSpy. Si configuró XMLSpy como editor predeterminado para archivos EPUB [1587], entonces puede hacer doble clic en el archivo EPUB y este se abrirá en la vista Archivo.



La vista de carpetas situada en el panel izquierdo muestra las carpetas del archivo, mientras que la ventana principal enumera los ficheros que contiene la carpeta seleccionada. El archivo EPUB tiene esta estructura y estos componentes principales:

```
Archivo
|-- Archivo Mimetype
|
|-- Carpeta META-INF
|     |-- container.xml
|
|-- Carpeta de DOCUMENTO (en la imagen anterior OEBPS es la carpeta de documento)
      |-- Contiene ficheros HTML, CSS, de imagen y ficheros OPF y NCX
```

Crear un archivo EPUB nuevo

Para crear un archivo EPUB nuevo, seleccione el comando de menú **Archivo | Nuevo**. Aparece el cuadro de diálogo "Crear documento nuevo". Seleccione el tipo de archivo `.epub` y pulse Aceptar. Aparece el cuadro de diálogo "Guardar como". Escriba el nombre del documento EPUB y pulse **Guardar**. Como resultado se abre una ventana nueva de la vista Archivo y se crea un archivo EPUB básico con todas las carpetas y los ficheros de un archivo EPUB válido (*vea la estructura anterior*). A continuación puede añadir carpetas y ficheros (vea las instrucciones que aparecen más abajo) y guardar el archivo EPUB. Para editar un archivo de almacenamiento en XMLSpy, haga doble clic en el archivo en la vista Archivo. El archivo de almacenamiento se abre en una ventana nueva de XMLSpy. Edítelo y guárdelo (Archivo **| Guardar** o Ctrl+S).

Ver una vista previa de un libro EPUB

Para ver una vista previa de un libro EPUB, abra el archivo EPUB en la vista Archivo y haga clic en el botón **Vista previa** de la barra de herramientas. El libro EPUB se abre en una ventana de Internet Explorer dentro de XMLSpy. Si algún fichero necesario para crear la vista previa (ya sea un fichero de contenido o estructural) se modificó pero no se guardó, se le pedirá que guarde el fichero. Si no guarda los cambios, la vista previa usará datos previos y no ofrecerá la versión más reciente del archivo. Si lo prefiere, puede configurar XMLSpy para

© 2016-2022 Altova GmbH

que todos los archivos modificados se guarden automáticamente antes de crear la vista previa (pulse el botón **Configuración** de la barra de herramientas de la vista Archivo y elija una opción).

Recuerde que:

- Si pulsa el botón **Vista previa** de la vista Archivo mientras tiene abierta una ventana de vista previa del mismo archivo EPUB, el archivo EPUB se vuelve a cargar en la ventana de vista previa actual.
- Si actualiza la ventana de vista previa (con el comando **Actualizar (F5)** de Internet Explorer), la vista previa del archivo EPUB no se actualiza. Si actualiza la ventana de vista previa (con el comando Actualizar (F5) de Internet Explorer), la vista previa del archivo EPUB no se actualiza. Es necesario pulsar el botón **Vista previa** de la vista Archivo (*ver punto anterior*.
- Para cerrar la vista previa, cierre la ventana de vista previa.

Nota: no todos los formatos EPUB son compatibles con Internet Explorer. Si el formato EPUB no es compatible con IE, la vista previa puede estar distorsionada. Además, si la publicación digital es XML y no HTML, puede que la vista previa no funcione. Las versiones más recientes de Internet Explorer funcionan mejor con los formatos EPUB. Por ello recomendamos que actualice su versión de Internet Explorer.

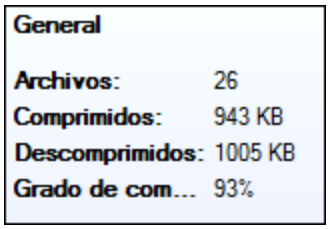
Modificar el contenido y la estructura de un archivo EPUB

Puede agregar carpetas (pulse el botón Carpeta nueva), ficheros nuevos (Agregar documento nuevo) y ficheros ya existentes (Agregar documento...) a una carpeta del archivo. Recuerde que si quiere renombrar el directorio o documento recién agregado es necesario hacerlo inmediatamente. En la vista Archivo no se pueden renombrar directorios ni documentos. Para eliminar un fichero o directorio del archivo, seleccione el directorio o fichero y pulse el botón **Eliminar del archivo**.

Cuando termine de realizar cambios, guarde el archivo EPUB (**Archivo | Guardar**).

Información y configuración

Al hacer clic en el botón **Información** en la parte inferior de la vista Archivo aparece información importante sobre el archivo (*imagen siguiente*). Si hace clic otra vez en el botón **Información**, la información desaparece. Esta información incluye el número de ficheros que contiene el archivo (incluidos el archivo MimeType y `container.xml`), el tamaño del archivo EPUB comprimido y el tamaño de total de los archivos descomprimidos.



Al hacer clic en el botón **Configuración** aparece un menú con dos opciones que se pueden activar o desactivar. Una opción permite guardar el archivo EPUB automáticamente (i) antes de validarlo y la otra permite guardar el archivo EPUB automáticamente (ii) antes de crear una vista previa con el botón **Vista previa**.

Editar archivos de almacenamiento en XMLSpy

Para editar un archivo de almacenamiento en XMLSpy, haga doble clic en el archivo en la vista Archivo. Otra opción es seleccionar el archivo en la vista Archivo y pulsar el botón **Abrir documento** de la barra de herramientas. El archivo se abre en una ventana nueva de XMLSpy. Edítelo y guárdelo (Archivo | **Guardar** o Ctrl+S).

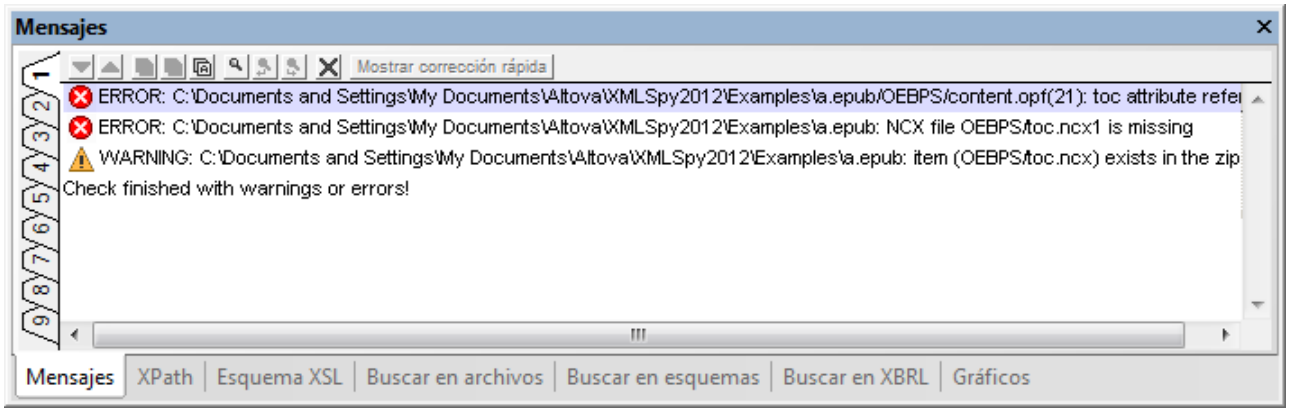
Ayudantes de entrada para archivos de almacenamiento

XMLSpy ofrece ayudantes de entrada para ciertos archivos de almacenamiento estándar. Estos archivos de almacenamiento son:

- El archivo OPF, que suele llamarse `content.opf`. Este archivo contiene los metadatos del libro EPUB. Está basado en la especificación de formato de empaquetado abierto (OPF).
- El archivo NCX (Archivo de control de navegación para XML), que suele llamarse `toc.ncx`. Este archivo contiene la tabla de contenido de la publicación. Está basado en la parte NCX de la especificación OPF.
- La carpeta `META-INF` debe incluir el archivo `container.xml`, que apunta al archivo que define el contenido del libro (el archivo OPF). El archivo `container.xml` indica cómo se deben organizar los archivos de almacenamiento de acuerdo con las reglas de la especificación de formato de empaquetado abierto (OPF).

Validar un archivo EPUB

Para validar un archivo EPUB, seleccione el comando **XML | Validar el documento XML (F8)**. Los resultados de la validación aparecen en la ventana Mensajes (*imagen siguiente*). Si algún archivo de almacenamiento está sin guardar, se le pedirá que guarde el archivo antes de validar el archivo EPUB. Para validar el archivo EPUB es necesario guardar los archivos modificados. Si lo prefiere, puede configurar XMLSpy para que todos los archivos modificados se guarden automáticamente antes de crear la vista previa (pulse el botón **Configuración** de la barra de herramientas de la vista Archivo y elija una opción).



En los mensajes de error se indica: (i) en qué archivo se encontró el error y en qué número de línea, si procede, y (ii) una descripción del error. En la imagen anterior, el error resaltado se encontró en la línea 21 del archivo `content.opf`. Al hacer clic en un error de la ventana Mensajes, el archivo pertinente se abre y el error aparece resaltado.

© 2016-2022 Altova GmbH

Nota: el motor de validación EPUB es una utilidad Java. Por tanto, para que funcione el motor de validación es necesario tener instalado Java en el equipo.

19 Bases de datos

Con XMLSpy puede conectarse a varios tipos de bases de datos (BD) y realizar varias operaciones: consultar la BD, importar la estructura de la BD como esquema XML, generar un archivo XML a partir de la BD, exportar datos a la BD, etc. Cada una de estas operaciones cuentan con un comando de menú en XMLSpy y se describen, comando por comando, en la sección *Comandos de menú*[1257] de la documentación. Más abajo aparece una lista de los comandos, con hipervínculos a su descripción.

En esta sección:

- Explicamos cómo se puede conectar a una BD[952]. Antes de ejecutar los comandos para BD de XMLSpy es necesario conectarse a una BD.
- Enumeramos las BD[1040] que son compatibles con XMLSpy.

Nota: si usa la versión de 64 bits de XMLSpy asegúrese de que tiene acceso a los controladores de bases de datos de 64 bits necesarios para el tipo de BD al que desea conectarse.

Funciones de XMLSpy para bases de datos

Las funciones de XMLSpy para trabajar con bases de datos se ejecutan con los comandos de los menús Base de datos[1411] y Conversión[1452].

- *Consultar la base de datos*[1423]*:* comando del menú **Base de datos**. Carga la estructura de la BD en una ventana de consulta de BD, donde puede consultar la BD. Los resultados aparecen en la ventana de consulta.
- *IBM DB2*[1439]*:* comando del menú **Base de datos**. IBM DB2 es una BD XML y en XMLSpy puede gestionar los esquemas XML de la BD XML, así como editar la BD XML y validarla.
- *SQL Server*[1445]*:* comando del menú **BD**. XMLSpy permite gestionar los esquemas XML de la BD e incluye funciones de edición y de validación.
- *Oracle XML DB*[1447]: comando del menú **Base de datos**. Ofrece varias funciones para trabajar con BD XML de Oracle XML, como gestionar esquemas XML, consultar la BD y generar archivos XML a partir del esquema de la BD.
- *Importar datos de una base de datos*[1455]: comando del menú **Conversión**. Importa datos de BD en un archivo XML.
- *Crear esquema XML a partir de la estructura de la BD*[1461]: comando del menú **Conversión**. Genera un esquema XML que está basado en la estructura de la BD.
- *Importar BD basada en el esquema XML*[1466]: comando del menú **Conversión**, disponible si el documento activo es un esquema XML. Se establece una conexión con una BD y los datos de la tabla de BD seleccionada se pueden importar. La estructura del documento XML resultante está basada en el esquema XML que estaba activo al establecer la conexión con la BD.
- *Crear estructura de la BD a partir del esquema XML*[1467]: comando del menú **Conversión**. Se crean tablas de BD sin datos basadas en la estructura de un esquema XML ya existente.
- *Exportar a una base de datos*[1474]: comando del menú **Conversión**. Los datos de un documento XML se pueden exportar a una BD. Las tablas de BD existentes se pueden actualizar con los datos XML o se pueden crear tablas nuevas que contengan los datos XML.

Conversiones de tipos de datos

Al realizar conversiones de datos de documentos XML y bases de datos, es necesario convertir los tipos de datos a tipos correspondientes del formato de destino. Para más información sobre el método utilizado por

© 2016-2022 Altova GmbH

XMLSpy para convertir tipos de datos, consulte los anexos Tipos de datos en esquemas XML generados a partir de BD[2233] y Tipos de datos en BD generadas a partir de esquemas XML[2240].

Altova DatabaseSpy

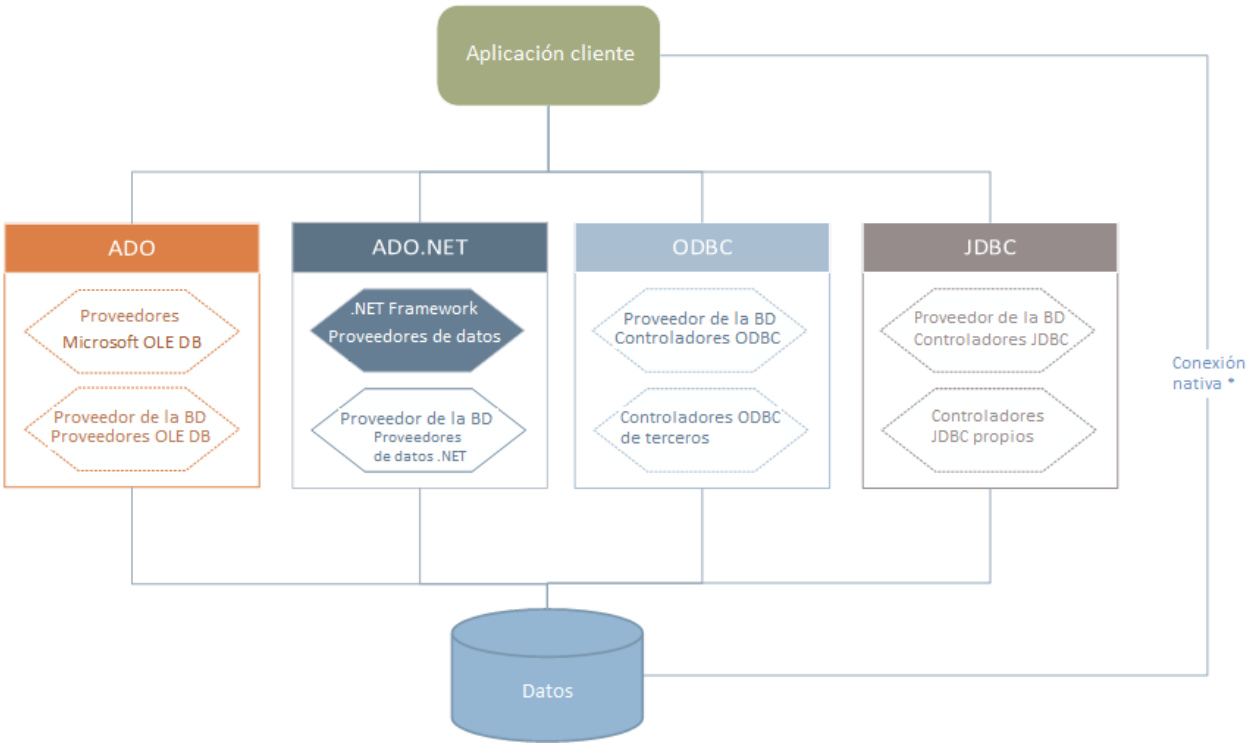
Altova DatabaseSpy es una herramienta de diseño y consulta de bases de datos compatible con varios tipos de datos que ofrece muchas otras funciones que no encontrará en XMLSpy. Para más información sobre Altova DatabaseSpy, visite el sitio web de Altova.

19.1 Conectarse a un origen de datos

En su definición más sencilla, una base de datos es un archivo local como un archivo de base de datos Microsoft Access o SQLite. En casos más complejos, una base de datos puede residir en un servidor de bases de datos remoto o de la red que no tienen por qué usar el mismo sistema operativo que la aplicación que se conecta a la BD y que consume los datos. Por ejemplo, mientras que XMLSpy se puede ejecutar en sistemas operativos Windows, puede que la base de datos a la que desea acceder (p. ej. una base de datos MySQL) esté en un equipo Linux.

Para interactuar con los diferentes tipos de bases de datos, XMLSpy se sirve de las interfaces de conexión de datos y los controladores de BD disponibles en su sistema operativo o publicados por los principales proveedores de BD. La tecnología de base de datos evoluciona constantemente y por tanto consideramos que este mecanismo garantiza la mayor compatibilidad y flexibilidad en las principales plataformas.

En el siguiente diagrama puede ver un resumen de las opciones de conectividad de datos que pueden existir entre XMLSpy (*aplicación cliente*) y un almacén de datos (que puede ser un archivo o un servidor de base de datos).



** Las bases de datos SQLite, PostgreSQL, CouchDB y MongoDB admiten conexiones nativas directas. Para conectarse a este tipo de bases de datos no es necesario tener instalado controladores específicos.*

Por tanto, tal y como muestra el diagrama, desde XMLSpy puede acceder a los principales tipos de BD con estas tecnologías de acceso de datos:

- ADO (Microsoft® ActiveX® Data Objects), que a su vez utiliza OLE DB
- ADO.NET (un conjunto de bibliotecas disponibles en Microsoft .NET Framework que permiten la interacción con datos)

© 2016-2022 Altova GmbH

- JDBC (Java Database Connectivity)
- ODBC (Open Database Connectivity)

Nota: algunos proveedores de ADO.NET no son compatibles o tienen una compatibilidad limitada. Consulte el apartado *Notas sobre compatibilidad con ADO.NET* 971 para obtener más información.

Tecnologías de acceso a datos

La interfaz de conexión de datos que se debe utilizar dependerá de la infraestructura de software con la que cuente. Lo normal es elegir una tecnología de acceso a datos y un controlador de base de datos que se integre bien con el sistema de base de datos al que desea conectarse. Por ejemplo, para conectarse con una base de datos Microsoft Access 2013, puede generar una cadena de conexión ADO que utilice un proveedor nativo como **Microsoft Office Access Database Engine OLE DB Provider**. Para conectarse a Oracle, por su parte, lo mejor será descargar e instalar las interfaces JDBC, ODBC o ADO .NET más recientes del sitio web de Oracle.

Lo más probable es que los controladores de los productos Windows (como Microsoft Access o SQL Server) ya estén disponibles en su sistema, pero puede que necesite descargar controladores para otros tipos de bases de datos. Los principales proveedores de BD publican software cliente y controladores con frecuencia. Además, puede encontrar otros controladores de otras organizaciones para las tecnologías de acceso de datos mencionadas. En la mayoría de los casos hay varias maneras de conectarse a una base de datos. Las características y el rendimiento de la base de datos dependerán de la tecnología de acceso o de los controladores utilizados

19.1.1 Iniciar el asistente para la conexión de base de datos

XMLSpy cuenta con un asistente que le guiará durante la conexión a un origen de datos. Antes de empezar a seguir las instrucciones del asistente, debe tener en cuenta que algunos tipos de BD requieren instalar y configurar algunos elementos, como controladores o software cliente. Normalmente este tipo de productos se obtienen del proveedor de BD correspondiente, que los acompañan de la documentación pertinente específica para su versión de Windows. Para ver una lista de controladores de base de datos agrupados por tipo de BD consulte el apartado Resumen de controladores de base de datos 955.

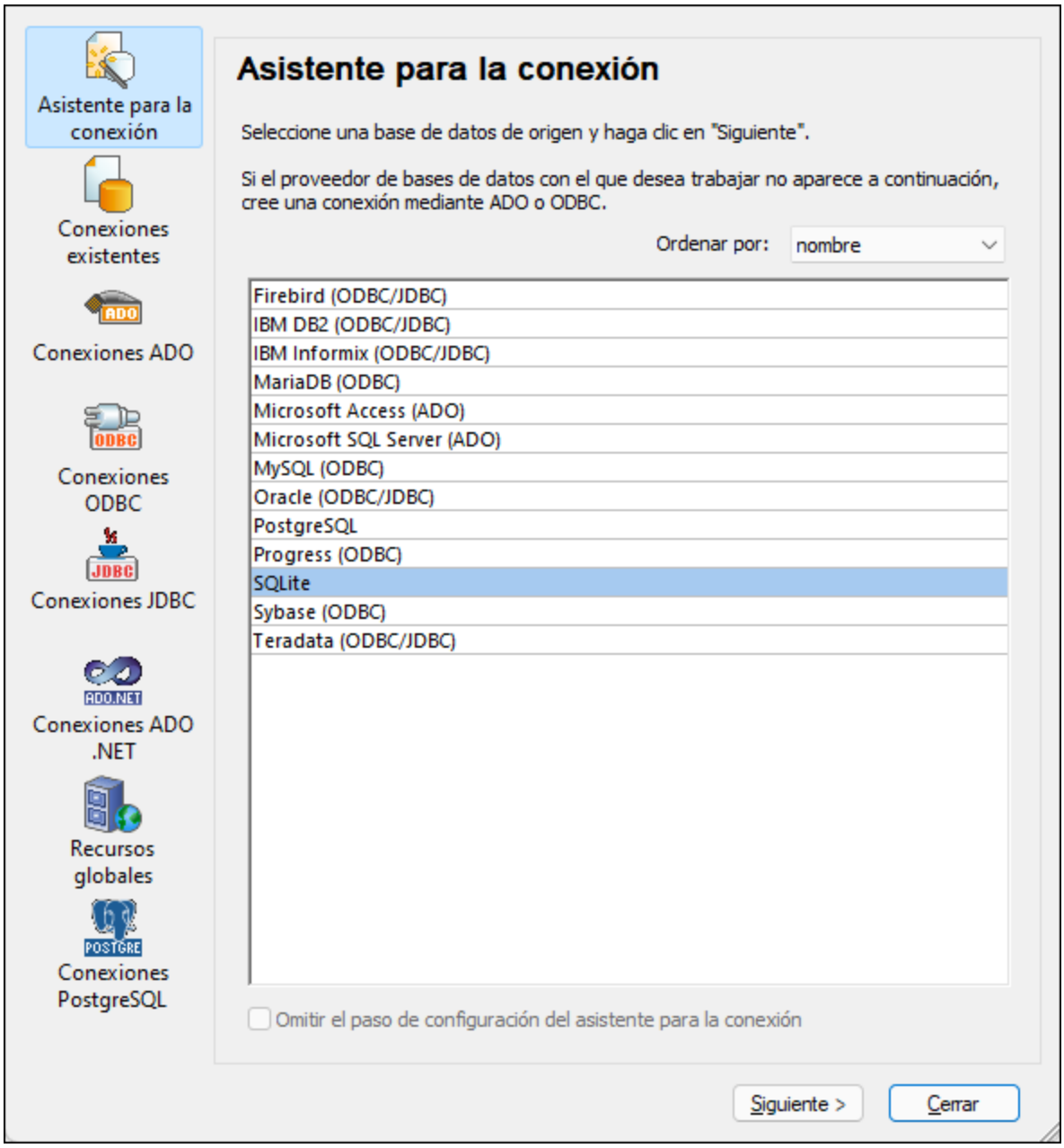
Para iniciar el asistente para la conexión de BD (*imagen siguiente*) siga estos pasos:

- En el menú **Base de datos** haga clic en **Consultar la base de datos**.

Con estos pasos ha iniciado correctamente el asistente para la conexión a bases de datos (*imagen siguiente*). En la parte izquierda de la ventana puede seleccionar la forma de conectarse a la BD que prefiera:

- Con el asistente para la conexión, que le pide que elija un tipo de base de datos y después le indica qué pasos seguir para conectarse a ese tipo de BD
- Con una conexión que ya existe
- Con uno de estos tipos de tecnología de acceso de datos: ADO, ADO.NET, ODBC o JDBC
- Con un recurso global de Altova en el que se haya guardado una conexión de BD
- Con una conexión PostgreSQL nativa

En el panel del asistente para la conexión (*imagen siguiente*) las bases de datos se pueden ordenar alfabéticamente por tipo de BD o por uso reciente. Seleccione la opción que prefiera en el cuadro combinado *Ordenar*. Una vez haya seleccionado el tipo de BD haga clic en **Siguiente**.



Siga las instrucciones que aparecen en pantalla, que dependerán del tipo de BD, de la tecnología de acceso (ADO, ADO.NET, ODBC, JDBC) y del controlador utilizados. Para ver ejemplos de conexión de cada tipo de BD consulte el apartado Ejemplos de conexión a bases de datos[985].

También puede consultar estos apartados:

- Conexiones ADO[958]
- Conexiones ADO.NET[964]

© 2016-2022 Altova GmbH

- Conexiones ODBC 971
- Conexiones JDBC 974

19.1.2 Resumen de controladores de base de datos

En la tabla que aparece más abajo puede ver una lista de controladores de BD que puede utilizar para conectarse a una base de datos a través de las diferentes tecnologías de acceso de datos. Esta lista no contiene todos los controladores disponibles en la actualidad y, por tanto, puede usar otros controladores que no aparecen en la lista.

Aunque puede que Windows venga con algunos controladores de BD, es posible que necesite descargar más controladores. Por lo general, se recomienda utilizar el controlador más reciente publicado por el proveedor de la BD.

Los proveedores de bases de datos pueden ofrecer controladores para descargar en paquetes o junto con el software cliente de base de datos. En este último caso, el cliente de base de datos suele incluir todos los controladores necesarios u ofrecer una opción durante la instalación para seleccionar los controladores y componentes que el usuario desee instalar. El software cliente de base de datos suele incluir funciones de administración y configuración que permiten simplificar la administración de la BD y documentación que explica cómo instalar y configurar el cliente y sus componentes.

Es muy importante configurar bien el cliente de BD para establecer correctamente la conexión con la BD. Antes de instalar y usar el software cliente de BD, recomendamos que lea detenidamente las instrucciones de instalación y configuración porque pueden variar según la versión de la BD y según la versión de Windows.

Si desea conocer las características y limitaciones de cada tecnología de acceso de datos con respecto a cada tipo de BD, consulte la documentación de la BD correspondiente y pruebe la conexión. Debe tener en cuenta estos puntos para evitar problemas de conexión:

- Algunos proveedores de ADO.NET no son compatibles o tienen una compatibilidad limitada (*véase* *Notas sobre compatibilidad con ADO.NET* 971).
- Cuando instale un controlador de BD, recomendamos que tenga la misma plataforma que la aplicación de Altova (32 o 64 bits). Por ejemplo, si usa una aplicación de Altova de 32 bits en un sistema operativo de 64 bits, instale el controlador de 32 bits y configure la conexión de base de datos con ayuda del controlador de 32 bits (*véase* *Ver los controladores ODBC disponibles* 973).
- Cuando configure un origen de datos ODBC, recomendamos crear el nombre de origen de datos (DSN) como DSN del sistema en lugar de como DSN de usuario (*véase* *Conexiones ODBC* 971).
- Cuando configure un origen de datos JDBC, compruebe que tiene instalado JRE (Java Runtime Environment) o Java Development Kit (JDK) y que la variable de entorno CLASSPATH del sistema operativo está configurado (*véase* *Conexiones JDBC* 974).
- Si necesita consultar las instrucciones de instalación e información sobre compatibilidad de los controladores o clientes de BD, consulte la documentación que viene con el paquete de instalación.

Base de datos	Interfaz	Controladores
Firebird	ADO.NET	Proveedor de datos ADO.NET (https://www.firebirdsql.org/en/additional-downloads/)
	JDBC	Controlador Firebird JDBC (https://www.firebirdsql.org/en/jdbc-driver/)

Base de datos	Interfaz	Controladores
	ODBC	Controlador Firebird ODBC (https://www.firebirdsql.org/en/odbc-driver/)
IBM DB2	ADO	Proveedor IBM OLE DB para DB2
	ADO.NET	Proveedor IBM Data Server para .NET
	JDBC	Controlador IBM Data Server para JDBC y SQLJ
	ODBC	Controlador IBM DB2 ODBC
IBM DB2 for i	ADO	• Proveedor IBM DB2 para i5/OS IBMDA400 OLE DB • Proveedor IBM DB2 para i5/OS IBMDARLA OLE DB • Proveedor IBM DB2 para i5/OS IBMDASQL OLE DB
	ADO.NET	Proveedor de datos .NET Framework Data Provider para IBM i
	JDBC	Controlador IBM Toolbox para Java JDBC
	ODBC	Controlador iSeries Access ODBC
IBM Informix	ADO	Controlador IBM Informix OLE DB
	JDBC	Controlador IBM Informix JDBC
	ODBC	Controlador IBM Informix ODBC
Microsoft Access	ADO	• Proveedor de Microsoft Jet OLE DB • Proveedor de Microsoft Access Database Engine OLE DB
	ADO.NET	Proveedor de datos de .NET Framework para OLE DB
	ODBC	• Controlador de Microsoft Access
MariaDB	ADO.NET	Si falta el conector .NET especial para MariaDB, use **Connector/NET** pararMySQL (https://dev.mysql.com/downloads/connector/net/).
	JDBC	MariaDB Connector/JDBC (https://downloads.mariadb.org/)
	ODBC	MariaDB Connector/ODBC (https://downloads.mariadb.org/)
Microsoft SQL Server	ADO	• Controlador Microsoft OLE DB para SQL Server (MSOLEDBSQL) • Proveedor de Microsoft OLE DB para SQL Server (SQLOLEDB) • Cliente nativo de SQL Server (SQLNCLI)
	ADO.NET	• Proveedor de datos .NET Framework para SQL Server • Proveedor de datos .NET Framework para OLE DB
	JDBC	• Controlador JDBC de Microsoft para SQL Server (https://msdn.microsoft.com/library/mt484311.aspx)
	ODBC	• Cliente nativo de SQL Server
MySQL	ADO.NET	Conector/NET (https://dev.mysql.com/downloads/connector/net/)

© 2016-2022 Altova GmbH

Base de datos	Interfaz	Controladores
	JDBC	Conector/JDBC (https://dev.mysql.com/downloads/connector/j/)
	ODBC	Conector/ODBC (https://dev.mysql.com/downloads/connector/odbc/)
Oracle	ADO	• Proveedor de datos Oracle para OLE DB • Proveedor de Microsoft OLE DB para Oracle
	ADO.NET	Proveedor de datos Oracle para .NET (https://www.oracle.com/technetwork/topics/dotnet/index-085163.html)
	JDBC	• Controlador JDBC Thin • Controlador JDBC Oracle Call Interface (OCI) Estos controladores se suelen instalar durante la instalación del cliente de base de datos Oracle. Conéctese con el controlador OCI (no con Thin) si usa el componente Oracle XML DB.
	ODBC	• Microsoft ODBC para Oracle • Controlador Oracle ODBC (por lo general se instala durante la instalación del cliente de base de datos Oracle)
PostgreSQL	JDBC	Controlador PostgreSQL JDBC (https://jdbc.postgresql.org/download.html)
	ODBC	psqlODBC (https://odbc.postgresql.org/)
	Conexión nativa	Disponible. No necesita instalar ningún controladores si usa una conexión nativa.
Progress OpenEdge	JDBC	Conector JDBC (https://www.progress.com/jdbc/openedge)
	ODBC	Conector ODBC (https://www.progress.com/odbc/openedge)
SQLite	Conexión nativa	Disponible. No necesita instalar ningún controladores si usa una conexión nativa.
Sybase	ADO	Proveedor de Sybase ASE OLE DB
	JDBC	jConnect™ para JDBC
	ODBC	Controlador Sybase ASE ODBC
Teradata	ADO.NET	Proveedor de datos .NET para Teradata (https://downloads.teradata.com/download/connectivity/net-data-provider-for-teradata)
	JDBC	Controlador JDBC de Teradata (https://downloads.teradata.com/download/connectivity/jdbc-driver)
	ODBC	Controlador ODBC de Teradata para Windows (https://downloads.teradata.com/download/connectivity/odbc-driver/windows)

19.1.3 Conexiones ADO

Microsoft ActiveX Data Objects (ADO) es una tecnología de acceso de datos que permite conectarse a gran variedad de orígenes de datos con OLE DB. OLE DB es una interfaz alternativa a ODBC y JDBC. Ofrece acceso uniforme a los datos en un entorno COM (Component Object Model). ADO es el precursor del nuevo ADO.NET[964] y suele utilizarse para conectarse a bases de datos Microsoft nativas como Microsoft Access o SQL Server, aunque también puede usar ADO para otros orígenes de datos.

Es importante saber que puede escoger entre varios proveedores ADO y que en el caso de algunos es necesario que los descargue e instale en su equipo antes de poder usarlos. Por ejemplo, para conectarse a SQL Server puede usar estos proveedores ADO:

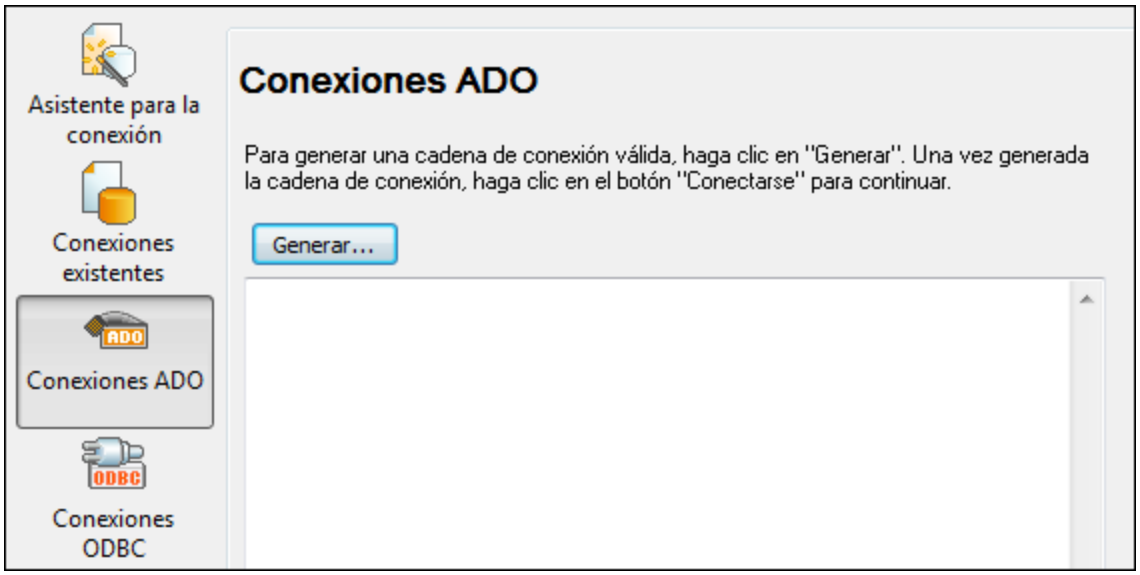
- *Controlador* de Microsoft OLE DB para SQL Server (MSOLEDBSQL)
- *Proveedor* de Microsoft OLE DB para SQL Server (SQLOLEDB)
- Cliente nativo de SQL Server (SQLNCLI)

De entre estos tres proveedores recomendamos que use MSOLEDBSQL, que puede descargar desde https://docs.microsoft.com/en-us/sql/connect/oledb/download-oledb-driver-for-sql-server?view=sql-server-ver15. Tenga en cuenta que la versión que descargue debe ser coincidir con la de la plataforma de XMLSpy (32 bits o 64 bits). Los proveedores SQLOLEDB y SQLNCLI se consideran obsoletos, por lo que no se recomiendan.

> Es un problema conocido que el proveedor de BD para SQL Server **Microsoft OLE DB Provider for SQL Server (SQLOLEDB)** tiene dificultades para enlazar parámetros de consultas complejas como las expresiones comunes de tabla (CTE) e instrucciones SELECT anidadas.

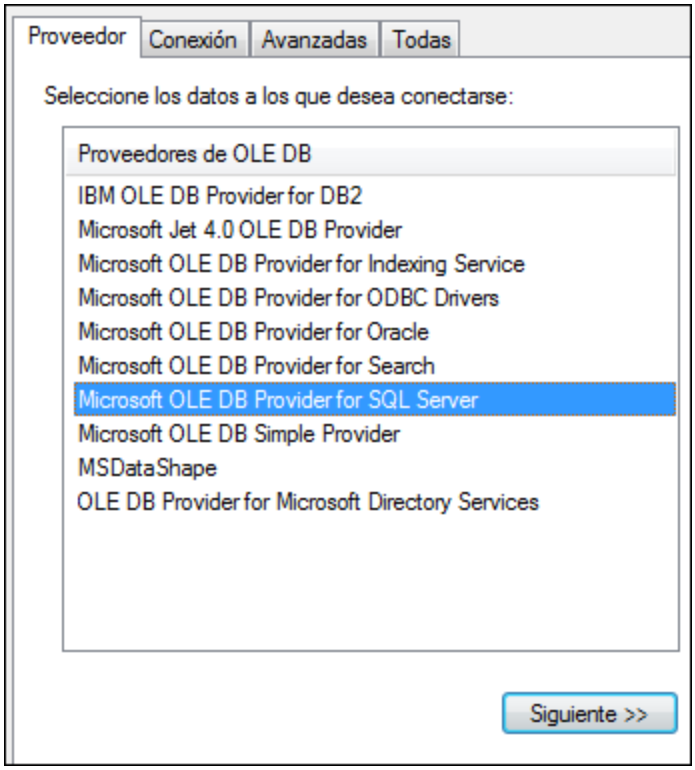
Para configurar una conexión ADO:

1. Inicie el asistente para la conexión de base de datos[953].
2. Haga clic en **Conexiones ADO**.



 © 2016-2022 Altova GmbH

3. Haga clic en **Generar**.



4. Seleccione el proveedor de datos que desea utilizar para establecer la conexión. La tabla que aparece a continuación enumera las combinaciones más comunes.

Para conectarse a esta BD...	Utilice este proveedor...
Microsoft Access	• **Proveedor de Microsoft Office Access Database Engine OLE DB** (recomendado) • **Proveedor de Microsoft Jet OLE DB** Si el proveedor **Microsoft Office Access Database Engine OLE DB Provider** no está disponible en la lista, asegúrese de que tiene instalados o bien Microsoft Access o el componente redistribuible del motor de base de datos de Microsoft Access (https://www.microsoft.com/en-us/download/details.aspx?id=54920) en su equipo.
SQL Server	• **Cliente nativo SQL Server** • **Proveedor para SQL Server de Microsoft OLE DB**
Otras bases de datos	Seleccione el proveedor correspondiente. Si no hay un proveedor OLE DB para su BD, instale el controlador necesario que ofrece el proveedor de la BD (consulte el Resumen de controladores de base de datos 955). Otra opción es configurar una conexión ADO.NET, ODBC o JDBC.

Para conectarse a esta BD...	Utilice este proveedor...
	Si el sistema operativo tiene un controlador ODBC para la BD, también puede usar **Microsoft OLE DB Provider for ODBC Drivers** o usar una conexión ODBC[971].

5. Haga clic en **Siguiente**.

Las instrucciones de las siguientes pantallas dependen del proveedor elegido. Para SQL Server deberá dar o seleccionar el nombre de host del servidor de BD, el método de autenticación y el nombre de usuario y la contraseña de la base de datos. Para ver un ejemplo, consulte Connecting to Microsoft SQL Server (ADO)[1007]. En el caso de Microsoft Access deberá buscar o introducir la ruta de acceso al archivo de BD. Para ver un ejemplo consulte Connecting to Microsoft Access (ADO)[1005].

La lista de propiedades de inicialización (parámetros de conexión) aparece en la pestaña *Todas* del cuadro de diálogo de conexión. Estas propiedades dependen del proveedor elegido. Consulte estos apartados para aprender a configurar las propiedades básicas de inicialización para bases de datos Microsoft Access y SQL Server:

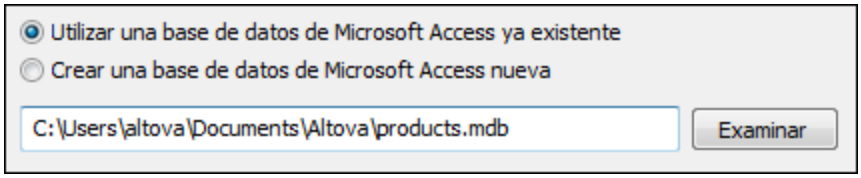
- Configurar las propiedades de vínculo de datos de SQL Server[961]
- Configurar las propiedades de vínculo de datos de Microsoft Access[962]

19.1.3.1 Conectarse a una base de datos Microsoft Access

Siga estas instrucciones si desea conectarse a una base de datos Microsoft Access que no requiere contraseña. Si la base de datos está protegida con contraseña, entonces consulte el apartado Conectarse a Microsoft Access (ADO)[1005].

Para conectarse a una BD Microsoft Access:

1. Inicie el asistente para la conexión de base de datos[953].
2. Seleccione el botón de opción *Microsoft Access (ADO)* y haga clic en **Siguiente**.



3. Seleccione el botón de opción *Utilizar una base de datos de Microsoft Access ya existente*.
4. Busque el archivo de BD o introduzca su ruta de acceso (relativa o absoluta).
5. Haga clic en **Conectarse**.

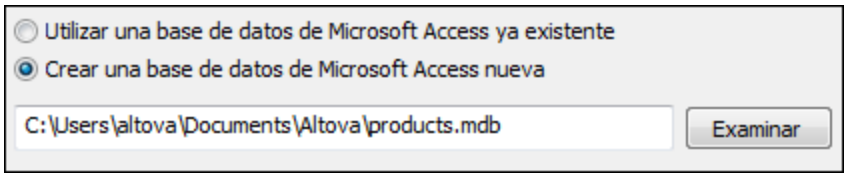
© 2016-2022 Altova GmbH

19.1.3.2 Crear una base de datos Microsoft Access

No sólo puede conectarse a un archivo de BD, sino que además puede crear un archivo de BD Microsoft Access nuevo (.accdb, .mdb) y conectarse a él aunque no tenga instalado Microsoft Access en el equipo. El archivo de BD que crea XMLSpy está vacío. Para crear la estructura de BD debe utilizar Microsoft Access o una herramienta como Altova DatabaseSpy (https://www.altova.com/es/databasespy.html).

Para crear una base de datos Microsoft Access nueva:

1. Inicie el asistente para la conexión de base de datos 953.
2. Seleccione el botón de opción *Microsoft Access (ADO)* y haga clic en **Siguiente**.



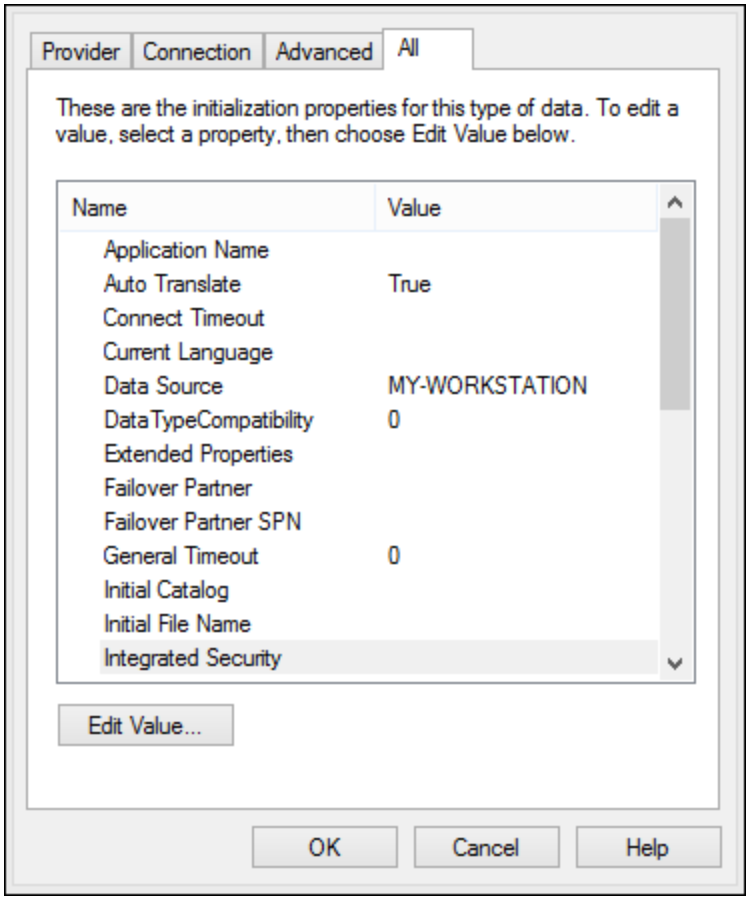
3. Seleccione el botón de opción *Crear una base de datos de Microsoft Access nueva* y escriba la ruta de acceso (relativa o absoluta) del archivo de BD que se debe crear. Si lo prefiere, haga clic en **Examinar** para seleccionar una carpeta, escriba el nombre del archivo en *Nombre del archivo* y haga clic en **Guardar**.

> Notas
>
> - Asegúrese de que tiene permisos de escritura para la carpeta donde desea crear el archivo de BD.
> - El nombre del archivo de BD debe tener la extensión **.mdb** o **.accdb**.

4. Haga clic en **Conectarse**.

19.1.3.3 Configurar las propiedades de vínculo de datos de SQL Server

Cuando se conecte a una base de datos Microsoft SQL Server por ADO (consulte Conexiones ADO 958), asegúrese de configurar correctamente estas propiedades de vínculo de datos en la pestaña *Todas* del cuadro de diálogo "Propiedades de vínculo de datos".



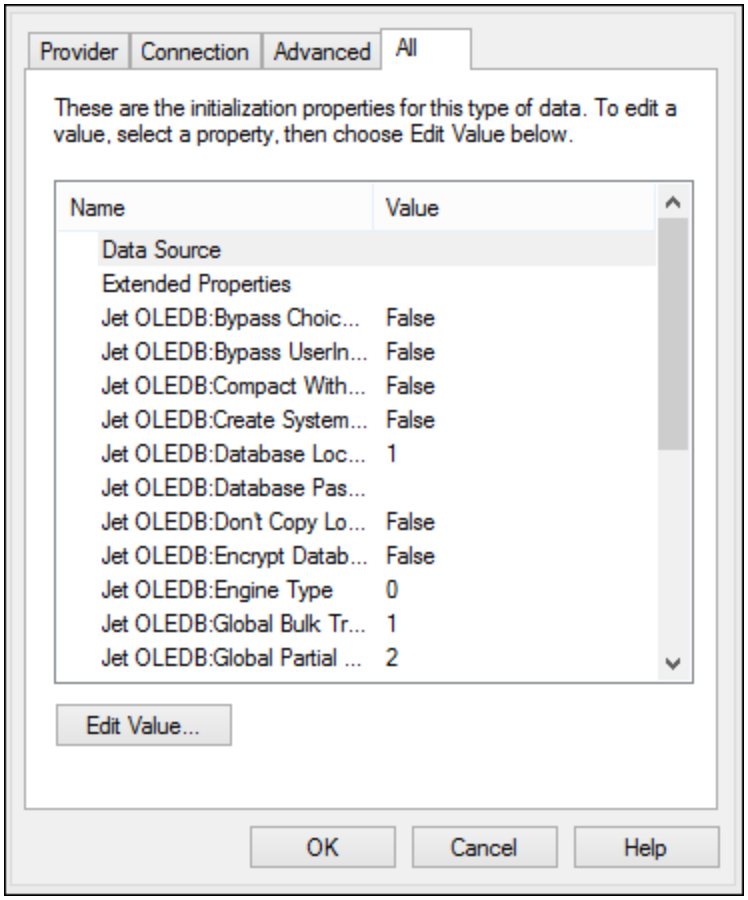
Data Link Properties dialog box

Propiedad	Notas
Seguridad integrada	Si seleccionó el proveedor de datos **SQL Server Native Client** en la pestaña *Proveedor*, esta propiedad debe ser un carácter de espacio en blanco.
Almacenar información de seguridad	Esta propiedad debe tener el valor **True**.

19.1.3.4 Configurar las propiedades de vínculo de datos de Microsoft Access

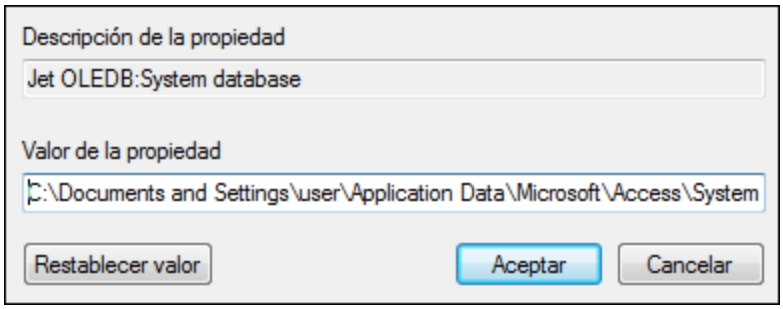
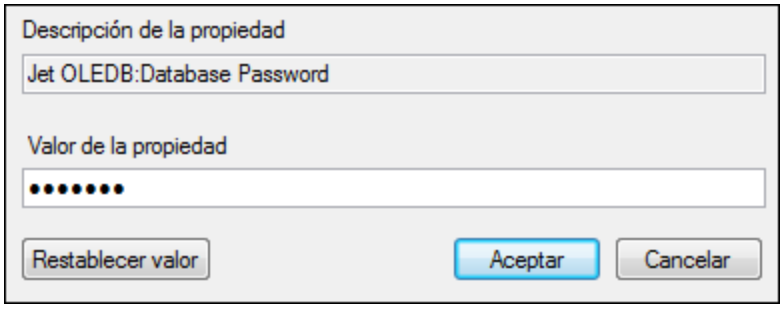
Cuando se conecte a una base de datos Microsoft Access por ADO (consulte Conexiones ADO 958), puede que deba configurar estas propiedades de vínculo de datos en la pestaña *Todas* del cuadro de diálogo "Propiedades de vínculo de datos".

 © 2016-2022 Altova GmbH



Data Link Properties dialog box

Propiedad	Notas
Origen de datos	Esta propiedad almacena la ruta de acceso del archivo de BD Microsoft Access. Para evitar problemas de conexión, recomendamos usar el formato de ruta de acceso UNC. Por ejemplo: `\\servidor\compartir$\rutaArchivo`
Base de datos OLEDB:System Database	Esta propiedad almacena la ruta de acceso del archivo de información de grupo de trabajo. Quizás sea necesario configurar el valor de esta propiedad para poder establecer la conexión con la BD Microsoft Access. Si se produce un error relacionado con el archivo de información de grupo de trabajo, busque el archivo de información de grupo de trabajo (**System.MDW**) que corresponda a su perfil de usuario e introduzca la ruta de acceso del archivo **System.MDW** como valor de esta propiedad.

Propiedad	Notas
	
Contraseña Jet OLEDB:Database	Si la base de datos está protegida con contraseña, el valor de esta propiedad debe ser la contraseña de la base de datos. 

19.1.4 Conexiones ADO.NET

ADO.NET es un conjunto de bibliotecas de Microsoft .NET Framework diseñado para interactuar con datos, incluidos datos de bases de datos. Para conectarse a una base de datos desde XMLSpy por ADO.NET es necesario tener instalado Microsoft .NET Framework 4 o superior. Como puede ver más abajo, la conexión a la base de datos a través de ADO.NET se hace seleccionando un proveedor .NET y aportando una cadena de conexión.

Un proveedor de datos .NET es una colección de clases que permite conectarse a un tipo concreto de origen de datos (p. ej. un servidor SQL Server o una base de datos Oracle), ejecutar comandos en él y recuperar sus datos. En otras palabras, con ADO .NET las aplicaciones como XMLSpy interactúan con una base de datos a través de un proveedor de datos. Cada proveedor de datos está optimizado para poder trabajar con el tipo concreto de origen de datos para el que está diseñado. Hay dos tipos de proveedores .NET:

1. El proveedor que viene por defecto con Microsoft .NET Framework.
2. Proveedores que ofrecen los principales proveedores de bases de datos como extensión para .NET Framework. Este tipo de proveedores ADO.NET deben instalarse por separado y por lo general se pueden descargar del sitio web del correspondiente proveedor de base de datos.

Nota: ciertos proveedores ADO.NET son incompatibles o tienen una compatibilidad limitada. Consulte el apartado Notas sobre compatibilidad con ADO.NET 971.

 © 2016-2022 Altova GmbH

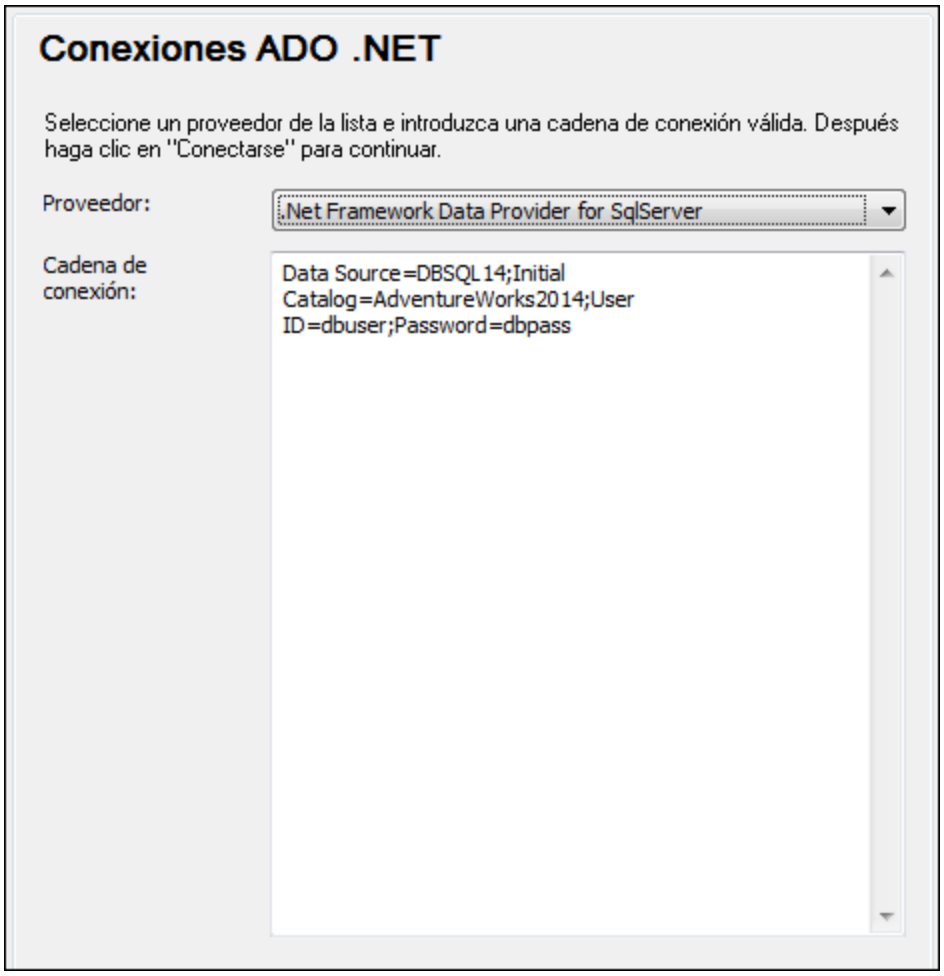
Para configurar una conexión ADO.NET:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en **Conexiones ADO.NET**.
3. Seleccione un proveedor de datos .NET de la lista.

> En la lista *Proveedores* aparecen todos los proveedores que vienen por defecto con .NET Framework. Los proveedores de datos .NET propios del proveedor de la base de datos solo aparecen en la lista si ya están instalados en el sistema. Deben instalarse en el caché global de ensamblados (GAC) con ayuda del archivo .msi o .exe que ofrece el proveedor de la base de datos.

4. Introduzca la cadena de conexión con la base de datos. Una cadena de conexión defina la información de conexión con la base de datos y está formada por pares clave/valor de parámetros de conexión delimitados por caracteres de punto y coma. Por ejemplo, la cadena de conexión `Data Source=DBSQLSERV;Initial Catalog=ProductsDB;User ID=dbuser;Password=dbpass` permite conectarse a la base de datos SQL Server `ProductsDB` del servidor `DBSQLSERV`, con el nombre de usuario `dbuser` y la contraseña `dbpass`. Puede crear la cadena de conexión tecleando los pares clave/valor directamente en el cuadro de texto *Cadena de conexión*, pero también puede crearla con Visual Studio (*véase* *Crear una cadena de conexión en Visual Studio* 966).

> La sintaxis de la cadena de conexión depende del proveedor que se seleccione en la lista *Proveedores*. Para ver un ejemplo consulte el apartado Ejemplo: cadenas de conexión ADO.NET 969.



5. Haga clic en **Conectarse** para terminar.

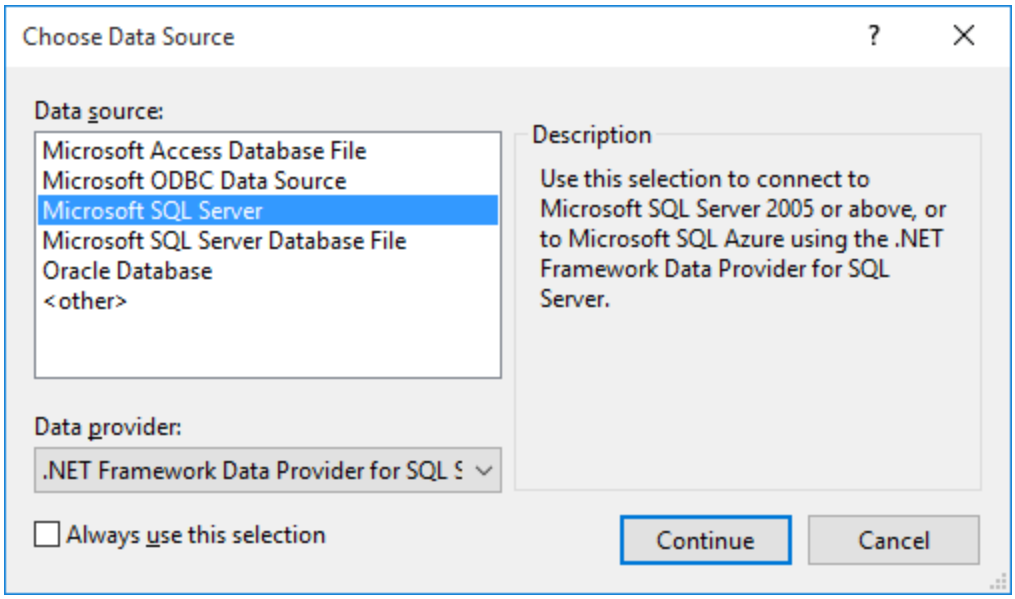
19.1.4.1 Crear una cadena de conexión en Visual Studio

Para conectarse a un origen de datos usando ADO.NET se necesita una cadena de conexión válida. A continuación explicamos cómo crear una cadena de conexión desde Visual Studio.

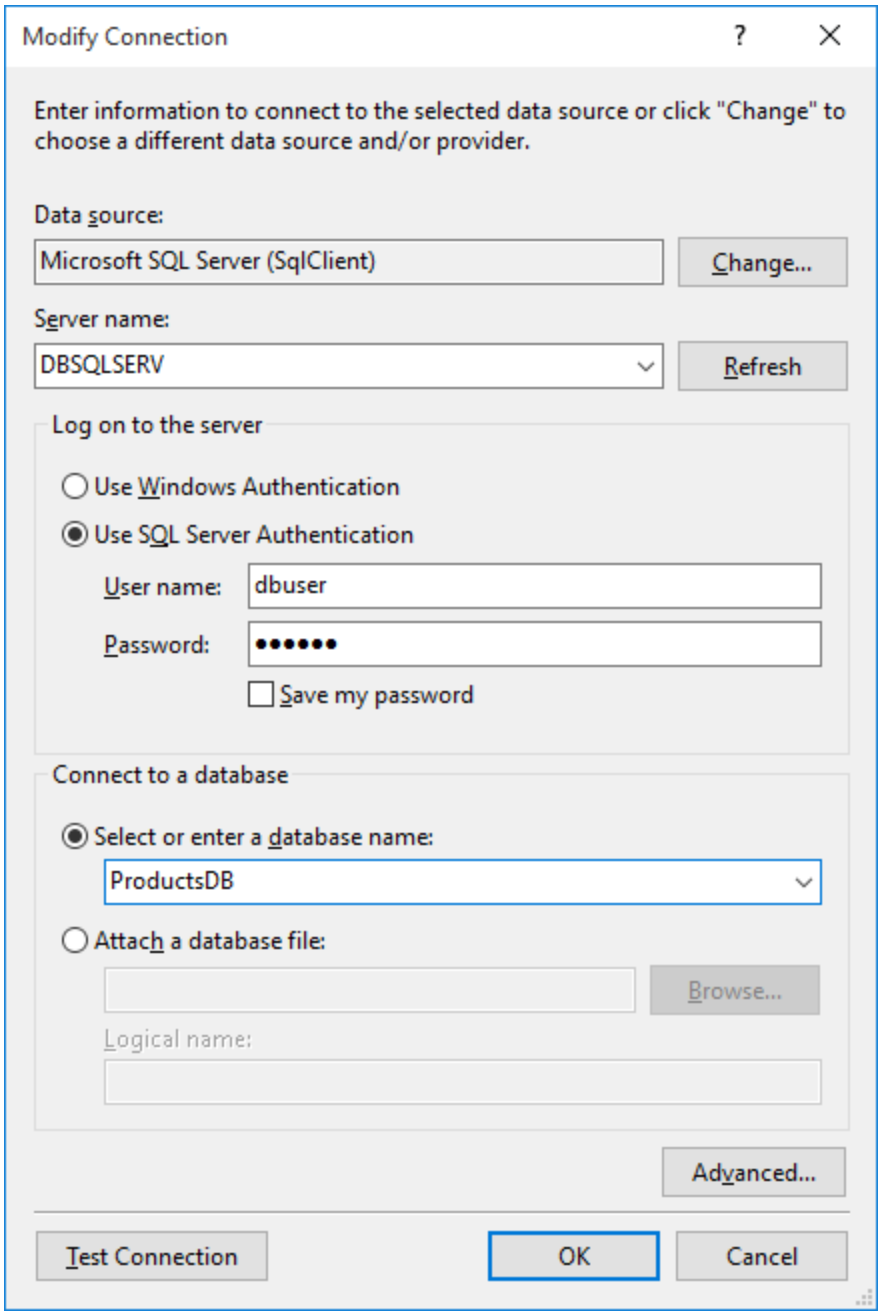
Para crear una cadena de conexión en Visual Studio:

1. En el menú **Herramientas** haga clic en **Conectar a base de datos**.
2. Seleccione un origen de datos de la lista (en este ejemplo Microsoft SQL Server). El proveedor de datos se rellena automáticamente en función de la opción elegida.

© 2016-2022 Altova GmbH



3. Haga clic en **Continuar**.



4. Introduzca el nombre de host del servidor, el nombre de usuario y la contraseña de la base de datos. En este ejemplo nos conectamos a la base de datos `ProductsDB` en el servidor `DBSQLSERV`, usando SQL Server para la autenticación.
5. Para terminar haga clic en **Aceptar**.

Si la conexión se establece correctamente, aparecerá en la ventana Explorador de servidores. Para abrir esta ventana puede usar el comando **Vista | Explorador de servidores**. Para obtener la cadena de conexión con la base de datos, haga clic con el botón derecho en la conexión en la ventana Explorador de servidores y seleccione el comando **Propiedades**. Ahora aparece la cadena de conexión en la ventana Propiedades de

© 2016-2022 Altova GmbH

Visual Studio. Recuerde que debe reemplazar los asteriscos con la contraseña antes de pegar la cadena en el cuadro de texto *Cadena de conexión* de XMLSpy.

19.1.4.2 Ejemplo: cadenas de conexión ADO.NET

Para configurar una conexión ADO.NET deberá seleccionar un proveedor ADO.NET en el cuadro de diálogo de conexión a la base de datos y deberá introducir una cadena de conexión (*véase* *Conexiones ADO.NET* 964). A continuación ofrecemos ejemplos de cadenas de conexión ADO.NET para varias bases de datos.

Proveedor de datos .NET para Teradata

Este proveedor se puede descargar del sitio web de Teradata (https://downloads.teradata.com/download/connectivity/net-data-provider-for-teradata). Un ejemplo de cadena de conexión sería:

```
Data Source=DirecciónServidor;User Id=usuario;Password=contraseña;
```

Proveedor de datos .NET Framework Data Provider para IBM i

Este proveedor se instala con *IBM i Access Client Solutions - Windows Application Package*. Un ejemplo de cadena de conexión sería:

```
DataSource=DirecciónServidor;UserID=usuario;Password=contraseña;DataCompression=True;
```

Para más información consulte el archivo de ayuda llamado ".NET Provider Technical Reference" que viene con el paquete de instalación.

Proveedor de datos .NET Framework Data Provider para MySQL

Este proveedor se puede descargar del sitio web de MySQL (https://dev.mysql.com/downloads/connector/net/). Un ejemplo de cadena de conexión sería:

```
Server=127.0.0.1;Uid=root;Pwd=12345;Database=test;
```

Véase también https://dev.mysql.com/doc/connector-net/en/connector-net-programming-connecting-connection-string.html

Proveedor de datos .NET Framework Data Provider para SQL Server

Un ejemplo de cadena de conexión sería:

```
Data Source=DBSQLSERV;Initial Catalog=ProductsDB;User ID=dbuser;Password=dbpass
```

Véase también https://msdn.microsoft.com/en-us/library/ms254500(v=vs.110).aspx

Proveedor de datos IBM DB2 Data Provider 10.1.2 para .NET Framework 4.0

```
Database=PRODUCTS;UID=usuario;Password=contraseña;Server=localhost:50000;
```

Nota: por lo general este proveedor se instala con el paquete de IBM DB2 Data Server Client. Si después de instalar el paquete de IBM DB2 Data Server Client el proveedor no aparece en la lista de proveedores ADO.NET, consulte esta nota técnica: https://www-01.ibm.com/support/docview.wss?uid=swg21429586.

Véase también
https://www.ibm.com/support/knowledgecenter/en/SSEPGG_10.1.0/com.ibm.swg.im.dbclient.adonet.ref.doc/doc/DB2ConnectionClassConnectionStringProperty.html

Proveedor de datos Oracle para .NET (ODP.NET)

El paquete de instalación que incluye el proveedor ODP.NET se puede descargar del sitio web de Oracle (http://www.oracle.com/technetwork/topics/dotnet/downloads/index.html). Un ejemplo de cadena de conexión sería:

```
Data Source=DSORCL;User Id=usuario;Password=contraseña;
```

En esta cadena de conexión, `DSORCL` es el nombre del origen de datos que apunta a un nombre de servicio Oracle que está definido en el archivo **tnsnames.ora** (tal y como se describe en el apartado Conectarse a Oracle (ODBC) 1020).

Si prefiere establecer la conexión sin configurar un nombre de servicio en el archivo **tnsnames.ora**, entonces use una cadena de conexión parecida a esta:

```
Data Source=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=host)(PORT=puerto)))
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=MyOracleSID)));User
Id=usuario;Password=contraseña;
```

Véase también https://docs.oracle.com/cd/B28359_01/win.111/b28375/featConnecting.htm

© 2016-2022 Altova GmbH

19.1.4.3 Notas sobre compatibilidad con ADO.NET

En esta tabla aparecen controladores de base de datos ADO.NET conocidos que por ahora no son compatibles con XMLSpy o que tienen una compatibilidad limitada.

Base de datos	Controlador	Notas
Todas las bases de datos	**Proveedor de datos .Net Framework para ODBC**	Compatibilidad limitada. Existen problemas conocidos con las conexiones Microsoft Access. Es mejor utilizar conexiones directas ODBC.
	Proveedor de datos .Net Framework para OleDb	Compatibilidad limitada. Existen problemas conocidos con las conexiones Microsoft Access. Es mejor utilizar conexiones directas ADO.
Firebird	**Proveedor de datos Firebird ADO.NET**	Compatibilidad limitada. Es mejor utilizar ODBC o JDBC.
Informix	**Proveedor de datos IBM Informix para .NET Framework 4.0**	Incompatible. Utilice el proveedor **DB2 Data Server Provider**.
IBM DB2 for i (iSeries)	**Proveedor de datos .Net Framework para i5/OS**	Incompatible. utilice el **proveedor de datos .Net Framework para IBM i**, que se instala con el paquete *IBM i Access Client Solutions - Windows Application Package*.
Oracle	**Proveedor de datos .Net Framework para Oracle**	Compatibilidad limitada. Aunque este controlador viene con .NET Framework, Microsoft no recomienda su uso porque es obsoleto.
PostgreSQL	-	Para este proveedor no hay controladores ADO.NET compatibles. Es mejor usar una conexión nativa.
Sybase	-	Para este proveedor no hay controladores ADO.NET compatibles.

19.1.5 Conexiones ODBC

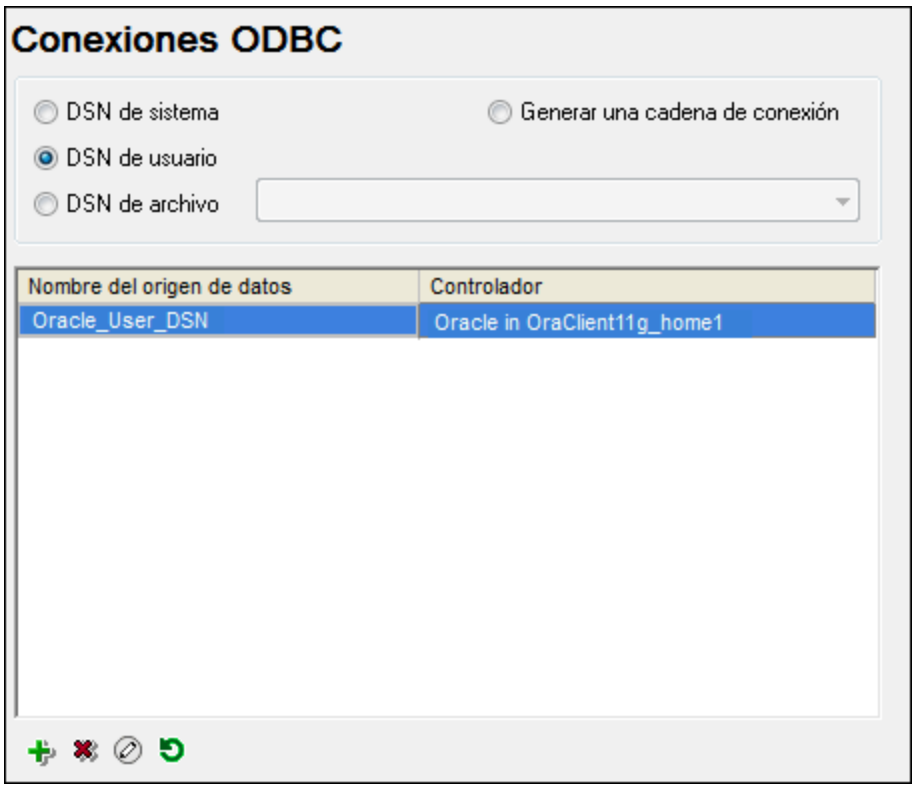
ODBC (Open Database Connectivity) es una tecnología de acceso de datos muy popular con la que se puede conectar a bases de datos desde XMLSpy. Puede utilizarse como método principal de conexión con bases de datos o como alternativa a las conexiones OLE DB o JDBC.

Para conectarse a una base de datos por ODBC primero es necesario contar con un DSN de ODBC en el sistema operativo. El DSN describe de manera uniforme la conexión de BD a todas las aplicaciones cliente compatibles con ODBC que estén en el sistema, incluido XMLSpy. Los DSN pueden ser de varios tipos:

- DSN de sistema
- DSN de usuario
- DSN de archivo

A un origen de datos de sistema pueden acceder todos los usuarios que tengan privilegios en el sistema operativo. A un origen de datos de usuario solo puede acceder el usuario que lo creó. Y, por último, si crea un DSN de archivo, el origen de datos se creará como archivo con extensión `.dsn` que podrá compartir con otros usuarios (siempre que tengan instalados los controladores que utiliza el origen de datos).

Los DSN que estén en el equipo aparecen en el cuadro de diálogo de conexión de base de datos al hacer clic en **Conexiones ODBC**.



Cuadro de diálogo Conexiones ODBC

Si el DSN de la base de datos no existe, el asistente para la conexión de base de datos de XMLSpy le ayudará a crearlo. Si lo prefiere puede crearlo en Windows directamente. En ambos casos, antes de continuar, compruebe que el controlador ODBC para la base de datos está en la lista de controladores disponibles del sistema operativo (consulte el apartado Ver los controladores disponibles 973).

Para conectarse usando un DSN nuevo:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en el botón **Conexiones ODBC**.
3. Seleccione un tipo de origen de datos (DSN de usuario, de sistema o de archivo).

 © 2016-2022 Altova GmbH

> **Nota:** para crear un DSN de sistema necesitará derechos de administrador en el sistema y UModel debe ejecutarse como administrador.

4. Haga clic en **Agregar** .
5. Seleccione un controlador y haga clic en **DSN de usuario** o **DSN de sistema**. Si el controlador correspondiente a la BD no aparece en la lista, descárguelo e instálelo.
6. En el cuadro de diálogo que aparece debe rellenar la información de conexión para terminar de configurar la conexión.

Para que la conexión se establezca correctamente deberá dar el nombre de host (o dirección IP) del servidor de la BD, así como el nombre de usuario y la contraseña. Quizás sean necesarios otros parámetros de conexión. Para más información consulte la documentación que ofrece el proveedor de la base de datos. Una vez creado, el DSN estará en la lista de nombres de orígenes de datos. Así podrá volver a usar los datos de la conexión cada vez que desee conectarse a la BD.

Para conectarse por medio de un DSN:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en el botón **Conexiones ODBC**.
3. Elija el tipo de origen de datos (DSN de usuario, de sistema o de archivo).
4. Haga clic en el DSN y después en **Conectarse**.

Para generar una cadena de conexión basada en un archivo .dsn:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en el botón **Conexiones ODBC**.
3. Seleccione *Generar una cadena de conexión* y después haga clic en **Generar**.
4. Si quiere generar la conexión a partir de un DSN de archivo, haga clic en la pestaña *Origen de datos de archivo.* Si no es así, haga clic en la pestaña *Origen de datos de equipo* (los DSN de sistema y de usuario se denominan orígenes de datos de equipo)
5. Seleccione el archivo `.dsn` correspondiente y haga clic en **Aceptar**.

Para conectarse a través de una cadena de conexión preparada previamente:

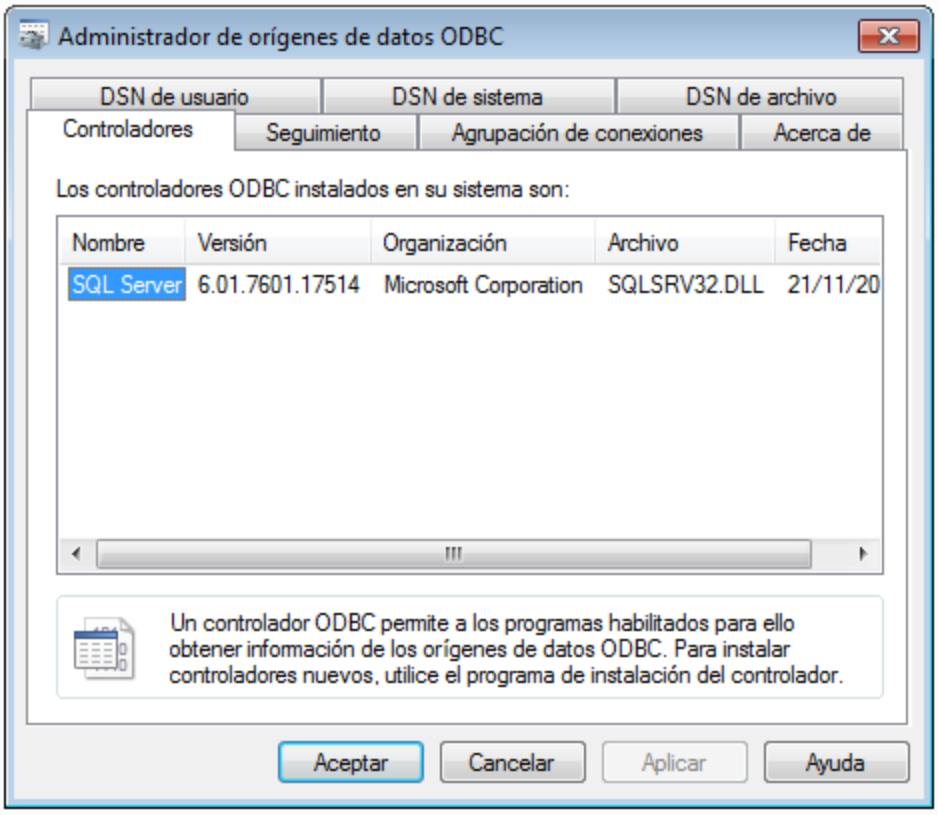
1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en el botón **Conexiones ODBC**.
3. Seleccione *Generar una cadena de conexión*.
4. Pegue la cadena de conexión en el cuadro de texto y haga clic en **Conectarse**.

19.1.5.1 Controladores ODBC disponibles

En el administrador de orígenes de datos ODBC puede ver qué controladores ODBC están disponibles en su sistema operativo. El administrador (**Odbcad32.exe**) se puede abrir desde el panel de control de Windows (desde **Herramientas administrativas**). En sistemas operativos de 64 bits encontrará dos versiones de este ejecutable:

- La versión de 32 bits está en el directorio **C:\Windows\SysWoW64** (siempre y cuando **C:** sea su unidad de sistema).
- La versión de 64 bits está en el directorio **C:\Windows\System32**.

Los controladores de BD de 43 bits aparecerán en la versión de 32 bits del administrador de orígenes de datos ODBC, mientras que los controladores de 64 bits aparecerán en la versión de 64 bits. Esto debe tenerse en cuenta a la hora de consultar los controladores en el administrador.



Si el controlador de la base de datos de destino no está en la lista del administrador o si desea agregar algún controlador, deberá descargar el controlador (Resumen de controladores de base de datos 955). Cuando el controlador esté disponible en el sistema podrá crear conexiones ODBC con él.

19.1.6 Conexiones JDBC

JDBC (Java Database Connectivity) es una interfaz de acceso a base de datos que forma parte de la plataforma de software Java de Oracle. Las conexiones JDBC suelen consumir más recursos que las conexiones ODBC pero pueden ofrecer más características.

Requisitos

- Tener instalado Java Runtime Environment (JRE) o Java Development Kit (JDK). Este último puede ser el JDK de Oracle o uno de código abierto, como Oracle OpenJDK. XMLSpy identifica la ruta de acceso

© 2016-2022 Altova GmbH

al equipo virtual Java (JVM) a partir de estas ubicaciones, en este orden: a) la ruta personal de acceso al JVM que puede indicar en en las **Opciones** ; b) la ruta de acceso al JVM que se encuentra en el registro; c) la variable de entorno `JAVA_HOME`.

- Asegúrese de que la plataforma de XMLSpy (32 o 64 bits) es la misma que la de JRE/JDK.
- Tener instalados los controladores JDBC del proveedor de la BD. Estos pueden ser los controladores JDBC que forman parte de la instalación de un cliente de BD o bibliotecas JDBC (archivos .jar) que haya descargado por separado, siempre que estos estén disponible y la BD sea compatible (*véase también* *Ejemplos de conexión a bases de datos* 985).
- La variable del entorno `CLASSPATH` debe incluir la ruta de acceso del controlador JDBC (que puede ser un archivo .jar o varios). Algunos clientes de base de datos configuran esta variable automáticamente durante la instalación. Para más información consulte el apartado Configurar la variable CLASSPATH 977.

Conectarse a un servidor SQL a través de JDBC con credenciales de Windows

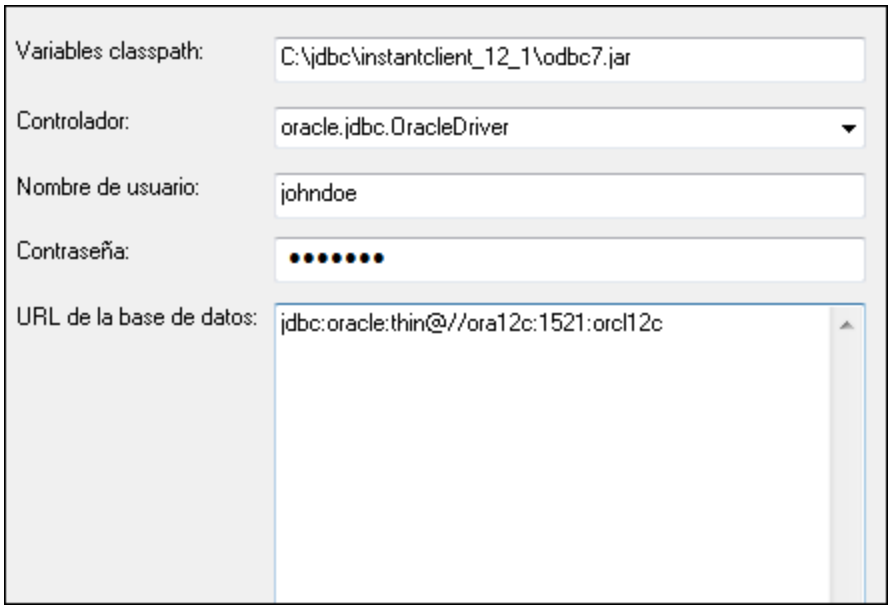
Si se conecta a un servidor SQL mediante JDBC con credenciales de Windows (seguridad integrada), tenga en cuenta lo siguiente:

- Debe copiar el archivo **sqljdbc_auth.dll** que viene con el paquete del controlador JDBC a un directorio que esté en la variable de entorno PATH del sistema. Existen dos archivos de este tipo, uno para la plataforma x86 y otro para la plataforma x64. Debe asegurarse de que añade a la ruta PATH el que corresponde con su plataforma de JDK.
- La cadena de conexión JDBC debe incluir la propiedad **`integratedSecurity=true`**.

Para más información consulte la documentación Controlador JDBC de Microsoft para SQL Server: https://docs.microsoft.com/es-es/sql/connect/jdbc/building-the-connection-url.

Para configurar una conexión JDBC:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en el botón **Conexiones JDBC**.
3. Si quiere, introduzca una lista de rutas de archivo .jar separadas por caracteres de punto y coma en el cuadro de texto *Variables classpath*. Las bibliotecas .jar que se introduzcan aquí se cargarán en el entorno junto a las que ya estén definidas en la variable de entorno `CLASSPATH`. Cuando termine de editar el cuadro de texto Variables classpath, todos los controladores JDBC encontrados en las bibliotecas .jar de origen se cargan automáticamente a la lista del cuadro combinado *Controlador*.



4. En el cuadro combinado *Controlador* seleccione un controlador JDBC de la lista o introduzca un nombre de clase Java. Observe que esta lista contiene todos los controladores JDBC configurados a través de la variable de entorno `CLASSPATH` (*véase* *Configurar la variable CLASSPATH* 977), así como los controladores encontrados en el campo *Variables classpath*.

> Las rutas de acceso del controlador JDBC definidas sen la variable CLASSPATH, así como las rutas de acceso de los archivos .jar introducidos directamente en el cuadro de diálogo de conexión a la base de datos se envían a Java Virtual Machine (JVM). JVM decide qué controladores se utilizan para establecer la conexión. Se recomienda realizar un seguimiento de las clases Java que se cargan en JVM para evitar conflictos y resultados inesperados a la hora de conectarse a la base de datos.

5. Introduzca el nombre de usuario y la contraseña de la BD.
6. En el cuadro de texto *URL de la base de datos:* introduzca la URL de la conexión JDBC en el formato propio del tipo de base de datos utilizado. En la siguiente tabla puede ver la sintaxis de las URL de conexión JDBC para los tipos de base de datos más frecuentes.

Base de datos	URL de conexión JDBC
Firebird	`jdbc:firebirdsql://<host>[:<puerto>]/<ruta de acceso o alias de la BD>`
IBM DB2	`jdbc:db2://nombreHost:puerto/nombreBaseDatos`
IBM DB2 for i	`jdbc:as400://[host]`
IBM Informix	`jdbc:informix-sqli://nombreHost:puerto/nombreBaseDatos:INFORMIXSERVER=miservidor`
MariaDB	`jdbc:mariadb://nombreHost:puerto/nombreBaseDatos`

© 2016-2022 Altova GmbH

Base de datos	URL de conexión JDBC
Microsoft SQL Server	jdbc:sqlserver://**nombreHost**:**puerto**;**nombreBaseDatos**=name
MySQL	jdbc:mysql://**nombreHost**:**puerto**/**nombreBaseDatos**
Oracle	jdbc:oracle:thin:@**nombreHost**:**puerto**:**SID** jdbc:oracle:thin:@//**nombreHost**:**puerto**:**servicio**
Oracle XML DB	jdbc:oracle:oci:@//**nombreHost**:**puerto**:**servicio**
PostgreSQL	jdbc:postgresql://**nombreHost**:**puerto**/**nombreBaseDatos**
Progress OpenEdge	jdbc:datadirect:openedge://**host**:**puerto**;databaseName=**nombre_bd**
Sybase	jdbc:sybase:Tds:**nombreHost**:**puerto**/**nombreBaseDatos**
Teradata	jdbc:teradata://**nombreServidorBaseDatos**

Nota: en algunos casos puede modificarse el formato sintáctico (p. ej. la URL de la base de datos puede excluir el puerto o puede incluir el nombre de usuario y la contraseña). Consulte la documentación del proveedor de BD para obtener más información.

7. Haga clic en **Conectarse**.

19.1.6.1 Configurar la variable CLASSPATH

La variable de entorno `CLASSPATH` es utilizada por Java Runtime Environment (JRE) o por Java Development Kit (JDK) para encontrar las clases Java y otros archivos de recursos del sistema operativo. Cuando se conecte a una BD con JDBC, esta variable debe incluir la ruta de acceso del controlador JDBC del sistema y, en algunos casos, la ruta de acceso de otros archivos de biblioteca relacionados con el tipo de BD que esté utilizando.

En la tabla que aparece a continuación puede ver ejemplos de variables `CLASSPATH`. Lo más importante que debe tener en cuenta es que quizás deba adaptar esta información dependiendo de la ubicación y del nombre del controlador JDBC y de la versión JRE/JDK que esté en su sistema operativo. Para evitar problemas de conexión recomendamos leer detenidamente las instrucciones de instalación del controlador JDBC que esté instalado en el sistema.

Base de datos	Ejemplo de variable CLASSPATH
Firebird	C:\Archivos de programa\Firebird\Jaybird-2.2.8-JDK_1.8\jaybird-full-2.2.8.jar
IBM DB2	C:\Archivos de programa (x86)\IBM\SQLLIB\java\db2jcc.jar;C:\Archivos de programa (x86)\IBM\SQLLIB\java\db2jcc_license_cu.jar;
IBM DB2 para i	C:\jt400\jt400.jar;

Base de datos	Ejemplo de variable CLASSPATH
IBM Informix	`C:\Informix_JDBC_Driver\lib\ifxjdbc.jar;`
MariaDB	**<directorio de instalación>**\mariadb-java-client-2.2.0.jar
Microsoft SQL Server	`C:\Archivos de programa\Microsoft JDBC Driver 4.0 for SQL Server\sqljdbc_4.0\enu\sqljdbc.jar`
MySQL	**<directorio de instalación>**\mysql-connector-java-*version*-bin.jar;
Oracle	**ORACLE_HOME**\jdbc\lib\ojdbc6.jar;
Oracle (with XML DB)	**ORACLE_HOME**\jdbc\lib\ojdbc6.jar;**ORACLE_HOME**\LIB\xmlparserv2.jar; **ORACLE_HOME**\RDBMS\jlib\xdb.jar;
PostgreSQL	**<directorio de instalación>**\postgresql.jar
Progress OpenEdge	`%DLC%\java\openedge.jar;%DLC%\java\pool.jar;` **Nota:** siempre y cuando Progress OpenEdge SDK esté instalado en el equipo, `%DLC%` es el directorio donde está instalado OpenEdge.
Sybase	`C:\sybase\jConnect-7_0\classes\jconn4.jar`
Teradata	**<directorio de instalación>**\tdgssconfig.jar;**<directorio de instalación>**\terajdbc4.jar

- Los cambios en la configuración de la variable CLASSPATH pueden afectar al comportamiento de las aplicaciones Java del equipo. Consulte la documentación de Java antes de continuar.
- Las variables de entorno pueden ser del sistema o del usuario. Para cambiar las variables de entorno del sistema es necesario tener derechos de administrador.
- Tras modificar la variable de entorno, reinicie los programas que estén en ejecución para que los cambios surtan efecto. También puede cerrar sesión o reiniciar el sistema.

Para configurar CLASSPATH en Windows 7:

1. Abra el menú **Inicio** y haga clic con el botón derecho en **Equipo**.
2. Haga clic en **Propiedades**.
3. Haga clic en **Configuración avanzada del sistema**.
4. En la pestaña *Avanzadas*, haga clic en **Variables de entorno**.
5. Busque la variable `CLASSPATH` del sistema o del usuario y haga clic en **Editar**. Si no existe, haga clic en **Nueva** para crearla.

© 2016-2022 Altova GmbH

6. Edite el valor de la variable e incluya la ruta de acceso del controlador JDBC. Utilice un punto y coma para separar la ruta de acceso del controlador de las demás partes de la variable.

Para configurar CLASSPATH en Windows 10:

1. Pulse la tecla Windows y teclee "variables de entorno".
2. Haga clic en la sugerencia **Editar las variables de entorno del sistema**.
3. Haga clic en **Variables de entorno**.
4. Busque la variable `CLASSPATH` del sistema o del usuario y haga clic en **Editar**. Si no existe, haga clic en **Nueva** para crearla.
5. Edite el valor de la variable e incluya la ruta de acceso del controlador JDBC. Utilice un punto y coma para separar la ruta de acceso del controlador de las demás partes de la variable.

19.1.7 Conexiones PostgreSQL

Las conexiones a bases de datos PostgreSQL pueden configurarse como conexiones nativas o a través de ODBC, JDBC y otros controladores. La ventaja de configurar una conexión nativa es que no es necesario tener controladores instalados en el sistema.

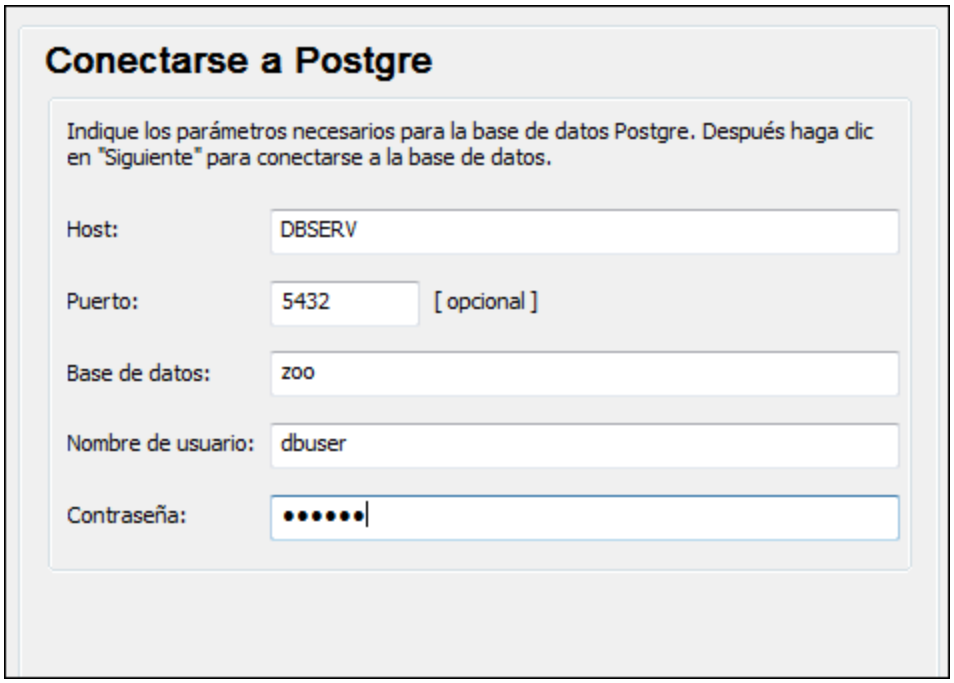
Si prefiere establecer la conexión por medio de un controlador no nativo, consulte estos temas de la documentación:

- Configurar una conexión JDBC 974
- Conectarse a PostgreSQL (ODBC) 1026

Si lo que quiere es configurar una conexión nativa con PostgreSQL, siga las instrucciones que aparecen más abajo. El único requisito es conocer el nombre de host, el puerto, el nombre de la base de datos, el nombre de usuario y la contraseña.

Para configurar una conexión PostgreSQL nativa:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en el botón **Conexiones PostgreSQL**.
3. Ahora introduzca el host (localhost si PostgreSQL se ejecuta en el mismo equipo), el puerto (es opcional, pero suele ser 5432), el nombre de la BD, el nombre de usuario y la contraseña.



4. Para terminar haga clic en **Conectarse**.

Si el servidor de base de datos PostgreSQL está en otro equipo, debe tener en cuenta lo siguiente:

- El servidor de base de datos PostgreSQL debe estar configurado para aceptar conexiones de clientes. Concretamente, debe configurar el archivo **pg_hba.conf** para que permita conexiones no locales. Además debe configurar el archivo **postgresql.conf** para que escuche determinadas direcciones IP y puertos. Para más información consulte la documentación de PostgreSQL (https://www.postgresql.org/docs/9.5/static/client-authentication-problems.html).
- El equipo servidor debe estar configurado para aceptar conexiones en el puerto correspondiente (suele ser 5432) a través del servidor de seguridad. Por ejemplo, imagine que tiene un servidor de BD que se ejecuta en un equipo Windows. Deberá crear una regla que permita conexiones en el puerto 5432 a través del servidor de seguridad (desde **Panel de control > Firewall de Windows > Configuración avanzada > Reglas de entrada**).

19.1.8 Conexiones SQLite

SQLite (https://www.sqlite.org/index.html) es un tipo de base de datos basado en archivos y con almacenamiento. Las bases de datos SQLite son compatibles de forma nativa con XMLSpy, así que no es necesario instalar ningún controlador para poder conectarse a este tipo de BD.

Notas sobre compatibilidad con bases de datos SQLite

- En Linux no hay un tiempo de espera de ejecución de instrucciones para bases de datos SQLite.
- No son compatibles con búsquedas de texto completo en tablas.
- SQLite admite valores de diferentes tipos de datos en cada fila de una tabla. En MapForce todos los valores procesados deben ser compatibles con el tipo de columna declarado. Por tanto, pueden darse

© 2016-2022 Altova GmbH

errores en tiempo de ejecución si la base de datos SQLite tiene valores de fila que no coinciden con el tipo de columna declarado.

Importante

Es recomendable que use la palabra clave `STRICT` al crear tablas para asegurarse de que el comportamiento de los datos es más predecible. De lo contrario, es posible que los datos no se lean o escriban correctamente si en una misma columna hay valores de distintos tipos mezclados. Para saber más sobre las tablas `STRICT` consulte la documentation SQLite.

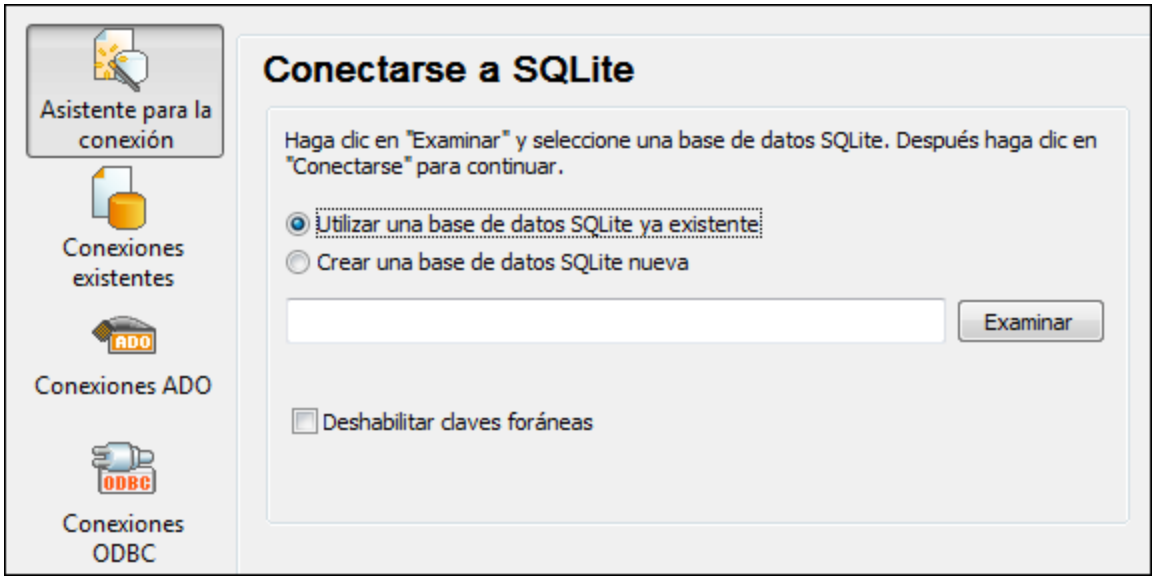
19.1.8.1 Conectarse a una base de datos SQLite

Para conectarse a una base de datos SQLite puede usar el asistente o puede conectarse directamente desde el explorador de Windows con el comando **Abrir con**.

Conectarse con el asistente para la conexión de base de datos

Para conectarse a una base de datos SQLite:

1. Inicie el asistente para la conexión de base de datos 953
2. Seleccione el botón de opción *SQLite* y después haga clic en **Siguiente**.



3. Seleccione la opción *Utilizar una base de datos SQLite ya existente* y busque el archivo de BD o introduzca su ruta de acceso (relativa o absoluta). El botón **Conectarse** se habilita una vez introducida la ruta de acceso del archivo.
4. Si quiere, ahora puede marcar la casilla *Deshabilitar claves foráneas* (*véase Restricciones de clave foránea* 983).
5. Ahora haga clic en **Conectarse**.

Conectarse desde el explorador de Windows

También puede abrir una DB SQLite directamente desde el explorador de Windows; para ello siga estos pasos:

1. En el explorador de Windows haga clic con el botón derecho en un archivo de BD en el explorador de Windows y seleccione **Abrir con** en el menú contextual.
2. Elija **Database Spy** de la lista de programas. Si no está disponible en la lista, seleccione **Elegir otra aplicación** y navegue hasta el ejecutable de Database Spy en el directorio de instalación.
3. Si aparece el Asistente para la conexión a una base de datos, haga clic en **Cerrar**.

También puede arrastrar y colocar el archivo de BD en DataBaseSpy si este está ya abierto.

Notas:

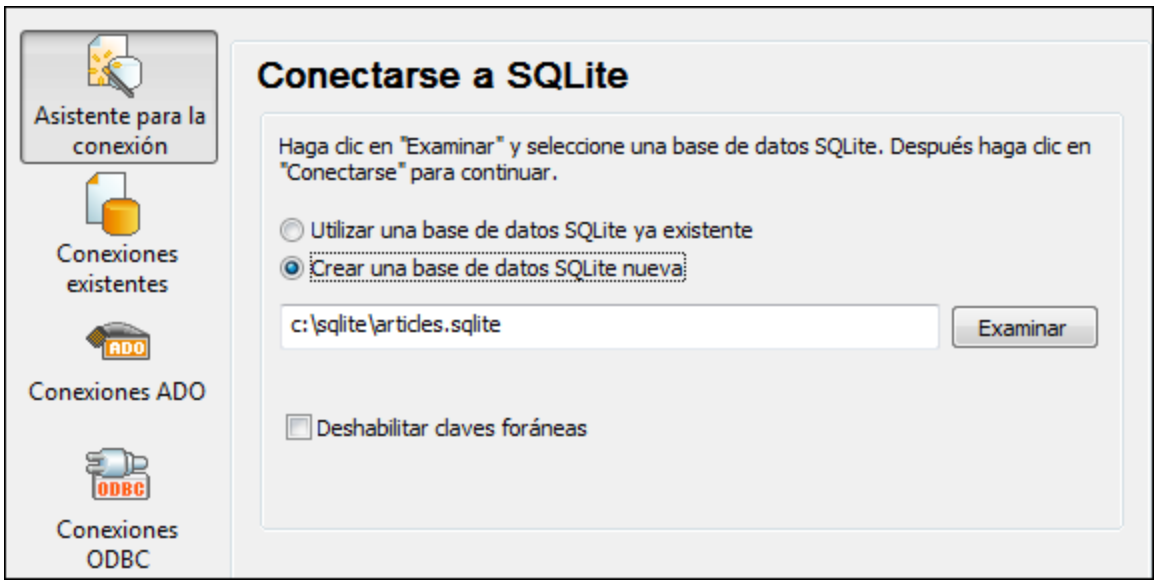
- puede configurar Database Spy desde **Herramientas | Opciones** para que no aparezca el Asistente para la conexión a una base de datos cada vez que se inicia la aplicación. Para ello, desmarque la casilla *Mostrar diálogo Crear una conexión a una base de datos* en el grupo de opciones **Generales**.
- Si DatabaseSpy está abierto y ya existe una fuente de datos para ese mismo archivo de BD, se usará este. Es decir, no se creará un duplicado. Ocurre lo mismo tanto si abre un archivo de BD con el comando de menú **Abrir como** que si lo hace mediante acciones de arrastrar y colocar.

19.1.8.2 Crear una base de datos SQLite

En lugar de conectarse a un base de datos SQLite ya existente también puede crear un archivo de BD SQLite nuevo y conectarse a él. El archivo de BD creado por XMLSpy está vacío. Deberá usar consultas y scripts para crear la estructura de BD necesaria y rellenarla con datos.

Para crear una base de datos SQLite nueva:

1. Inicie el asistente para la conexión de base de datos 953
2. Seleccione el botón de opción *SQLite* y después haga clic en **Siguiente**.



3. Seleccione la opción *Crear una base de datos SQLite nueva* e introduzca la ruta de acceso (relativa o absoluta) del archivo de BD que se debe crear. También puede hacer clic en **Examinar** para navegar

 © 2016-2022 Altova GmbH

por el sistema de archivos y seleccionar la carpeta donde se debe crear el archivo de BD. Escriba el nombre del archivo en el cuadro de texto *Nombre del archivo:* y haga clic en **Guardar**.

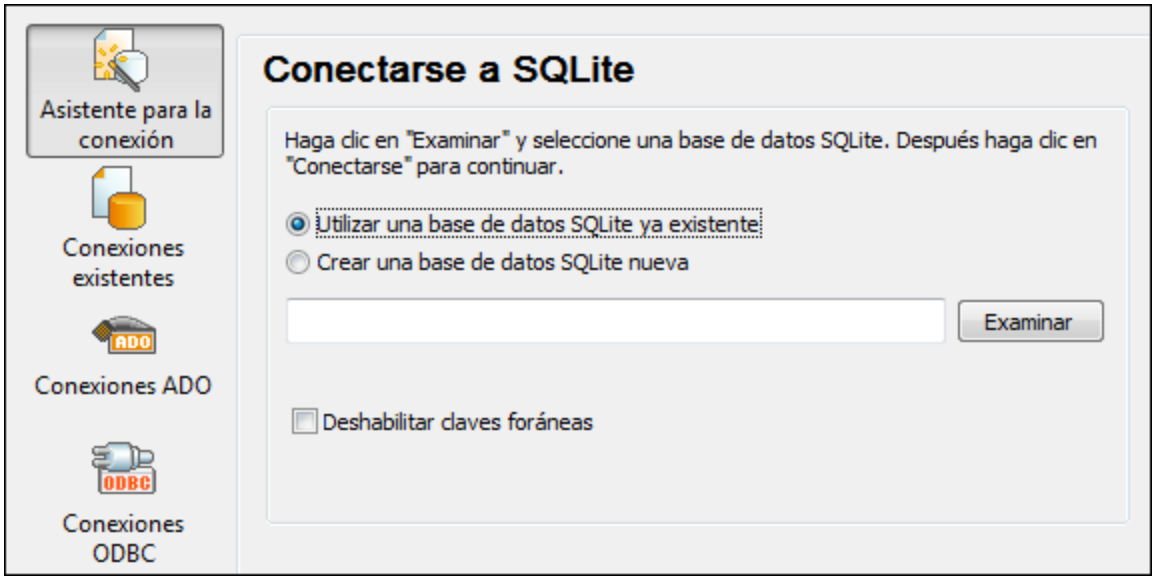
> Compruebe que tiene permisos de escritura en la carpeta donde desea crear el archivo de base de datos.

4. Si quiere, ahora puede marcar la casilla *Deshabilitar claves foráneas* (*véase* *Restricciones de clave foránea* 983).
5. Ahora haga clic en **Conectarse**.

19.1.8.3 Restricciones de clave foránea

Cuando se conecte con una base de datos SQLite desde XMLSpy o cuando cree una base de datos SQLite nueva se habilitan por defecto las restricciones de clave foránea. Las restricciones de clave foránea sirven para preservar la integridad de los datos de la base de datos. Por ejemplo, cuando las claves foráneas están habilitadas, no se pueden eliminar los registros de una tabla que tenga dependencias en otra tabla.

En algunos casos puede ser necesario invalidar temporalmente este comportamiento deshabilitando las claves foráneas. Por ejemplo, cuando necesite actualizar o insertar varias filas de datos sin obtener errores de validación de datos. Para deshabilitar las claves foráneas explícitamente antes de conectarse a la base de datos SQLite, seleccione la opción *Deshabilitar claves foráneas* del asistente para la conexión.



Página "Conectarse a SQLite" del asistente

Cuando las claves foráneas están deshabilitadas, se pueden realizar operaciones con datos que no serían posibles de otra manera (debido a las comprobaciones de validez). No obstante, al deshabilitar las claves foráneas corre el riesgo de introducir datos incorrectos en la base de datos o de crear filas huérfanas (un ejemplo de fila huérfana sería una dirección de la tabla "direcciones" que no esté vinculada con ninguna persona de la tabla "person" porque se eliminó la persona pero no su dirección.)

19.1.9 Usar una conexión a través de recursos globales

Si ya configuró una conexión de base de datos como recurso global, podrá volver a utilizar la conexión siempre que quiera e incluso desde aplicaciones de Altova diferentes.

Crear una base de datos como recurso global

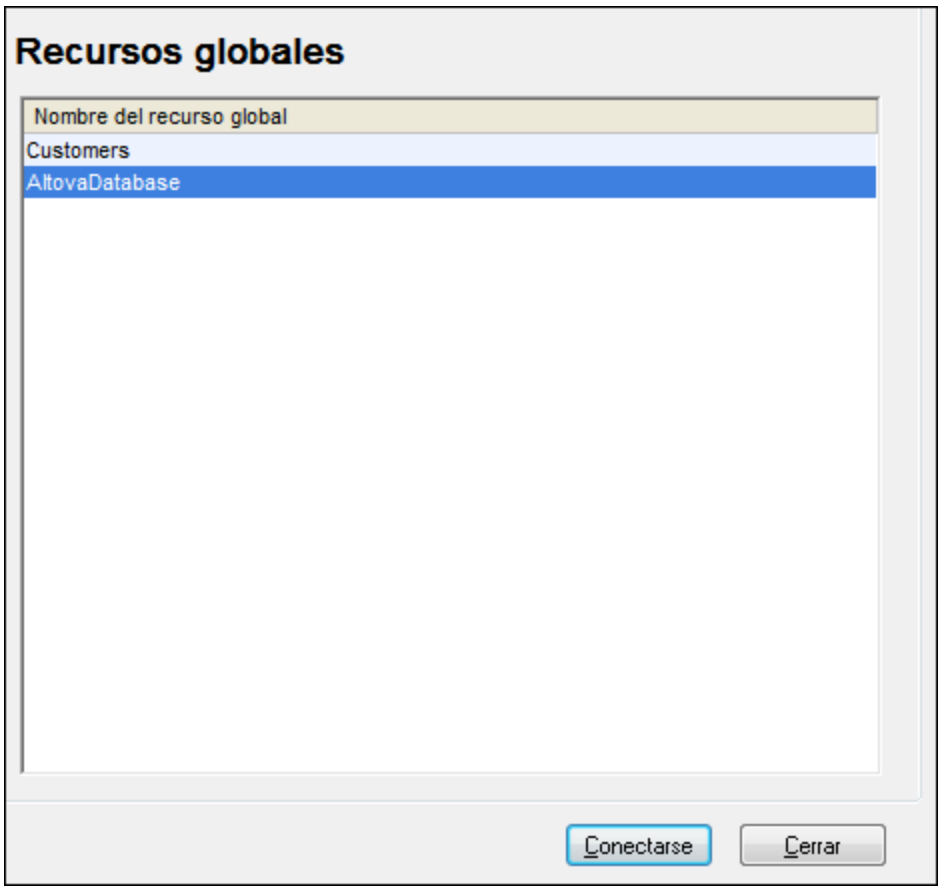
Para crear una BD como recurso global siga estos pasos:

1. En el menú **Herramientas** de XMLSpy, haga clic en **Recursos globales**.
2. Haga clic en **Agregar** y después en Base de datos.
3. Escriba un nombre para el recurso global en el campo *Alias del recurso*.
4. Haga clic en **Elegir base de datos**. Aparece el asistente para la conexión de base de datos 953.
5. Use el asistente para agregar una conexión de base de datos como hemos explicado.

Usar una conexión de base de datos disponible como recurso global

Para usar una conexión de BD disponible como recurso global (*punto anterior*) siga estos pasos:

1. Inicie el asistente para la conexión de base de datos tal y como se explica en el punto anterior.
2. Haga clic en **Recursos globales** en el panel izquierdo. En pantalla aparecen ahora las conexiones de base de datos configuradas previamente como recurso global (*imagen siguiente*).



© 2016-2022 Altova GmbH

3. Seleccione la conexión de base de datos correspondiente y haga clic en **Conectarse**. Consejo: para obtener información sobre un recurso global pase el puntero por encima de la entrada.

19.1.10 Ejemplos de conexión a bases de datos

Esta sección incluye ejemplos de cómo conectarse a una base de datos desde XMLSpy con ADO, ODBC o JDBC. Para ver los ejemplos de ADO.NET consulte el apartado Ejemplo: cadenas de conexión ADO.NET[969]. Para ver cómo establecer una conexión nativa a PostgreSQL y SQLite (véanse respectivamente Conexiones PostgreSQL[979] y Conexiones SQLite[980]).

Tenga en cuenta lo siguiente:

- Puede que las instrucciones sean distintas si su configuración de Windows, entorno de red y cliente de BD o software de servidor no son los mismos que los descritos en el ejemplo correspondiente.
- En la mayoría de los casos se puede establecer la conexión por medio de varias tecnologías de acceso (ADO, ADO.NET, ODBC, JDBC) o controladores. El rendimiento de la conexión de BD, así como sus características y limitaciones, dependerá del controlador, software cliente y parámetros de conexión seleccionados.

19.1.10.1 Firebird (JDBC)

Este ejemplo explica cómo conectarse a una base de datos Firebird por JDBC.

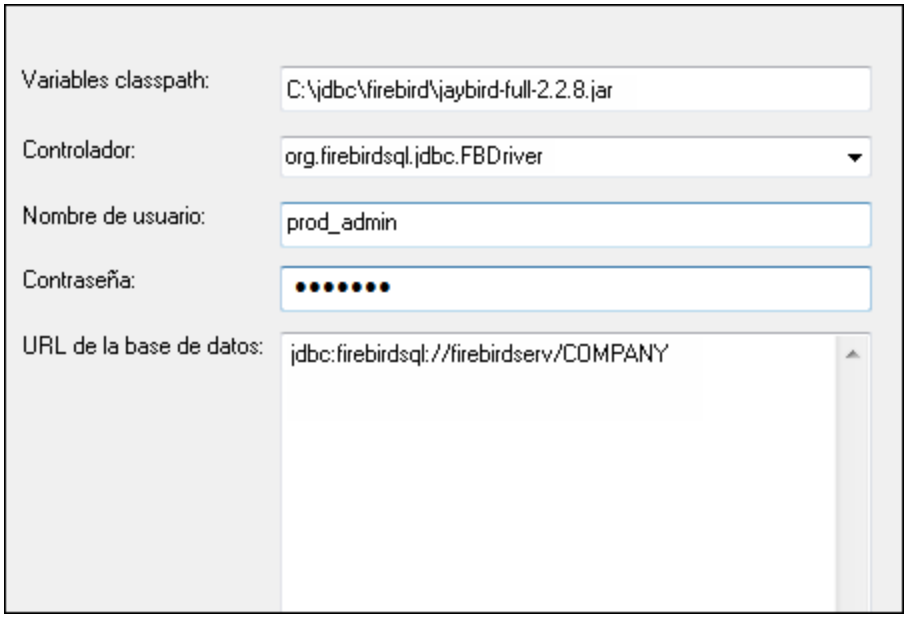
Requisitos:

- JRE (Java Runtime Environment) o Java Development Kit (JDK) está instalado. Este último puede ser el JDK de Oracle o uno de código abierto, como Oracle OpenJDK. XMLSpy identifica la ruta de acceso al equipo virtual Java (JVM) a partir de estas ubicaciones, en este orden: a) la ruta personal de acceso al JVM que puede indicar en en las **Opciones** ; b) la ruta de acceso al JVM que se encuentra en el registro; c) la variable de entorno `JAVA_HOME`.
- Asegúrese de que la plataforma de XMLSpy (32 o 64 bits) es la misma que la de JRE/JDK.
- El controlador Firebird JDBC debe estar disponible en el sistema operativo (se trata de un archivo .jar que ofrece conectividad con la base de datos). El controlador se puede descargar del sitio web de Firebird (https://www.firebirdsql.org/). En este ejemplo usamos el controlador *Jaybird 2.2.8*.
- Disponer de los datos de conexión: host, ruta de acceso (o alias) de la base de datos, nombre de usuario y contraseña.

Para conectarse a Firebird por JDBC:

1. Inicie el asistente para la conexión de base de datos[953].
2. Haga clic en **Conexiones JDBC**.
3. En el campo *Variables classpath* debe introducir la ruta de acceso del archivo .jar que ofrece conectividad con la base de datos. Si fuera necesario, también puede introducir una lista de rutas de archivo .jar separadas por caracteres de punto y coma. En este ejemplo, el archivo .jar está ubicado en esta ruta de acceso: `C:\jdbc\firebird\jaybird-full-2.2.8.jar`. Tenga en cuenta que este campo puede dejarse en blanco si añadió la ruta de acceso de los archivos .jar a la variable de entorno CLASSPATH del sistema operativo (*véase Configurar la variable CLASSPATH*[977]).

4. En el campo *Controlador* seleccione **org.firebirdsql.jdbc.FBDriver**. Recuerde que esta entrada solo estará disponible si se encuentra una ruta de archivo .jar válida en el campo Variables *classpath* o en la variable de entorno CLASSPATH del sistema operativo.



5. Introduzca el nombre de usuario y la contraseña de la base de datos.
6. Introduzca la cadena de conexión para el servidor de BD en el cuadro de texto *URL de la base de datos* (reemplace lo valores resaltados con los de su base de datos).

```
jdbc:firebirdsql://<host>[:<puerto>]/<ruta o alias de la BD>
```

7. Haga clic en **Conectarse**.

19.1.10.2 Firebird (ODBC)

Este ejemplo explica cómo conectarse a una base de datos Firebird 2.5.4 de un servidor Linux.

Requisitos:

- El servidor de BD Firebird está configurado para aceptar conexiones TCP/IP desde clientes.
- El controlador ODBC de Firebird está instalado en el sistema. Este ejemplo usa la versión 2.0.3.154 del controlador (https://www.firebirdsql.org/).
- El cliente Firebird está instalado en el sistema. Recuerde que el cliente forma parte del paquete de instalación del servidor Firebird. Puede descargar este paquete del sitio web de Firebird (https://www.firebirdsql.org/) y buscar el instalador para Windows (Windows executable installer for full Superclassic/Classic or Superserver). Para instalar los archivos del cliente solamente elija la opción **Minimum client install - no server, no tools** del asistente para la instalación.

© 2016-2022 Altova GmbH

Nota:

- La plataforma del controlador ODBC de Firebird y del cliente (de 32 o 64 bits) debe coincidir con la plataforma de XMLSpy.
- La versión del cliente Firebird debe coincidir con la versión del servidor Firebird al que desea conectarse.

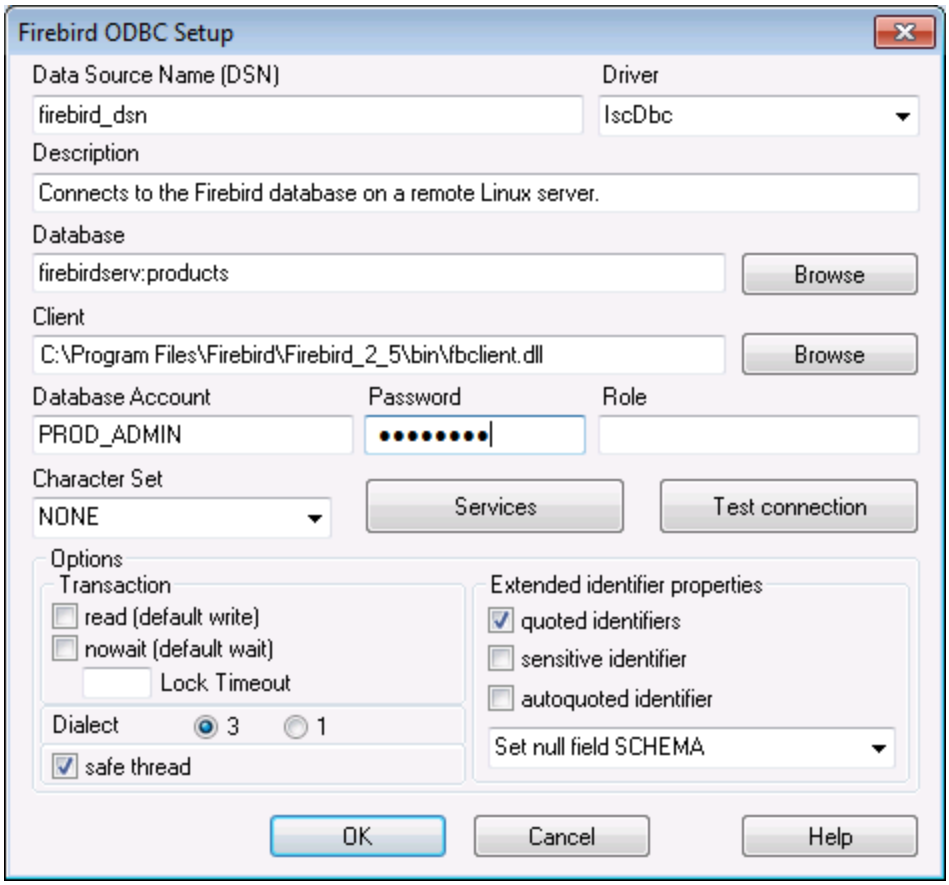
- Disponer de los datos de conexión: nombre de host o dirección IP, ruta de acceso (o alias) de la base de datos en el servidor, nombre de usuario y contraseña.

Para conectarse a Firebird por ODBC:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en **Conexiones ODBC**.
3. Seleccione la opción *DSN de usuario* (o *DSN de sistema* si tiene privilegios de administrador) y después haga clic en **Agregar** .



4. Seleccione el controlador Firebird y después haga clic en *DSN de usuario* (o *DSN de sistema* dependiendo de la opción seleccionada en el paso anterior). Si el controlador Firebird no aparece en la lista, compruebe que está instalado en el sistema.



5. Introduzca los datos de conexión:

Nombre del origen de datos (DSN)	Introduzca un nombre para el origen de datos que desea crear.

Base de datos	Introduzca el nombre de host o dirección IP del servidor, seguida de dos puntos, seguido del alias (o ruta de acceso) de la base de datos. En este ejemplo el nombre de host es `firebirdserv` y el alias de la BD es `products`:
	
	`firebirdserv:products`
	
	Usamos el alias de la BD porque damos por hecho que en el lado servidor el administrador de la BD ha configurado el alias *products* para apuntar al archivo de BD Firebird (.fdb) en el servidor.
	
	En lugar del nombre de host también puede usar la dirección IP del servidor. Y en lugar del alias puede usar una ruta de acceso. Por tanto, también podría usar estas otras dos cadenas de conexión:
	
	`firebirdserver:/var/Firebird/databases/butterflies.fdb`
	`127.0.0.1:D:\Misc\Lenders.fdb`

 © 2016-2022 Altova GmbH

	Si la base de datos está en el equipo Windows local, haga clic en **Examinar** y seleccione el archivo de base de datos directamente.
Cliente	Introduzca la ruta de acceso del archivo `fbclient.dll`. Su ubicación predeterminada es el subdirectorio `bin` del directorio de instalación de Firebird.
Cuenta de base de datos	Introduzca el nombre de usuario que recibió del administrador de la BD (en este ejemplo es `PROD_ADMIN`).
Contraseña	Introduzca la contraseña de la BD que recibió del administrador.

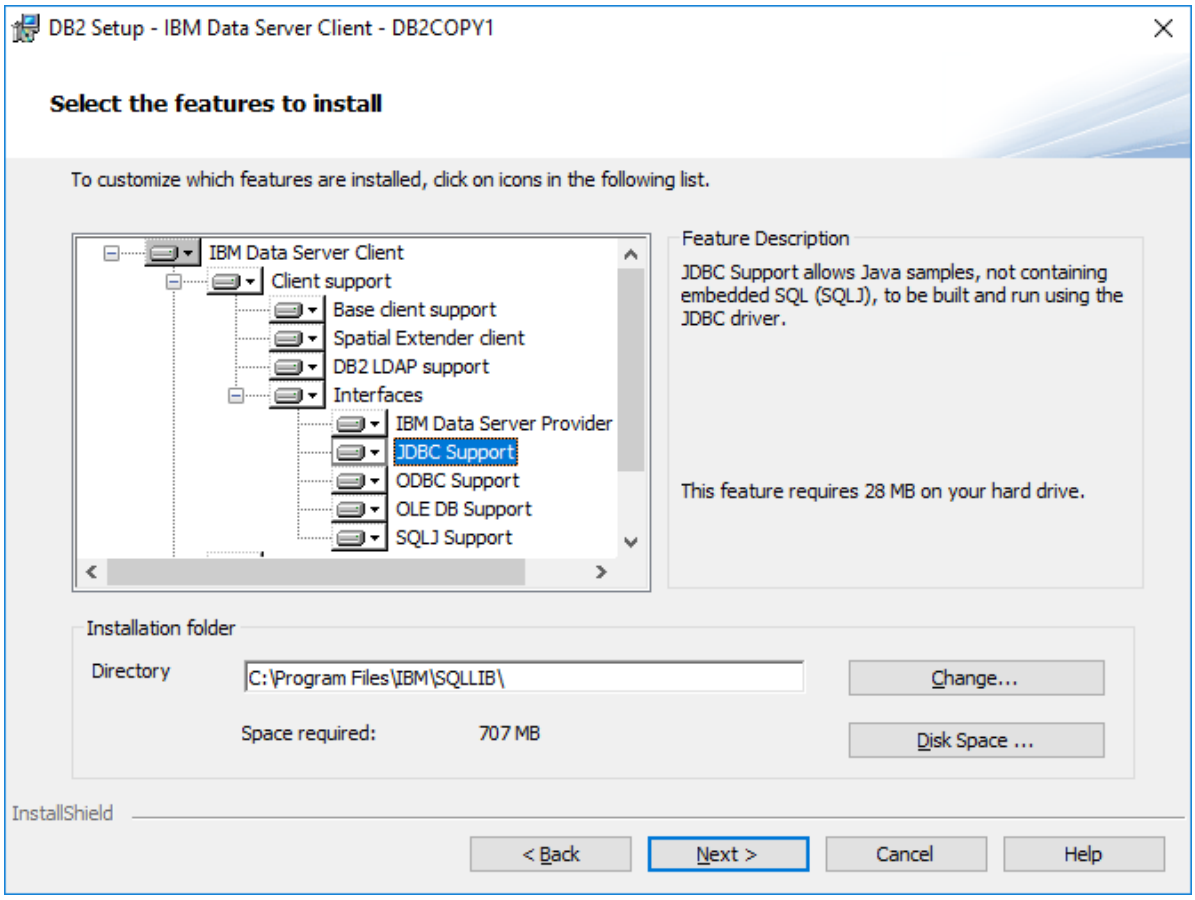
6. Haga clic en **Aceptar**.

19.1.10.3 IBM DB2 (JDBC)

Este ejemplo explica cómo conectarse a una base de datos IBM DB2 por JDBC.

Requisitos:

- JRE (Java Runtime Environment) o Java Development Kit (JDK) está instalado. Este último puede ser el JDK de Oracle o uno de código abierto, como Oracle OpenJDK. XMLSpy identifica la ruta de acceso al equipo virtual Java (JVM) a partir de estas ubicaciones, en este orden: a) la ruta personal de acceso al JVM que puede indicar en en las **Opciones** ; b) la ruta de acceso al JVM que se encuentra en el registro; c) la variable de entorno `JAVA_HOME`.
- Asegúrese de que la plataforma de XMLSpy (32 o 64 bits) es del mismo tipo que la de JRE/JDK. Este ejemplo usa Oracle OpenJDK 11.0 de 64 bits, por lo que usa la versión de 64 bits de XMLSpy.
- El controlador JDBC (uno o más archivos .jar que permiten conectarse a la BD) debe estar disponible en el sistema operativo. Este ejemplo usa el controlador JDBC que está disponible tras instalar la versión 10.1 del **IBM Data Server Client** (64 bits). Al instalar los controladores, elija la instalación típica o seleccione esa opción en el asistente de instalación.



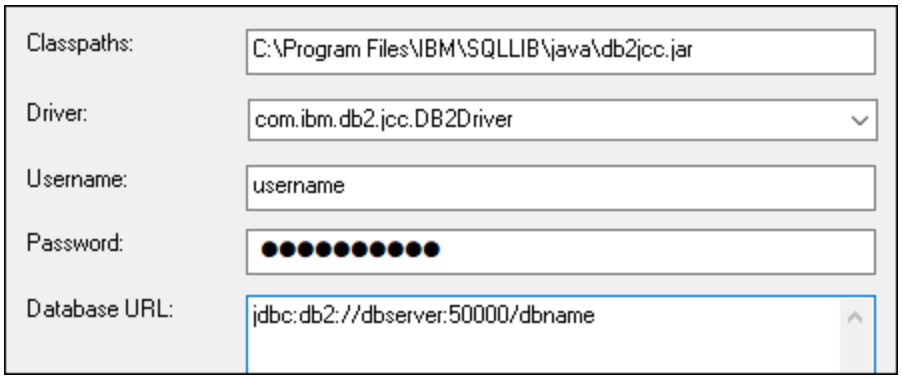
Si no ha modificado la ruta predeterminada de la instalación, una vez esta haya finalizado los archivos .jar requeridos están en el directorio **C:\Program Files\IBM\SQLLIB\java**.

- Necesitará esta información sobre la conexión a la BD: host, puerto, nombre de la BD, nombre de usuario y contraseña.

Para conectarse a IBM DB2 for i por JDBC:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en **Conexiones JDBC**.
3. Junto a "Variables classpath" introduzca la ruta al archivo .jar que permite conectarse a la BD. En este ejemplo la ruta es **C:\Program Files\IBM\SQLLIB\java\db2jcc.jar**. Puede que necesite hacer referencia al controlador **db2jcc4.jar** en función de la versión del servidor de BD. Si tiene dudas sobre si el controlador es compatible consulte la documentación de IBM (http://www-01.ibm.com/support/docview.wss?uid=swg21363866). Puede dejar el campo "Variables classpath" vacío si ha añadido la ruta de acceso del archivo .jar (también pueden ser varios) a la variable de entorno CLASSPATH del sistema operativo (*véase también el apartado Configurar la variable CLASSPATH* 977).
4. En el campo "Controlador" seleccione **com.ibm.db2.jcc.DB2Driver**. Esta entrada solo está disponible si se encuentra una ruta de acceso válida a un archivo .jar en el campo "Variables classpath" o en la variable de entorno CLASSPATH del sistema operativo (*véase el paso anterior*).

© 2016-2022 Altova GmbH



5. Introduzca el nombre de usuario y la contraseña del usuario de la BD en los campos correspondientes.
6. Introduzca la cadena de conexión JDBC en el campo **URL de la BD**. Asegúrese de reemplazar los detalles de la conexión con los de su servidor de BD.

```
jdbc:db2://hostName:port/databaseName
```

7. Haga clic en **Conectarse**.

19.1.10.4 IBM DB2 (ODBC)

Este ejemplo explica cómo conectarse a una base de datos IBM DB2 por ODBC.

Requisitos:

- El cliente IBM Data Server está instalado y configurado en el sistema (en este ejemplo usamos IBM Data Server Client 9.7). Después de instalar el cliente IBM Data Server compruebe que los controladores ODBC están disponibles en el equipo (consulte Ver los controladores ODBC disponibles 973).
- Tiene un alias para la base de datos. El alias se puede crear de varias maneras:
  - Con el asistente de configuración de IBM DB2
  - Con el procesador de línea de comandos de IBM DB2
  - Con el asistente para orígenes de datos ODBC (instrucciones más abajo).
- Disponer de los datos de conexión: host, base de datos, puerto, nombre de usuario y contraseña.

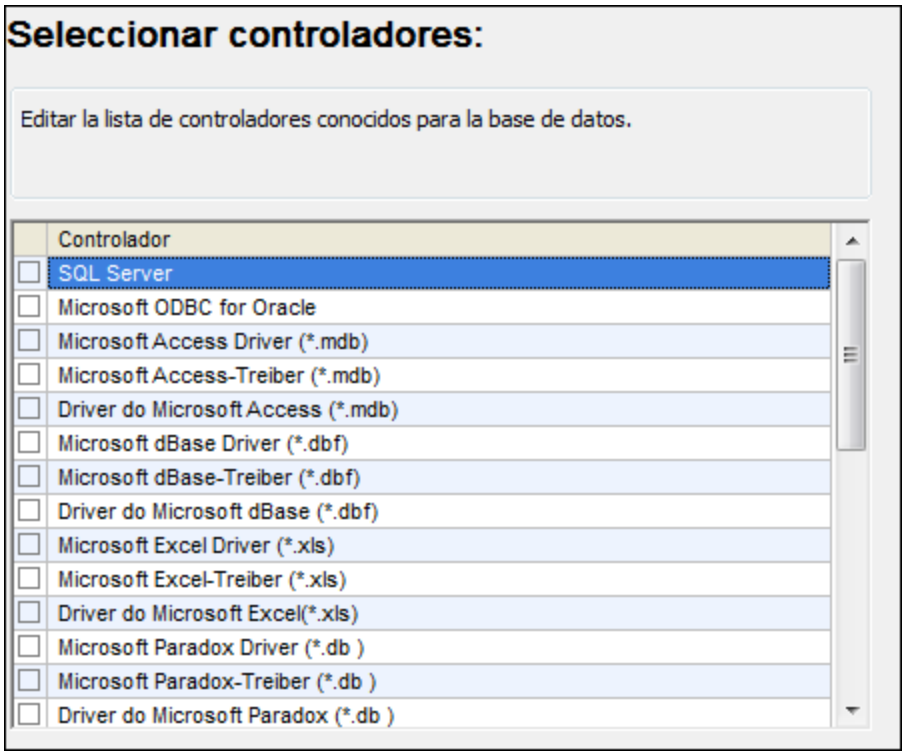
Para conectarse a IBM DB2:

1. Inicie el asistente para la conexión de base de datos 953 y seleccione la opción *IBM DB2 (ODBC/JDBC)*.
2. Haga clic en **Siguiente**.



3. Seleccione la opción *ODBC* y haga clic en **Siguiente**. Si necesita editar la lista de controladores conocidos para la base de datos, seleccione los controladores que corresponden a IBM DB2 (ver Requisitos [991]) y haga clic en **Siguiente**.

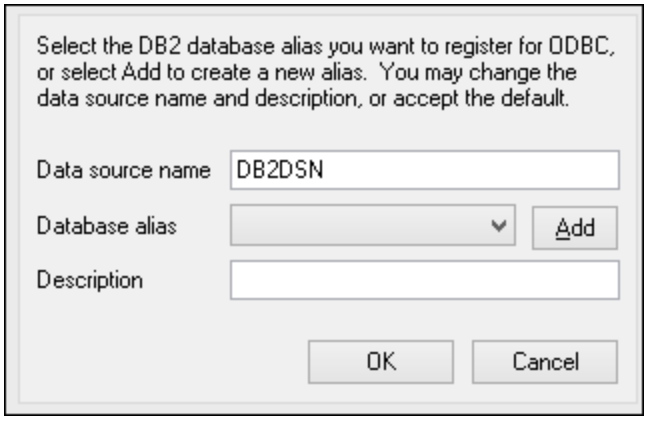
© 2016-2022 Altova GmbH



4. Seleccione el controlador de la lista y haga clic en **Conectarse**. (Para editar la lista de controladores disponibles haga clic en **Editar controladores** y active/desactive los controladores que desea agregar o eliminar)

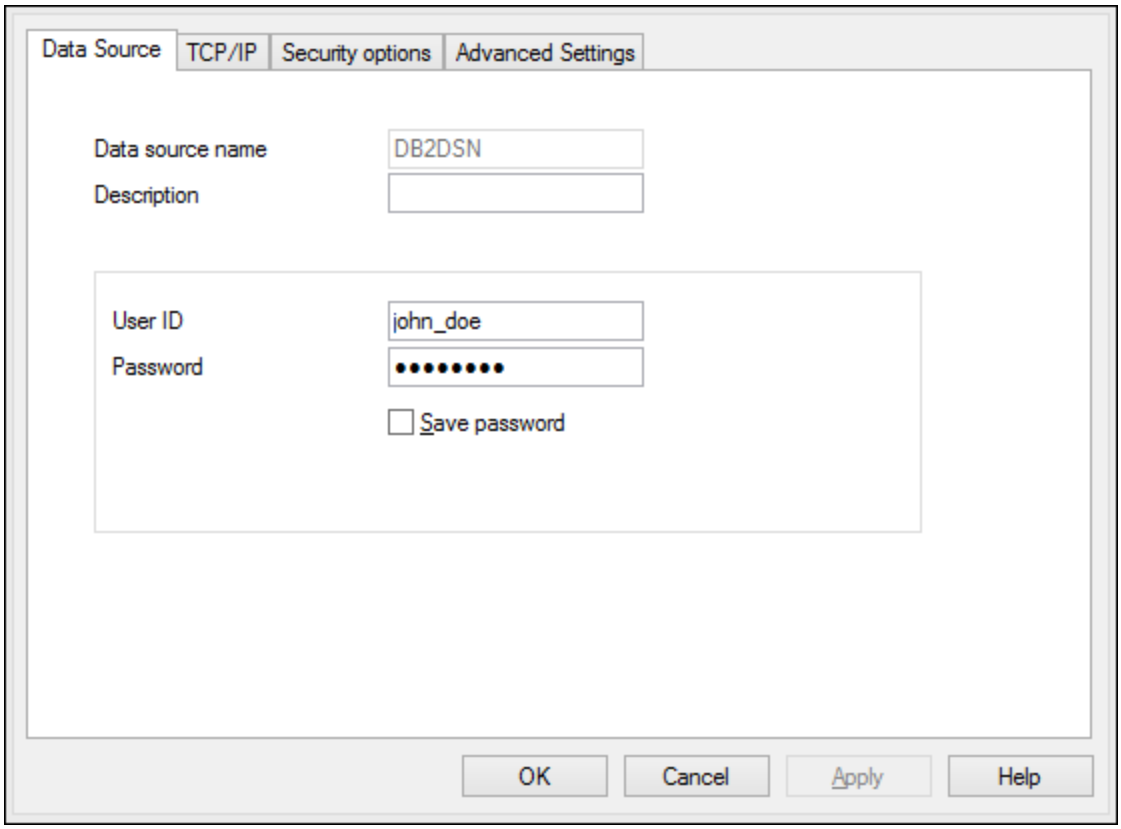


5. Introduzca el DSN (**DB2DSN**) y haga clic en **Agregar**.

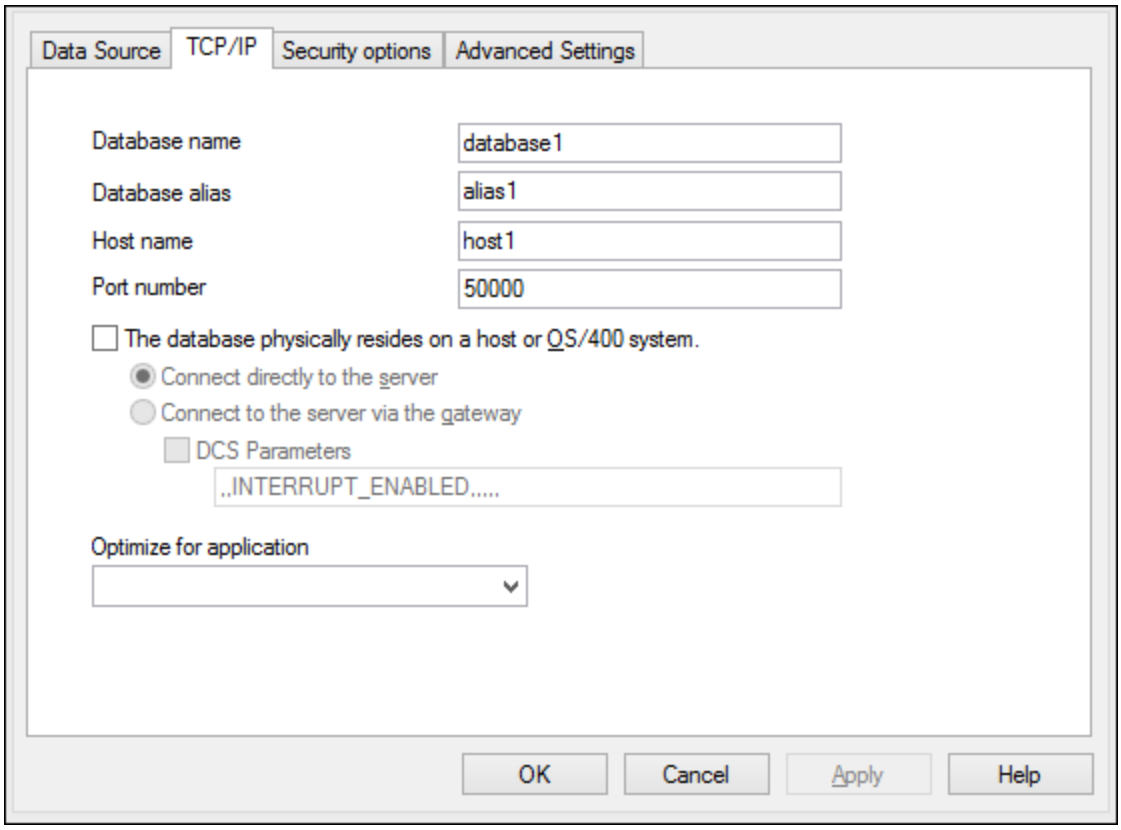


6. En la pestaña *Origen de datos* introduzca el nombre de usuario y la contraseña de la base de datos.

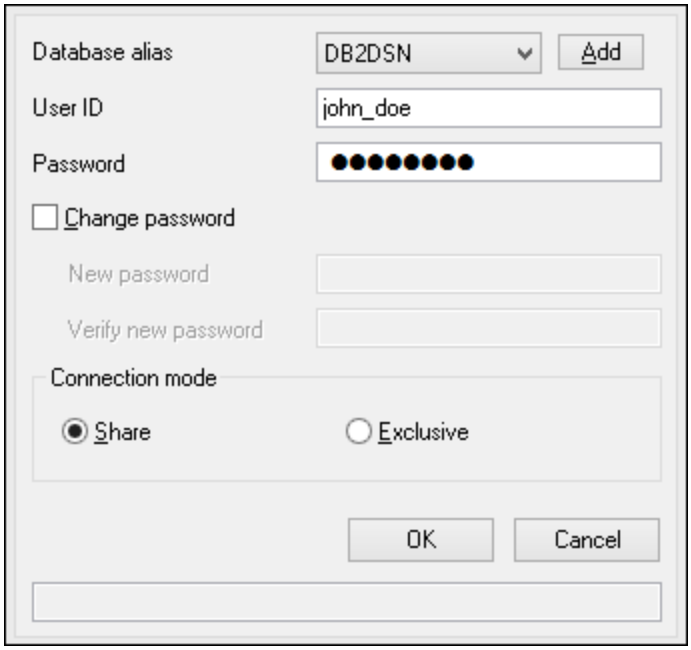
© 2016-2022 Altova GmbH



7. En la pestaña **TCP/IP** introduzca el nombre de la base de datos, un nombre para el alias, el nombre de host y el número de puerto. Después haga clic en **Aceptar**.



8. Vuelva a introducir el nombre de usuario y la contraseña y haga clic en **Aceptar**.



 © 2016-2022 Altova GmbH

19.1.10.5 IBM DB2 para i (JDBC)

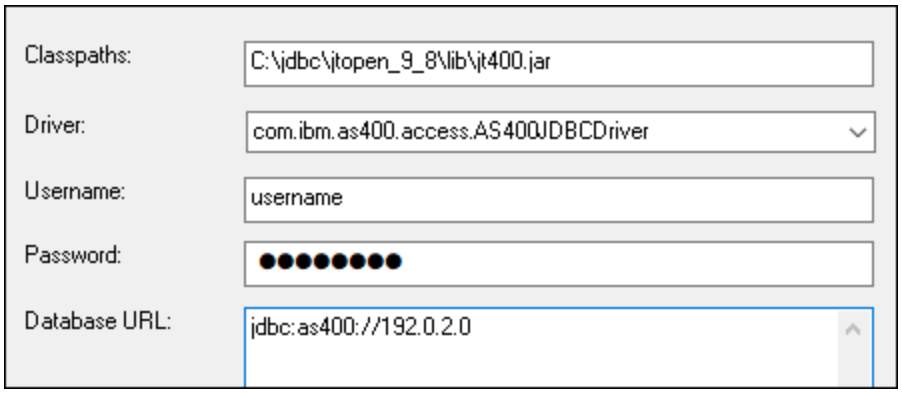
Este ejemplo explica cómo conectarse a una base de datos IBM DB2 for i por JDBC.

Requisitos:

- JRE (Java Runtime Environment) o Java Development Kit (JDK) está instalado. Este último puede ser el JDK de Oracle o uno de código abierto, como Oracle OpenJDK. XMLSpy identifica la ruta de acceso al equipo virtual Java (JVM) a partir de estas ubicaciones, en este orden: a) la ruta personal de acceso al JVM que puede indicar en en las **Opciones** ; b) la ruta de acceso al JVM que se encuentra en el registro; c) la variable de entorno `JAVA_HOME`.
- Asegúrese de que la plataforma de XMLSpy (32 o 64 bits) es del mismo tipo que la de JRE/JDK. Este ejemplo usa Oracle OpenJDK 11.0 de 64 bits, por lo que usa la versión de 64 bits de XMLSpy.
- El controlador JDBC (uno o más archivos .jar que permitan conectarse a la BD) debe estar disponible en su sistema operativo. En este ejemplo se usa la biblioteca de código abierto **Toolbox for Java/JTOpen** versión 9.8 (http://jt400.sourceforge.net/). Una vez haya descargado el paquete y lo haya desempaquetado en un directorio local, los archivos .jar necesarios estarán disponibles en el subdirectorio **lib**.
- Necesitará esta información sobre la conexión a la BD: host, nombre de usuario y contraseña.

Para conectarse a IBM DB2 for i con JDBC:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en **JDBC Connections**.
3. Junto a "Variables classpath" introduzca la ruta de acceso al archivo .jar que permite conectarse a la BD. En este ejemplo la ruta es **C:\jdbc\jtopen_9_8\jt400.jar**. Tenga en cuenta que puede dejar vacío el campo "Variables classpath" si ha añadido la ruta de acceso al archivo .jar (también pueden ser varios) a la variable de entorno CLASSPATH del sistema operativo (*véase también Configurar la variable CLASSPATH* 977).
4. En el campo "Controlador" seleccione **com.ibm.as400.access.AS400JDBCDriver**. Esta entrada solo está disponible si se encuentra una ruta de acceso válida a un archivo .jar en el campo "Variables classpath" o en la variable de entorno CLASSPATH del sistema operativo (*véase el paso anterior*).



5. Introduzca el nombre de usuario y la contraseña del usuario de la BD en los campos correspondientes.
6. Introduzca la cadena de conexión JDBC en el campo **URL de la BD**. Asegúrese de reemplazar `host` con el nombre de host o la dirección IP de su servidor de BD.

```
jdbc:as400://host
```

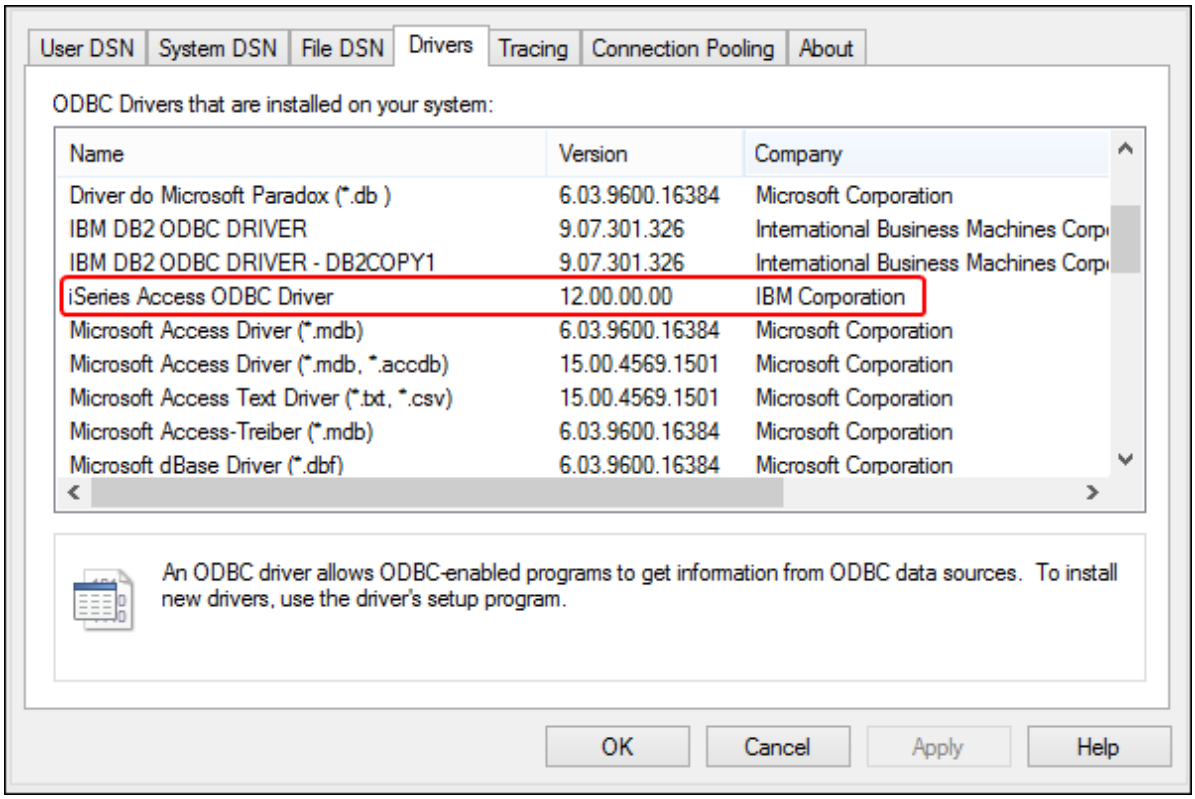
7. Haga clic en **Conectarse**.

19.1.10.6 IBM DB2 para i (ODBC)

Este ejemplo explica cómo conectarse a una base de datos IBM DB2 for i por ODBC.

Requisitos:

- *IBM System i Access for Windows* está instalado en el sistema (para este ejemplo usamos *IBM System i Access for Windows V6R1M0*). Compruebe que el controlador ODBC está en el equipo (consulte Ver los controladores ODBC disponibles 973).

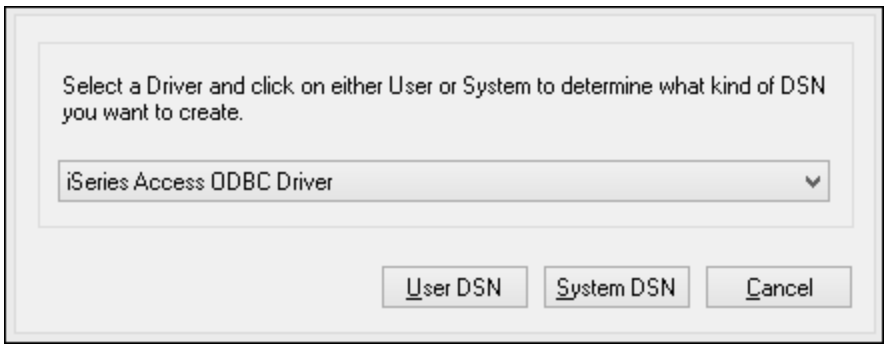


- Disponer de los datos de conexión: dirección IP del servidor de BD, nombre de usuario y contraseña.
- Ejecute *System i Navigator* y siga las instrucciones que aparecen en pantalla para crear una conexión nueva. Llegado el momento de especificar un sistema, introduzca la dirección IP del servidor de BD. Tras crear la conexión se recomienda verificarla (haga clic en la conexión y seleccione **Archivo** > **Diagnóstico** > **Verificar conexión**). Si recibe errores de conexión, póngase en contacto con el administrador del servidor de BD.

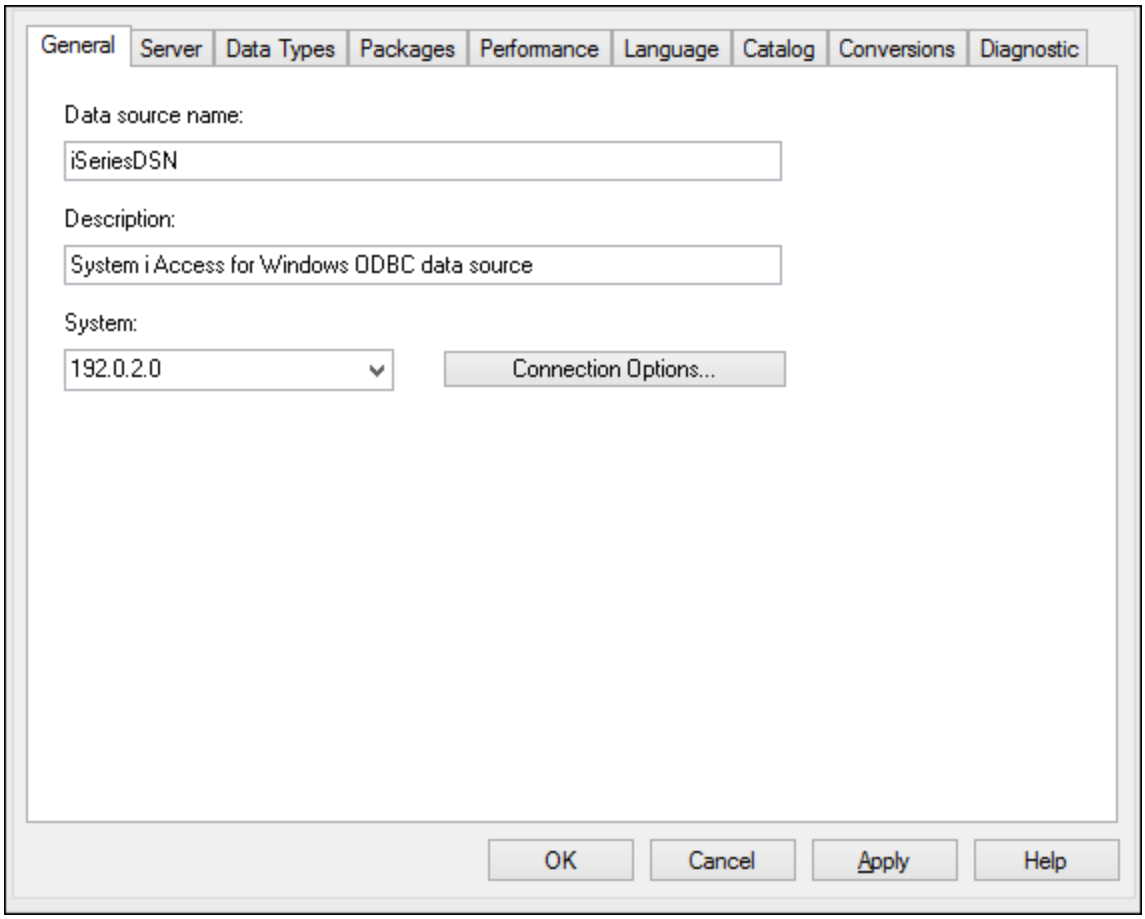
© 2016-2022 Altova GmbH

Para conectarse a IBM DB2 for i:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en **Conexiones ODBC**.
3. Seleccione *DSN de usuario* (o *DSN de sistema* o *DSN de archivo*).
4. Haga clic en **Agregar** .
5. Seleccione el controlador **iSeries Access ODBC Driver** de la lista y después haga clic en **DSN de usuario** (o **DSN de sistema**).

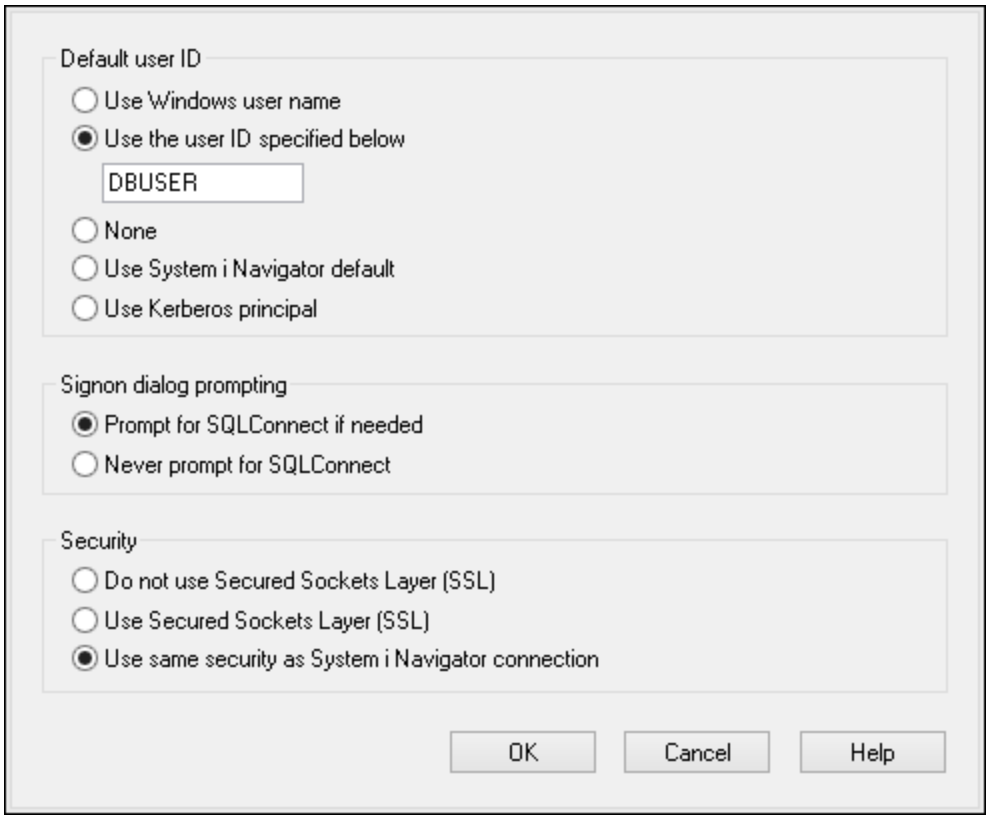


6. Introduzca el DSN y seleccione la conexión del cuadro combinado *Sistema*. En este ejemplo el DSN es **iSeriesDSN** y el sistema es **192.0.2.0**.



7. Haga clic en el botón **Opciones de conexión** y seleccione *Utilizar el Id. de usuario indicado* y escriba el nombre del usuario de la BD (en este ejemplo es **DBUSER**).

© 2016-2022 Altova GmbH



8. Haga clic en **Aceptar**. El origen de datos nuevo aparece ahora en la lista de DSN.
9. Haga clic en **Conectarse**.
10. Introduzca el nombre de usuario y la contraseña de la BD cuando sea necesario y después haga clic en **Aceptar**.

19.1.10.7 IBM Informix (JDBC)

Este ejemplo explica cómo conectarse a una base de datos IBM por JDBC.

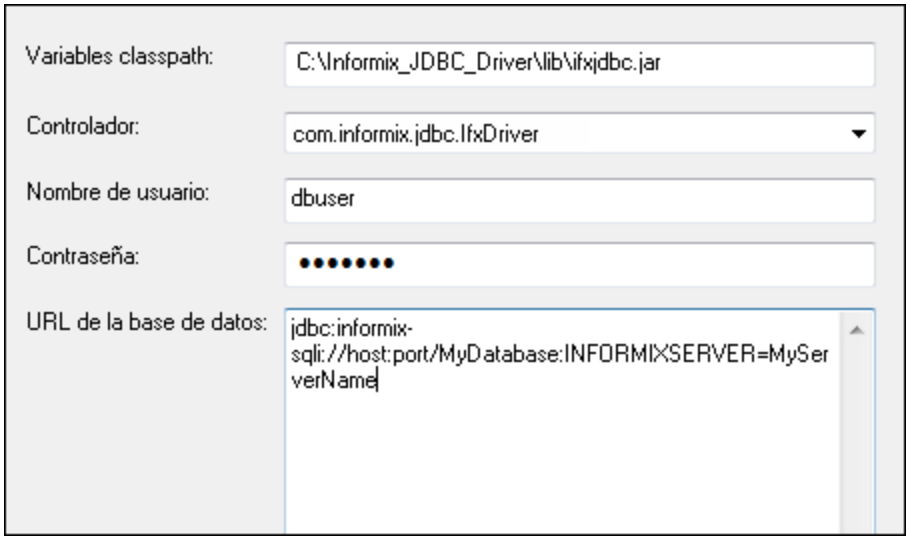
Requisitos:

- Debe tener instalado Java Runtime Environment (JRE) o Java Development Kit (JDK). Este último puede ser el JDK de Oracle o uno de código abierto, como Oracle OpenJDK. XMLSpy identifica la ruta de acceso al equipo virtual Java (JVM) a partir de estas ubicaciones, en este orden: a) la ruta personal de acceso al JVM que puede indicar en en las **Opciones** ; b) la ruta de acceso al JVM que se encuentra en el registro; c) la variable de entorno `JAVA_HOME`.
- Asegúrese de que la plataforma de XMLSpy (32 o 64 bits) es la misma que la de JRE/JDK.
- El controlador JDBC (se trata de un archivo .jar o varios archivos .jar que ofrecen conectividad con la base de datos) debe estar disponible en el sistema operativo. En este ejemplo se utiliza el controlador JDBC IBM Informix versión 3.70. Consulte la documentación del controlador para ver las instrucciones de instalación o la guía *IBM Informix JDBC Driver Programmer's Guide*.

- Disponer de los datos de conexión: host, nombre del servidor Informix, base de datos, puerto, nombre de usuario y contraseña.

Para conectarse a IBM Informix por JDBC:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en **Conexiones JDBC**.
3. En el campo *Variables classpath* debe introducir la ruta de acceso del archivo .jar que ofrece conectividad con la base de datos. Si fuera necesario, también puede introducir una lista de rutas de archivo .jar separadas por caracteres de punto y coma. En este ejemplo, el archivo .jar está ubicado en esta ruta de acceso: `C:\Informix_JDBC_Driver\lib\ifxjdbc.jar`. Tenga en cuenta que este campo puede dejarse en blanco si añadió la ruta de acceso de los archivos .jar a la variable de entorno CLASSPATH del sistema operativo (*véase Configurar la variable CLASSPATH* 977).
4. En el campo *Controlador* seleccione **com.informix.jdbc.IfxDriver**. Recuerde que esta entrada solo estará disponible si se encuentra una ruta de archivo .jar válida en el campo Variables *classpath* o en la variable de entorno CLASSPATH del sistema operativo.



5. Introduzca el nombre de usuario y la contraseña de la base de datos.
6. Introduzca la cadena de conexión para el servidor de BD en el cuadro de texto *URL de la base de datos* (reemplace lo valores resaltados con los de su base de datos).

```
jdbc:informix-sqli://nombreHost:puerto/nombreBD:INFORMIXSERVER=myserver;
```

7. Haga clic en **Conectarse**.

© 2016-2022 Altova GmbH

19.1.10.8 MariaDB (ODBC)

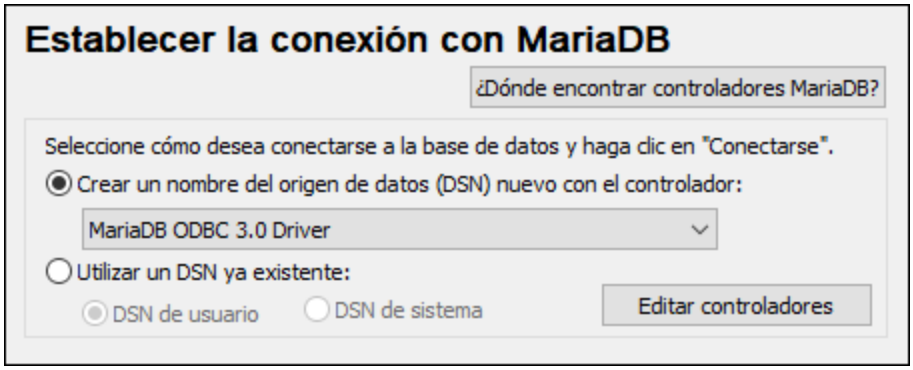
Este ejemplo explica cómo conectarse a un servidor de base de datos MariaDB por ODBC.

Requisitos:

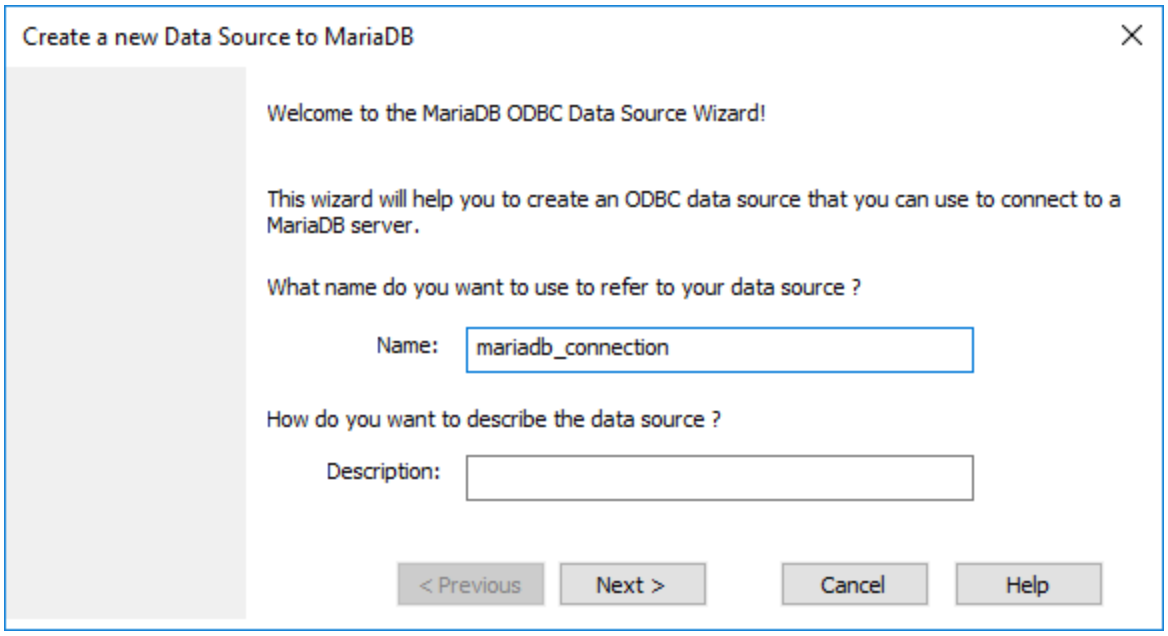
- Debe tener instalado el conector de MariaDB para ODBC (https://downloads.mariadb.org/connector-odbc/).
- Disponer de los datos de conexión: host, base de datos, puerto, nombre de usuario y contraseña.

Para conectarse a MariaDB por ODBC:

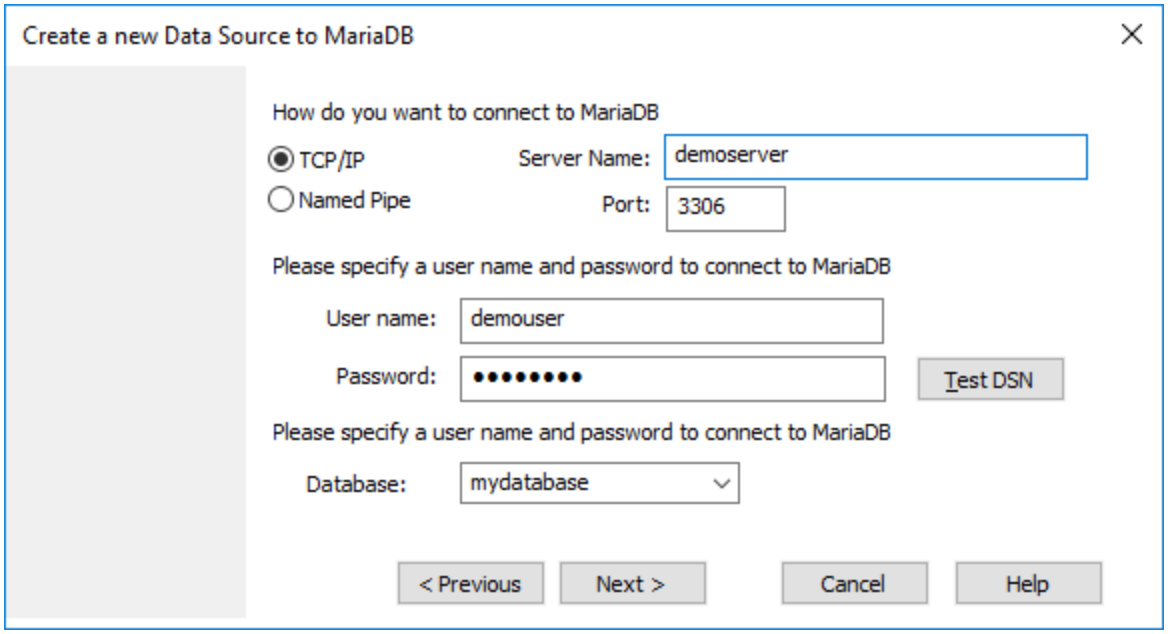
1. Inicie el asistente para la conexión de base de datos 953.
2. Seleccione *MariaDB (ODBC)* y después haga clic en **Siguiente**.



3. Seleccione *Crear un nombre del origen de datos (DSN) nuevo con el controlador* y elija el controlador **MariaDB ODBC 3.0 Driver**. Si este controlador no aparece en la lista, haga clic en **Editar controladores** y seleccione cualquier controlador MariaDB (la lista contiene todos los controladores ODBC que están instalados en el sistema operativo).
4. Haga clic en **Conectarse**.

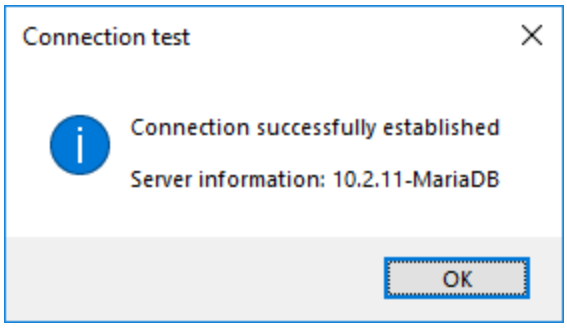


5. Introduzca el nombre y, si quiere, una descripción que le ayude a identificar este origen de datos ODBC más adelante.



6. Rellene las credenciales de la conexión de base de datos (servidor TCP/IP, usuario, contraseña), seleccione una base de datos y después haga clic en **Probar DSN**. Cuando se establezca la conexión aparecerá este mensaje:

© 2016-2022 Altova GmbH



7. Haga clic en **Siguiente** y siga los pasos del asistente hasta el final. Dependiendo del caso, puede que sean necesarios más parámetros (p. ej. certificados SSL si se conecta a MariaDB a través de una conexión segura).

Nota: si el servidor de base de datos es remoto, deberá estar configurado por el administrador para que acepte conexiones remotas desde la dirección IP de su equipo.

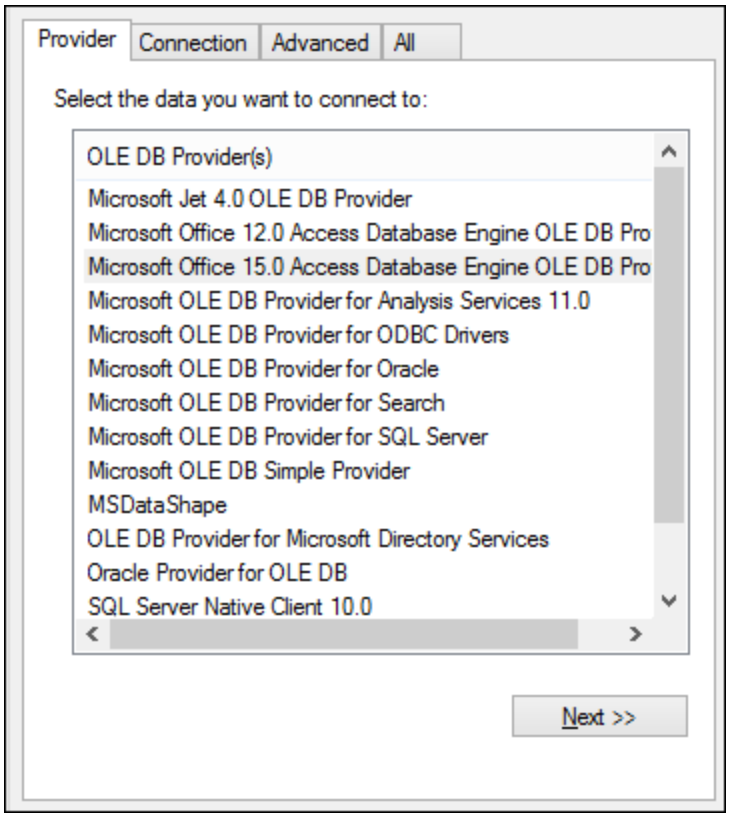
19.1.10.9 Microsoft Access (ADO)

Lo más sencillo para conectarse a una base de datos Microsoft Access es seguir las instrucciones del asistente para la conexión de base de datos y buscar el archivo de base de datos, como se muestra en el apartado Conectarse a una base de datos Microsoft Access[960]. También puede configurar explícitamente una conexión ADO, como muestra este ejemplo. Esta segunda opción se recomienda si la base de datos está protegida con contraseña.

También puede conectarse a Microsoft Access por ODBC, pero esto implica algunas restricciones por lo que recomendamos evitar este tipo de conexión.

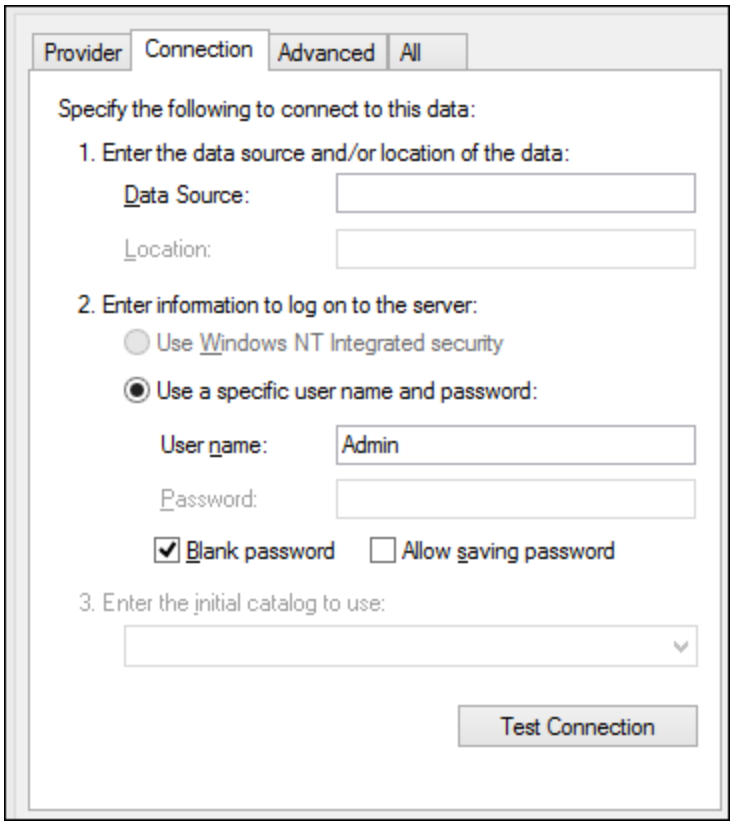
Para conectarse a una base de datos Microsoft Access protegida con contraseña:

1. Inicie el asistente para la conexión de base de datos[953].
2. Haga clic en **Conexiones ADO**.
3. Ahora haga clic en **Generar**.

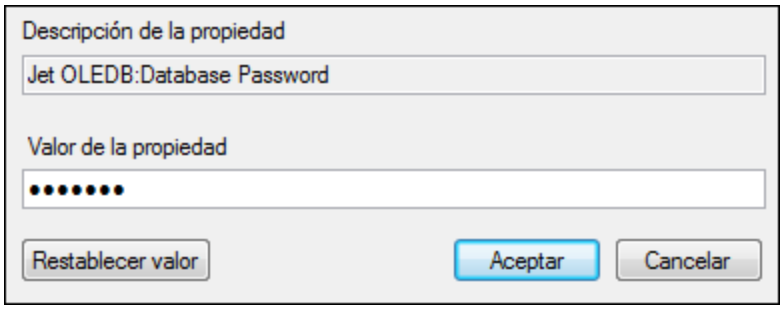


4. Seleccione **Microsoft Office 15.0 Access Database Engine OLE DB Provider** y haga clic en **Siguiente**.

 © 2016-2022 Altova GmbH



5. En el cuadro de texto Origen de datos introduzca la ruta de acceso del archivo de Microsoft Access en formato UNC, por ejemplo, `\\myserver\\mynetworkshare\Reports\Revenue.accdb`, donde `myserver` es el nombre del servidor y `mynetworkshare` el nombre del recurso compartido de red.
6. En la pestaña *Todos* haga doble clic en la propiedad **Jet OLEDB:Database Password** e introduzca la contraseña de la base de datos.



Nota: si no es capaz de establecer la conexión, busque el archivo de información del grupo de trabajo (`System.MDW`) de su perfil de usuario y establezca el valor de la propiedad **Jet OLEDB: System database** en la ruta de acceso del archivo `System.MDW`.

19.1.10.10 Microsoft SQL Server (ADO)

En este ejemplo explicamos cómo conectarse a una base de datos SQL Server con ADO. Estas instrucciones asumen que usa el driver recomendado para SQL Server **Microsoft OLE DB Driver for SQL Server**

(MSOLEDBSQL), que puede descargar en https://docs.microsoft.com/en-us/sql/connect/oledb/download-oledb-driver-for-sql-server?view=sql-server-ver15.

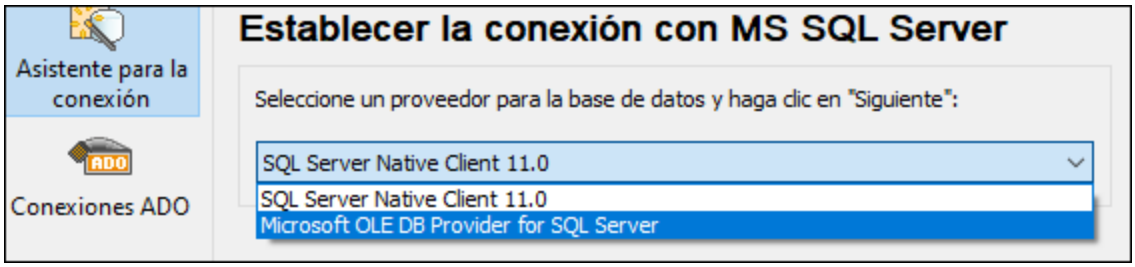
> Antes se seguir estas instrucciones, asegúrese de que descarga e instala el proveedor que mencionado más arriba en su equipo de trabajo. El proveedor ADO debe ser de la misma versión de plataforma que XMLSpy (32 bits o 64 bits).

Si quiere usar otro proveedor ADO, como **QL Server Native Client (SQLNCLI)** or **Microsoft OLE DB Provider for SQL Server (SQLOLEDB)**, las instrucciones son parecidas; sin embargo, estos proveedores están obsoletos, por lo que no se recomiendan. Además, para poder conectarse correctamente con un proveedor obsoleto puede que necesite configurar otras propiedades de conexión, como se describe en Configurar las propiedades de vínculo de datos de SQL Server 961.

> Es un problema conocido que el proveedor de BD para SQL Server **Microsoft OLE DB Provider for SQL Server (SQLOLEDB)** tiene dificultades para enlazar parámetros de consultas complejas como las expresiones comunes de tabla (CTE) e instrucciones SELECT anidadas.

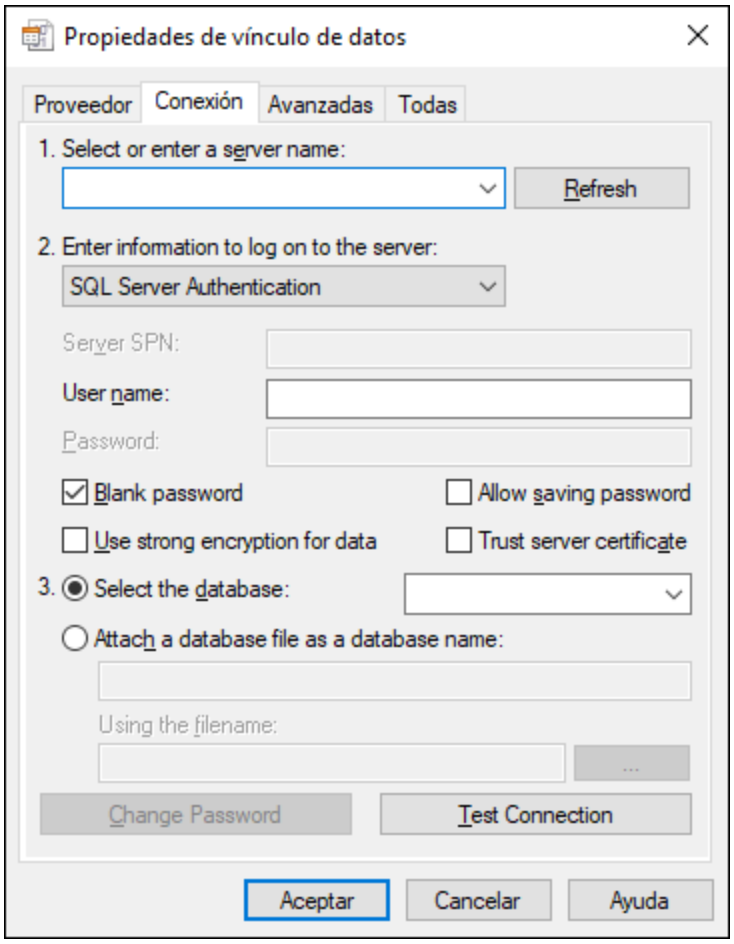
Para conectarse a SQL Server:

1. Inicie el asistente de conexión 953.
2. Seleccione **Microsoft SQL Server (ADO)** y después haga clic en **Siguiente**. Verá que aparece la lista de proveedores ADO disponibles. En este ejemplo usaremos **Microsoft OLE DB Driver for SQL Server**. Si no está en la lista, asegúrese de que está instalado en su equipo, como hemos mencionado antes.

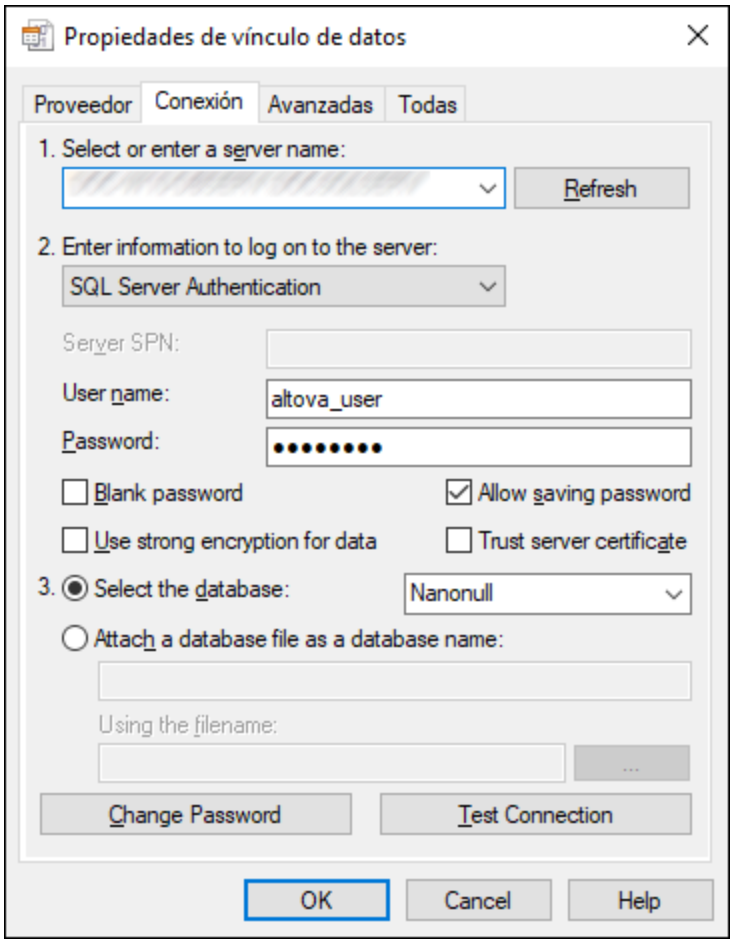


3. Haga clic en **Siguiente**. Se abre el cuadro de diálogo "Propiedades de enlace de datos".

© 2016-2022 Altova GmbH



4. Seleccione o introduzca el nombre del servidor de BD, por ejemplo **SQLSERV01**. Si se conecta a una instancia de SQL Server con nombre, el nombre del servidor se parecerá a: **SQLSERV01\SOMEINSTANCE**.
5. Si configuró el servidor de BD para que permita conexiones de usuarios autenticados en el dominio Windows, seleccione **Autenticación Windows**. De lo contrario seleccione **Autenticación SQL Server**, desmarque la casilla *Contraseña en blanco* e introduzca las credenciales de BD en los campos correspondientes.
6. Marque la casilla *Permitir guardar contraseña* y seleccione la BD a la que se quiere conectar (en este ejemplo, "Nanonull").



7. Para comprobar la conexión haga clic en **Comprobar conexión**. Este paso es opcional, pero recomendamos no saltárselo.
8. Haga clic en **Aceptar**.

19.1.10.11 Microsoft SQL Server (ODBC)

En este ejemplo aprenderá a conectar una base de datos a SQL Server con ODBC.

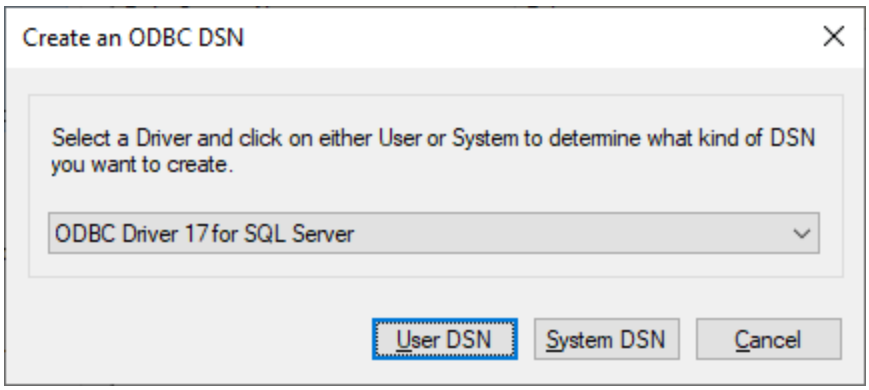
Requisitos previos:

- Descargue e instale el controlador **Microsoft ODBC Driver for SQL Server**, que encontrará en el sitio web de Microsoft (*véase* *https://docs.microsoft.com/en-us/SQL/connect/odbc/download-odbc-driver-for-sql-server*).Este ejemplo usa **Microsoft ODBC Driver 17 for SQL Server** para conectarse a la base de datos **SQL Server 2016**. Puede descargar un controlador distinto en función de la versión de SQL Server a la que se quiera conectar. Para información sobre las versiones del controlador ODBC compatibles con su base de datos SQL Server consulte los requisitos del sistema del controlador.

© 2016-2022 Altova GmbH

Para conectarse a SQL Server con ODBC:

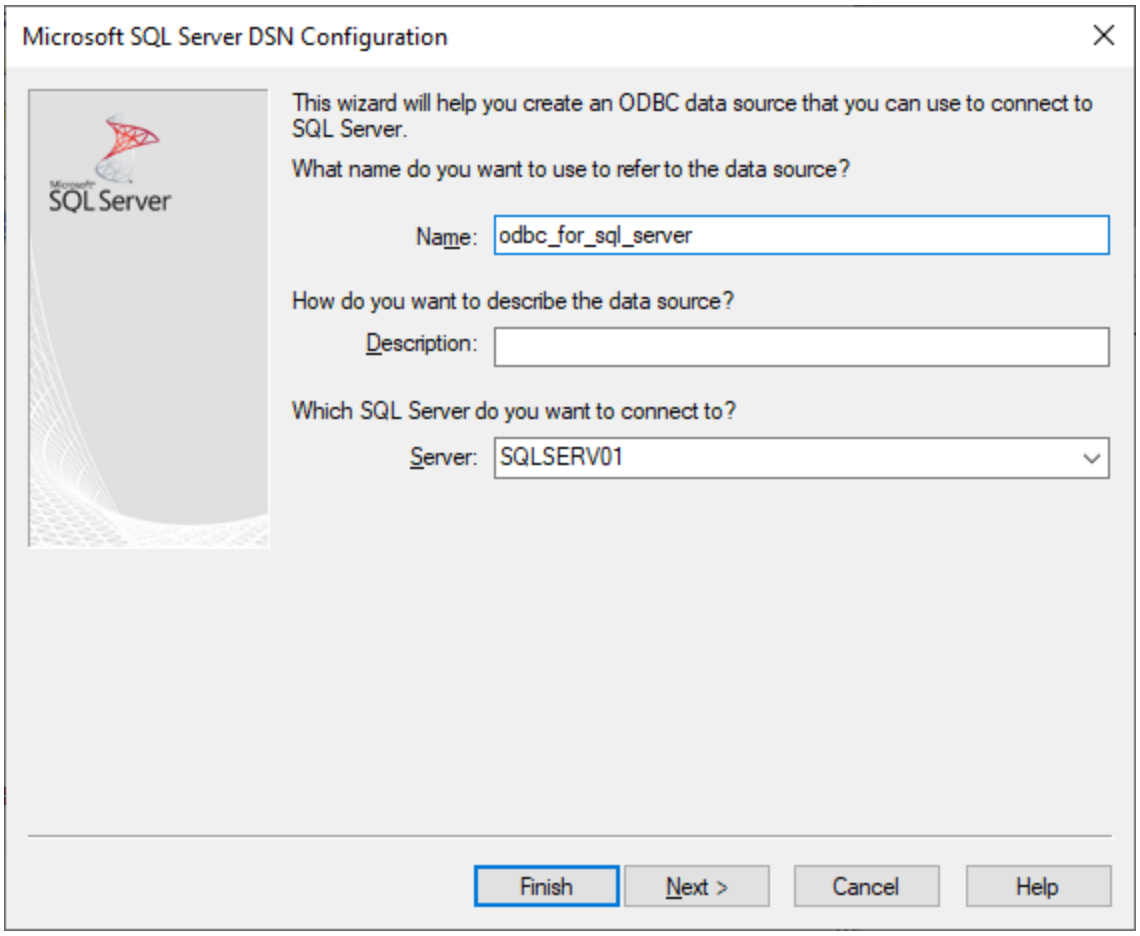
1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en **Conexiones ODBC**.
3. Seleccione **DSN de usuario** (o **DSN de sistema** si tiene privilegios de administrador) y haga clic en **Agregar** .
4. Seleccione el controlador de la lista. Tenga en cuenta que este solo aparece en esta lista una vez se ha instalado.



5. Haga clic en **DSN de usuario** (o en **DSN de sistema** si está creando un sistema DNS).

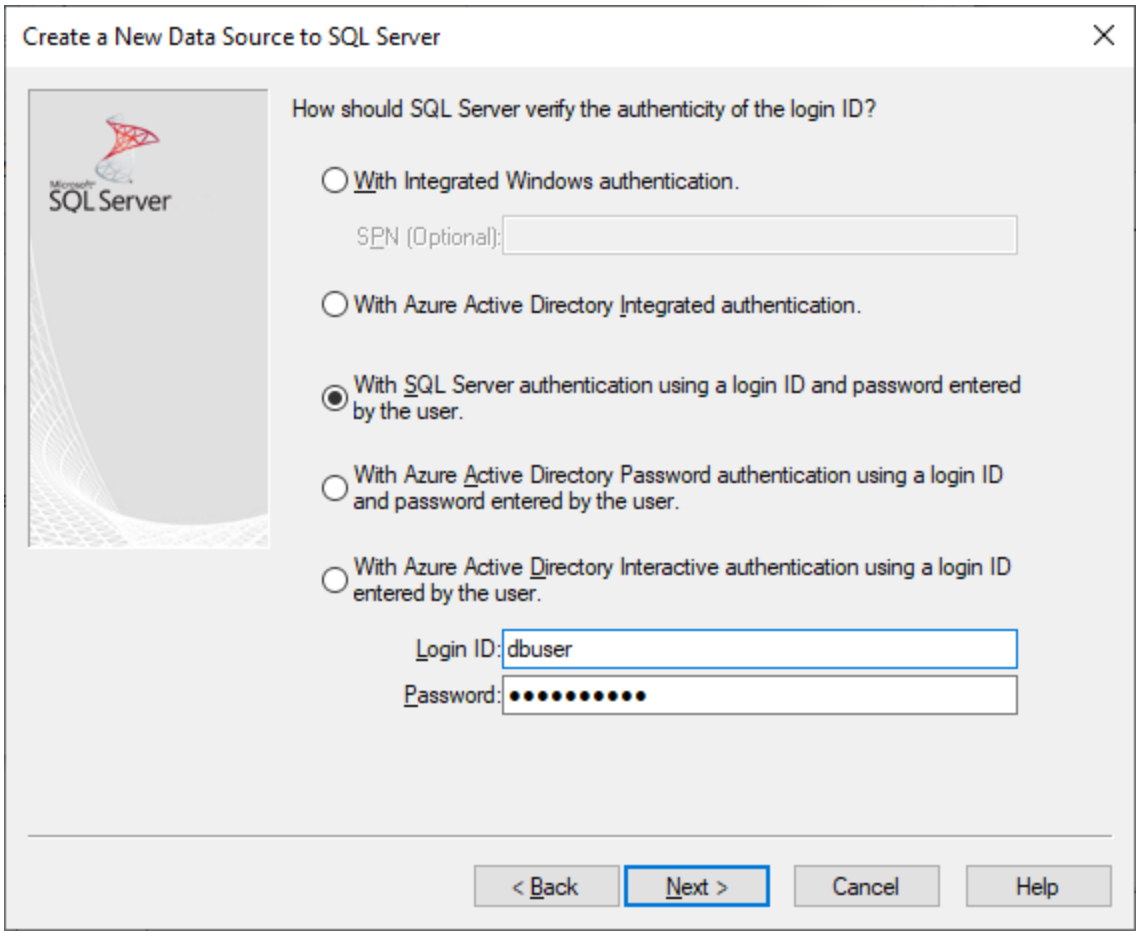
> Para crear un **Sistema DSN** necesita ejecutar XMLSpy como administrador. Si lo que quiere es crear este sistema, salga del asistente y vuelva a ejecutar XMLSpy como administrador y siga los pasos desde el principio.

6. Introduzca un nombre y, si quiere, una descripción que identifique esta conexión. A continuación seleccione en la lista el SQL Server al que se quiere conectar (en este ejemplo, **SQLSERV01**).

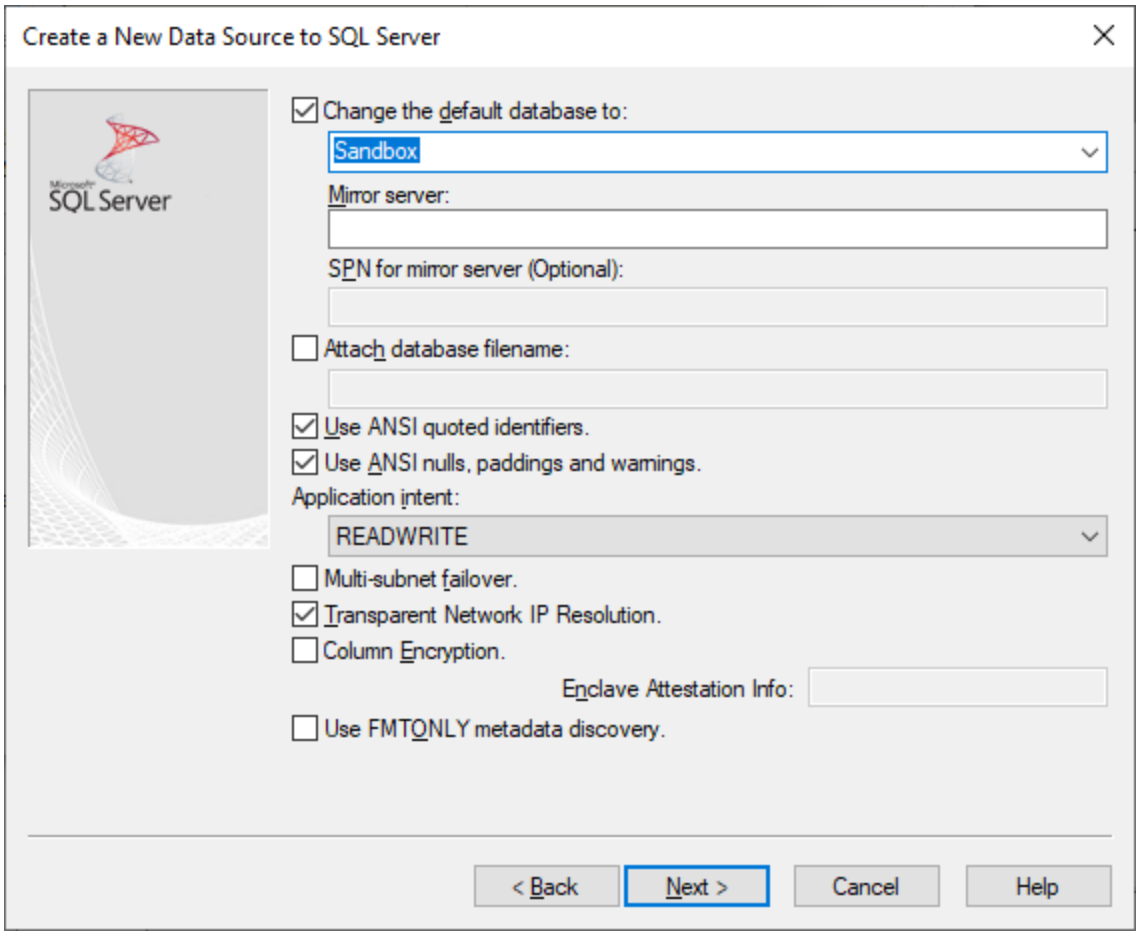


7. Si el servidor de BD se configuró para que permita conexiones de usuarios autenticados en el dominio Windows, seleccione **Con autenticación integrada de Windows**. De lo contrario, seleccione la opción que necesite. En este ejemplo usamos **Con autenticación de SQL Server...**, que necesita que se introduzcan el nombre de usuario y la contraseña en los campos correspondientes.

© 2016-2022 Altova GmbH

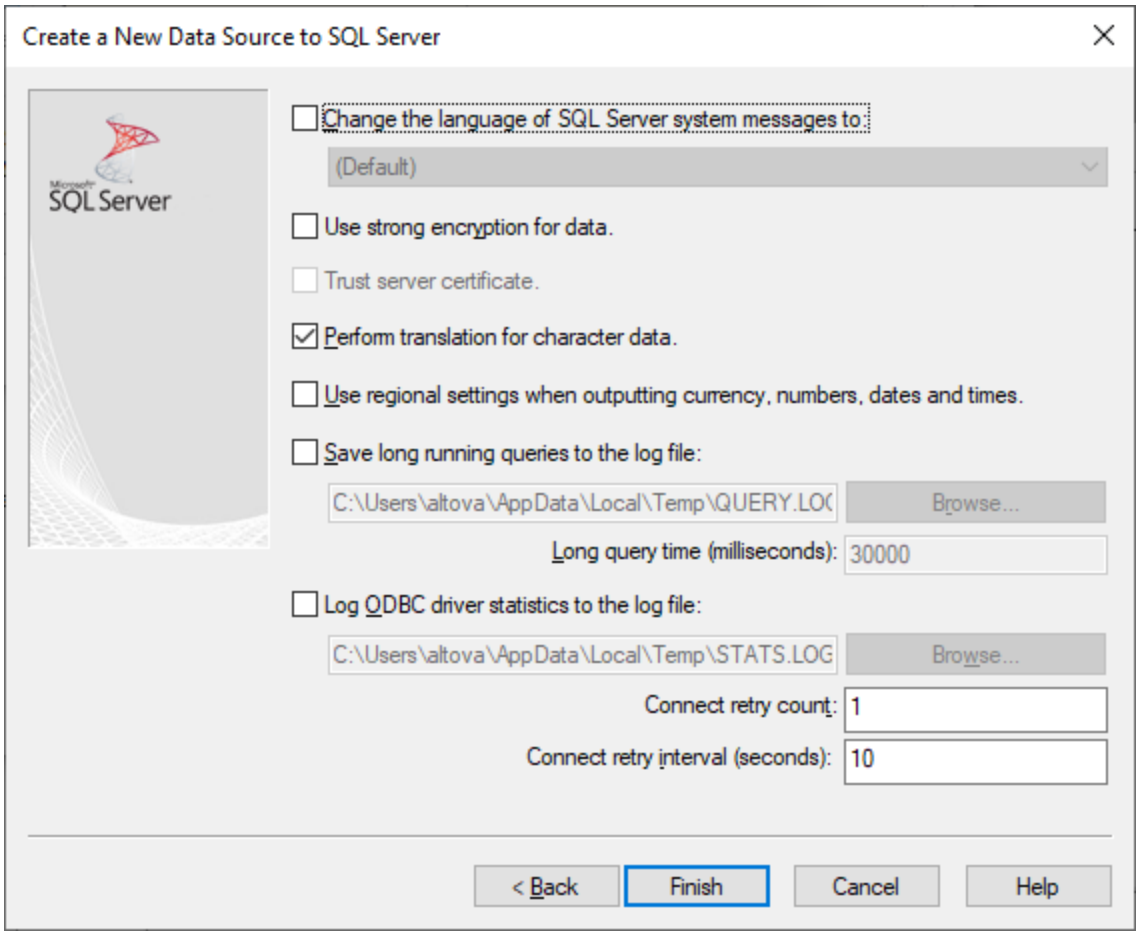


8. También puede marcar la casilla **Cambiar la base de datos predeterminada a** e introducir el nombre de la base de datos a la que se va a conectar (en este ejemplo, **Sandbox**).

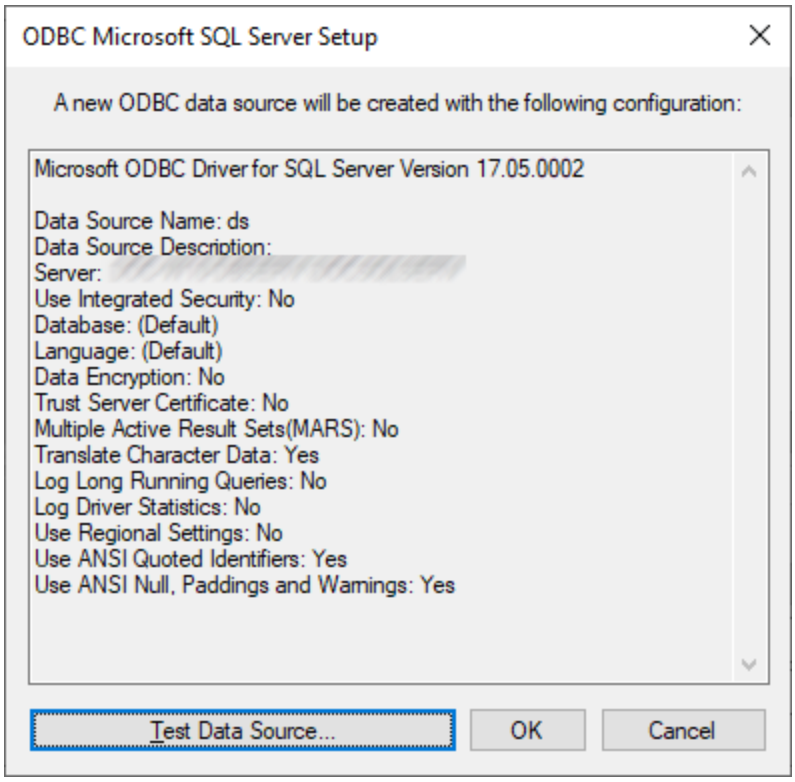


9. Haga clic en **Siguiente** y, si quiere, configure el resto de los parámetros de esta conexión.

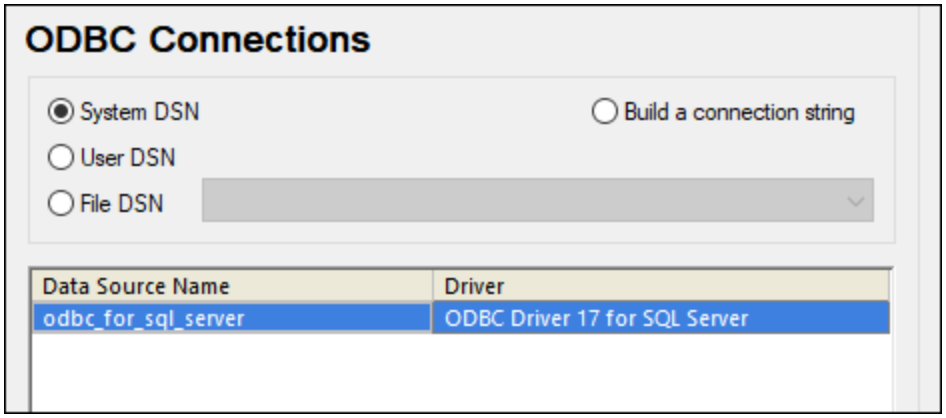
© 2016-2022 Altova GmbH



10. Haga clic en **Finalizar**. Aparece un cuadro de diálogo de conformación que contiene los detalles de la conexión.



11. Haga clic en **Aceptar**. Ahora la fuente de datos aparece en la lista de fuentes de datos de **Usuario** o **Sistema**, según la configuración; por ejemplo:



19.1.10.12 MySQL (ODBC)

Este ejemplo explica cómo conectarse a un servidor de base de datos MySQL desde un equipo Windows con el controlador ODBC. El controlador ODBC MySQL no está disponible en Windows así que deberá descargarlo e instalarlo por separado. En este ejemplo usamos el conector MySQL/ODBC 8.0.

© 2016-2022 Altova GmbH

Requisitos previos:

- El controlador MySQL ODBC está instalado en el sistema. Consulte la documentación de MySQL para conocer qué controlador debe usar para su versión del servidor de base de datos (*véase https://dev.mysql.com/downloads/connector/odbc/*).
- Disponer de los datos de conexión: host, base de datos, puerto, nombre de usuario y contraseña.

Conector MySQL/ODBC 8.0
Si instala el controlador MySQL ODBC para plataformas de 64 bits, asegúrese de que también instala la versión de XMLSpy para plataformas de 64 bits.

Para conectarse a MySQL por ODBC:

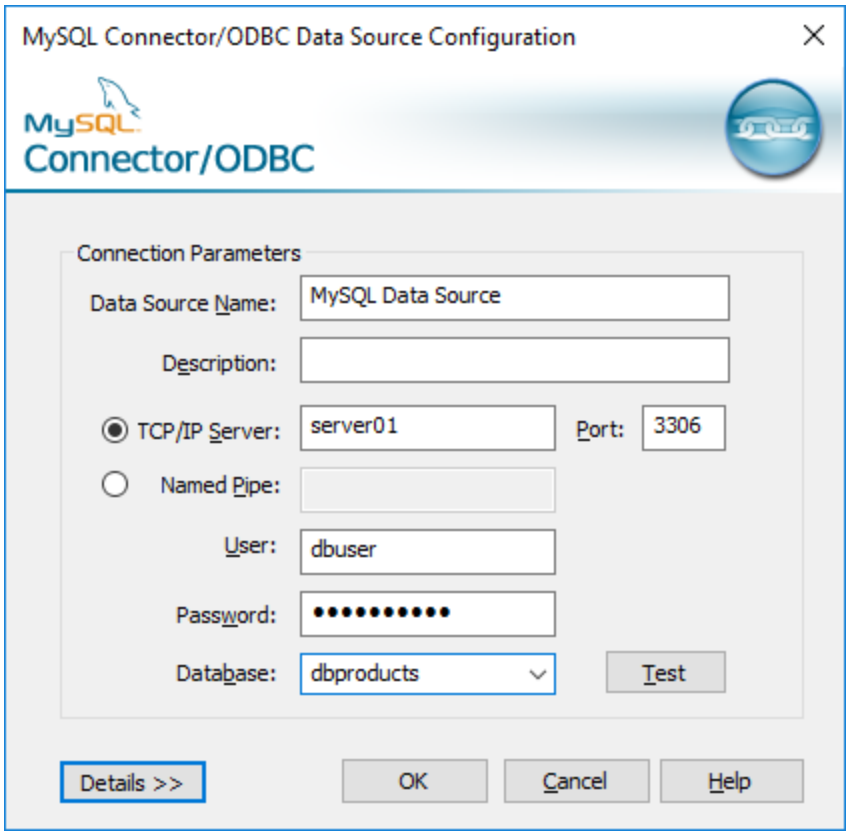
1. Inicie el asistente para la conexión de base de datos 953.
2. Seleccione la opción *MySQL (ODBC)* y después haga clic en **Siguiente**.



3. Seleccione la opción *Crear un DNS nuevo con el controlador* y seleccione un controlador MySQL. Si no hay ningún controlador MySQL en la lista, haga clic en **Editar controladores** y seleccione uno (la lista contiene todos los controladores ODBC que están instalados en el sistema).

   Si instaló la versión de XMLSpy para plataformas de 64 bits se mostrarán los controladores ODBC en la lista. De lo contrario se mostrarán los controladores para la versión de 32 bits. Consulte también el apartado Ver los controladores ODBC disponibles 973.

4. Haga clic en **Conectarse**.



5. En el cuadro de texto *Nombre del origen de datos* introduzca un nombre que le ayude a identificar este origen de datos ODBC más adelante.
6. Rellene las credenciales de la conexión de BD (servidor TCP/IP, usuario, contraseña), seleccione una base de datos y haga clic en **Aceptar**.

Nota: si el servidor de BD es remoto, el administrador del servidor debe configurarlo para que acepte conexiones remotas desde la dirección IP de su equipo. Además, si hace clic en **Detalles>>**, podrá configurar algunos parámetros más. Consulte la documentación del controlador antes de cambiar los valores predeterminados.

19.1.10.13 Oracle (JDBC)

Este ejemplo explica cómo conectarse a un servidor de base de datos Oracle desde un equipo cliente, usando la interfaz JDBC. La conexión se crea como una conexión Java pura, usando el paquete **Oracle Instant Client Package (Basic)** que se puede descargar del sitio web de Oracle. La ventaja de este tipo de conexión es que solamente exige el entorno Java y las bibliotecas .jar que vienen con el paquete Oracle Instant Client Package. Es decir, no es necesario instalar ni configurar clientes de base de datos más complejos.

Requisitos:

- JRE (Java Runtime Environment) o Java Development Kit (JDK) está instalado. Este último puede ser el JDK de Oracle o uno de código abierto, como Oracle OpenJDK. XMLSpy identifica la ruta de acceso

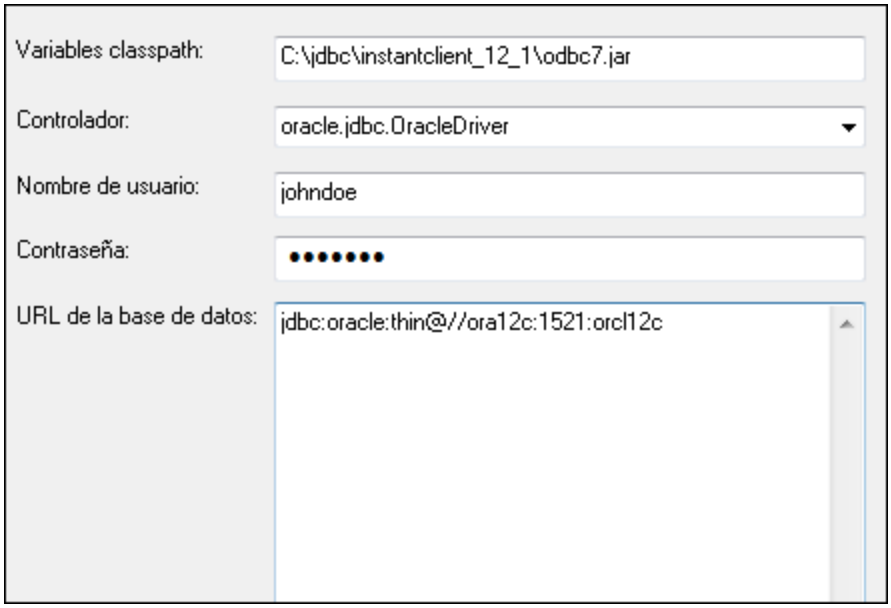
 © 2016-2022 Altova GmbH

al equipo virtual Java (JVM) a partir de estas ubicaciones, en este orden: a) la ruta personal de acceso al JVM que puede indicar en en las **Opciones** ; b) la ruta de acceso al JVM que se encuentra en el registro; c) la variable de entorno `JAVA_HOME`.

- Asegúrese de que la plataforma de XMLSpy (32 o 64 bits) es la misma que la de JRE/JDK.
- El paquete **Oracle Instant Client Package (Basic)** está disponible en el sistema operativo. El paquete se puede descargar del sitio web oficial de Oracle. En este ejemplo utilizamos la versión 12.1.0.2.0 para Windows de 32 bits y, por tanto, Oracle JDK de 32 bits.
- Disponer de los datos de conexión: host, puerto, nombre del servicio, nombre de usuario y contraseña.

Para conectarse a Oracle a través del paquete Instant Client Package:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en **Conexiones JDBC**.
3. En el campo *Variables classpath* debe introducir la ruta de acceso del archivo .jar que ofrece conectividad con la base de datos. Si fuera necesario, también puede introducir una lista de rutas de archivo .jar separadas por caracteres de punto y coma. En este ejemplo, el archivo .jar está ubicado en esta ruta de acceso: `C:\jdbc\instantclient_12_1\ojdbc7.jar`. Tenga en cuenta que este campo puede dejarse en blanco si añadió la ruta de acceso de los archivos .jar a la variable de entorno CLASSPATH del sistema operativo (*véase Configurar la variable CLASSPATH* 977).
4. En el campo *Controlador* seleccione **oracle.jdbc.OracleDriver** o **oracle.jdbc.driver.OracleDriver**. Recuerde que esta entrada solo estará disponible si se encuentra una ruta de archivo .jar válida en el campo Variables *classpath* o en la variable de entorno CLASSPATH del sistema operativo.
5. Introduzca el nombre de usuario y contraseña de la base de datos.



6. Introduzca la cadena de conexión para el servidor de BD en el cuadro de texto *URL de la base de datos* (reemplace lo valores resaltados con los de su servidor de base de datos).

```
jdbc:oracle:thin:@//host:puerto:servicio
```

7. Haga clic en **Conectarse**.

19.1.10.14 Oracle (ODBC)

Este ejemplo explica cómo conectarse a un servidor de base de datos Oracle desde XMLSpy. En este ejemplo la base de datos Oracle está en un equipo de la red y la conexión se establece a través de un cliente de base de datos Oracle instalado en el equipo local.

En este ejemplo ofrecemos instrucciones para configurar un DSN ODBC con el asistente para la conexión de base de datos de XMLSpy. Si ya tiene un DSN o prefiere crear uno desde el administrador de orígenes de datos ODBC de Windows, puede seleccionarlo desde el asistente. Para más información consulte el apartado Conexiones ODBC 971.

Requisitos:

- El cliente de BD Oracle (que incluye el controlador ODBC Oracle) está instalado y configurado en el sistema. Para más información consulte la documentación del software de Oracle.
- El archivo `tnsnames.ora` ubicado en el directorio de inicio de Oracle contiene una entrada que describe los parámetros de conexión de la base de datos:

```
ORCL =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = server01)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = orcl)
      (SERVER = DEDICATED)
    )
  )
```

La ruta de acceso del archivo `tnsnames.ora` depende de la ubicación del directorio de inicio de Oracle. Por ejemplo, en el caso del cliente de base de datos Oracle 11.2.0, la ruta de acceso predeterminada del directorio de inicio podría ser:

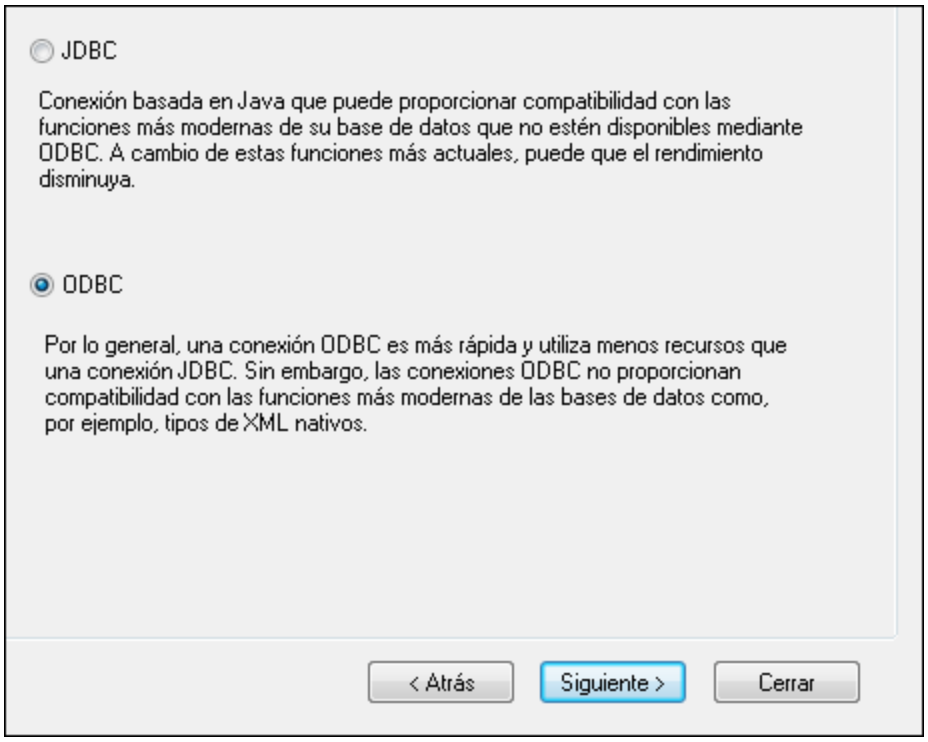
```
C:\app\nombreUsuario\product\11.2.0\client_1\network\admin\tnsnames.ora
```

En el archivo `tnsnames.ora` puede introducir entradas nuevas, pegando los datos de conexión y guardando el archivo o ejecutando el asistente *Net Configuration Assistant* de Oracle (si está disponible). Si quiere que estos valores aparezcan en las listas desplegables durante el proceso de configuración, puede que necesite añadir la ruta de acceso a la carpeta de administrador como una variable de entorno **TNS_ADMIN**.

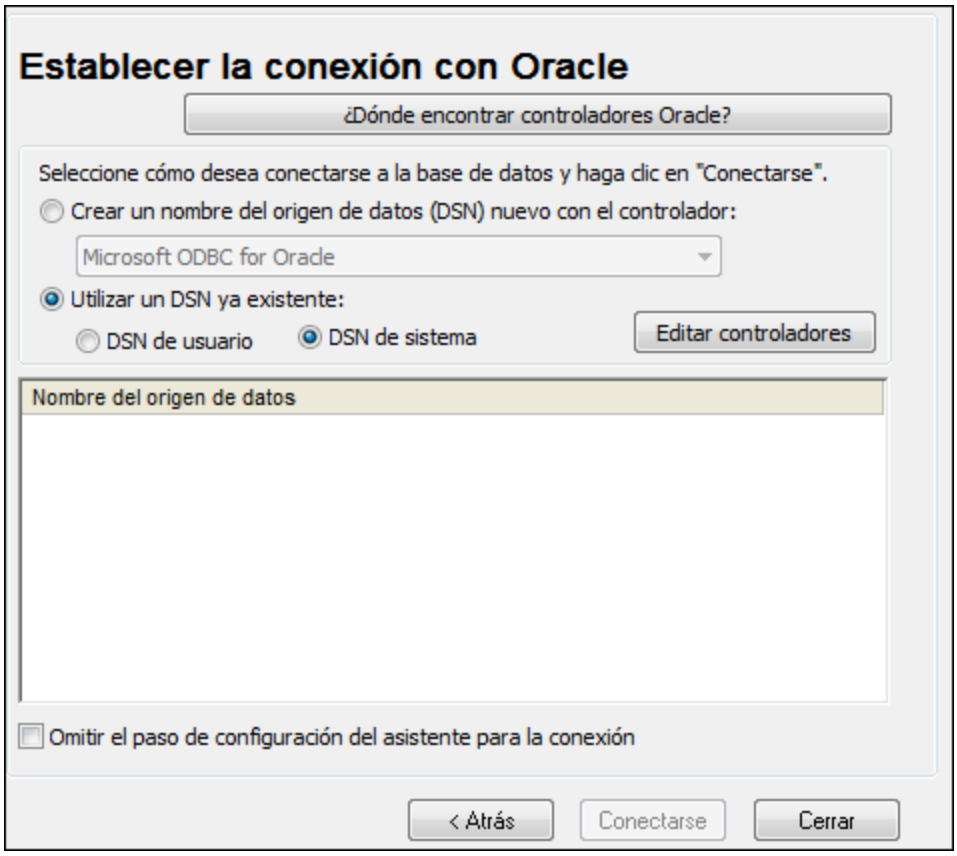
Para conectarse a Oracle por ODBC:

1. Inicie el asistente para la conexión de base de datos 953.
2. Seleccione la opción *Oracle (ODBC / JDBC)* y después haga clic en **Siguiente**.

© 2016-2022 Altova GmbH

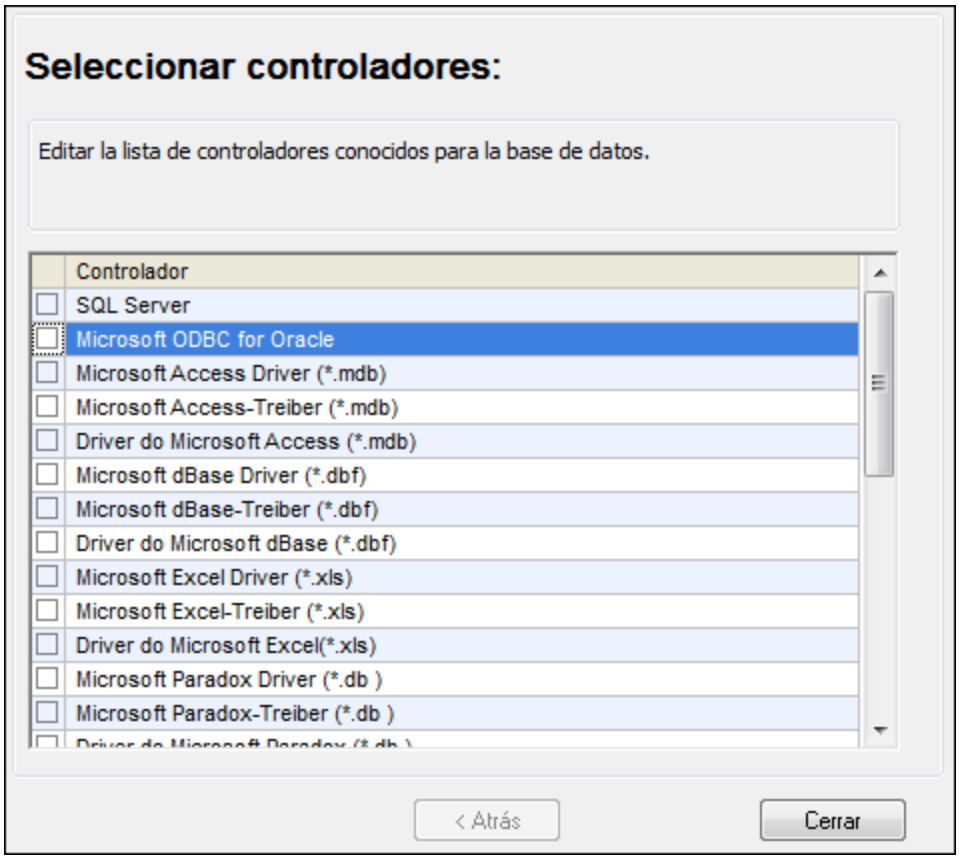


3. Seleccione el botón de opción *ODBC*.

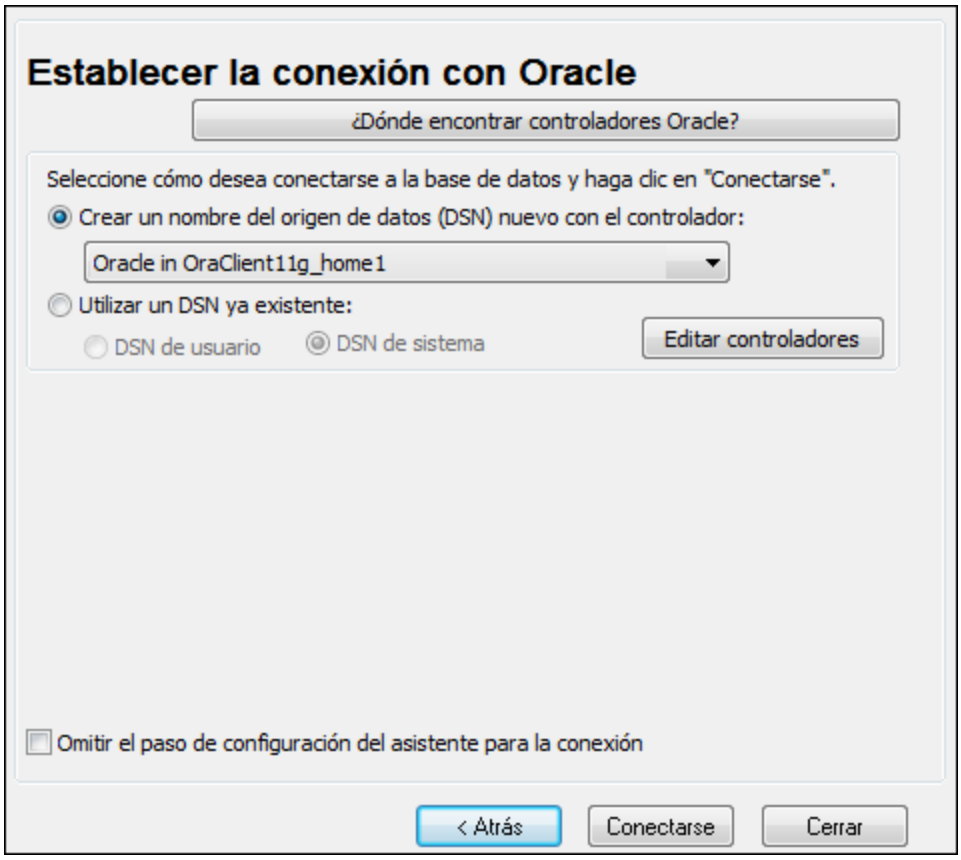


4. Haga clic en **Editar controladores**.

© 2016-2022 Altova GmbH



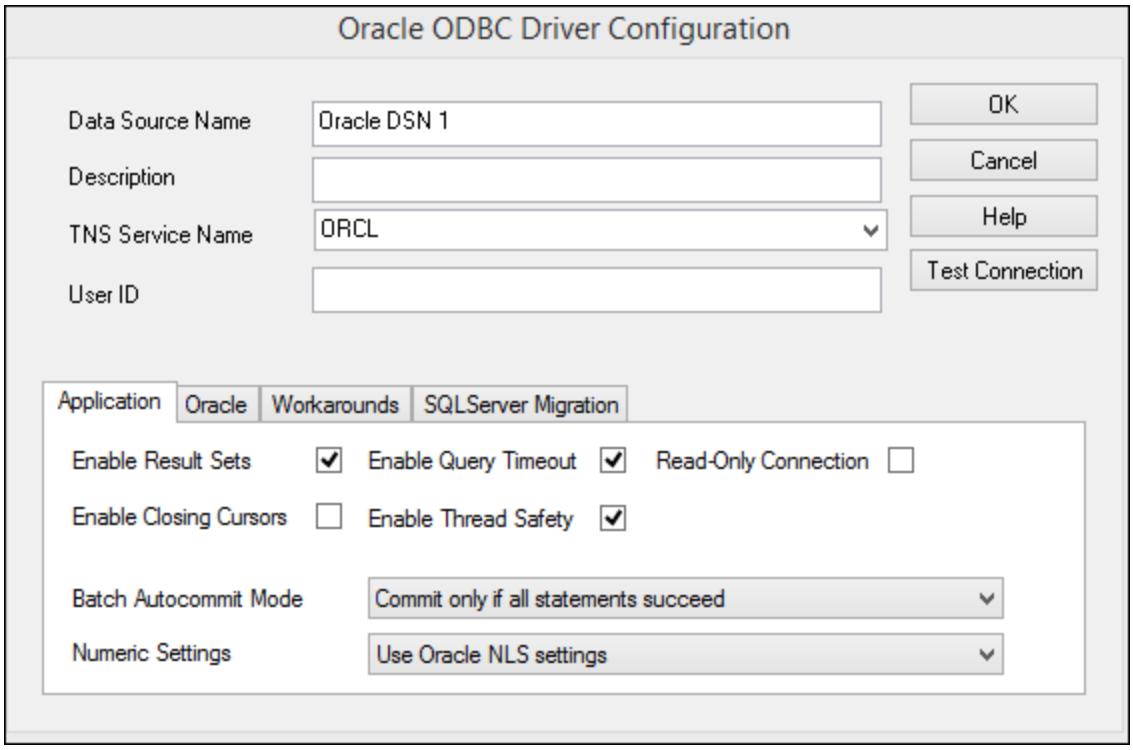
5. Seleccione los controladores Oracle que desea usar (en este ejemplo usamos **Oracle in OraClient11g_home1**). La lista incluye todos los controladores Oracle que están disponibles en el sistema después de instalar el cliente Oracle.
6. Haga clic en **Atrás**.
7. Seleccione la opción *Crear un DSN nuevo con el controlador* y después seleccione el controlador de Oracle elegido en el paso nº 4.



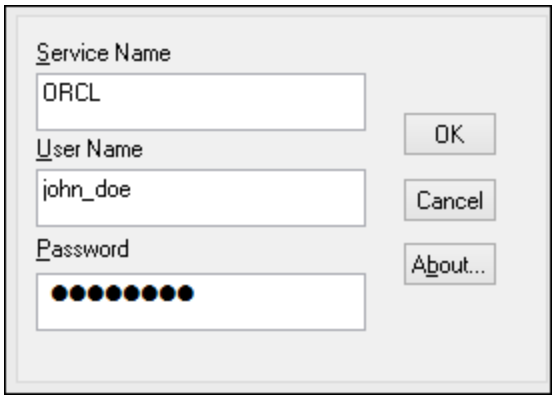
Recomendamos no utilizar el controlador **Microsoft ODBC for Oracle**. Microsoft recomienda utilizar el controlador ODBC que ofrece Oracle (consulte http://msdn.microsoft.com/en-us/library/ms714756%28v=vs.85%29.aspx)

8. Haga clic en **Conectarse**.

© 2016-2022 Altova GmbH



9. En el cuadro de texto *Nombre del origen de datos* introduzca un nombre que le ayude a identificar el origen de datos más adelante (en este ejemplo **Oracle DSN 1**).
10. En el cuadro de texto *Nombre del servicio TNS* introduzca el nombre de la conexión tal y como se define en el archivo `tnsnames.ora` (ver Requisitos[1020]). En este ejemplo el nombre de la conexión es **ORCL**.
11. Haga clic en **Aceptar**.



12. Escriba el nombre de usuario y la contraseña de la base de datos y haga clic en **Aceptar** para terminar.

19.1.10.15 PostgreSQL (ODBC)

Este ejemplo explica cómo conectarse a un servidor de base de datos PostgreSQL desde un equipo Windows con el controlador ODBC. El controlador ODBC PostgreSQL no está disponible en Windows así que deberá descargarlo e instalarlo por separado. En este ejemplo usamos la versión 11.0 del controlador psqlODBC, que se puede descargar del sitio web oficial (consulte el apartado Resumen de controladores de base de datos 955).

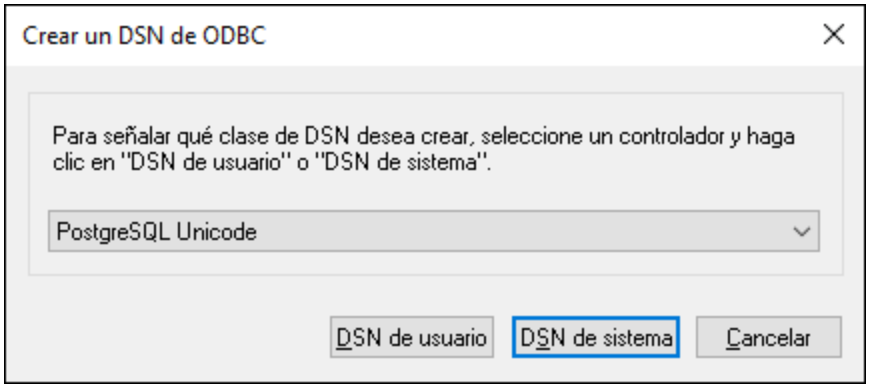
Nota: también se puede conectar a un servidor de base de datos PostgreSQL directamente (es decir, sin el controlador ODBC). Consulte el apartado Conexiones PostgreSQL 979 para más información.

Requisitos:

- El controlador *psqlODBC* está instalado en el sistema.
- Disponer de los datos de conexión: servidor, puerto, base de datos, nombre de usuario y contraseña.

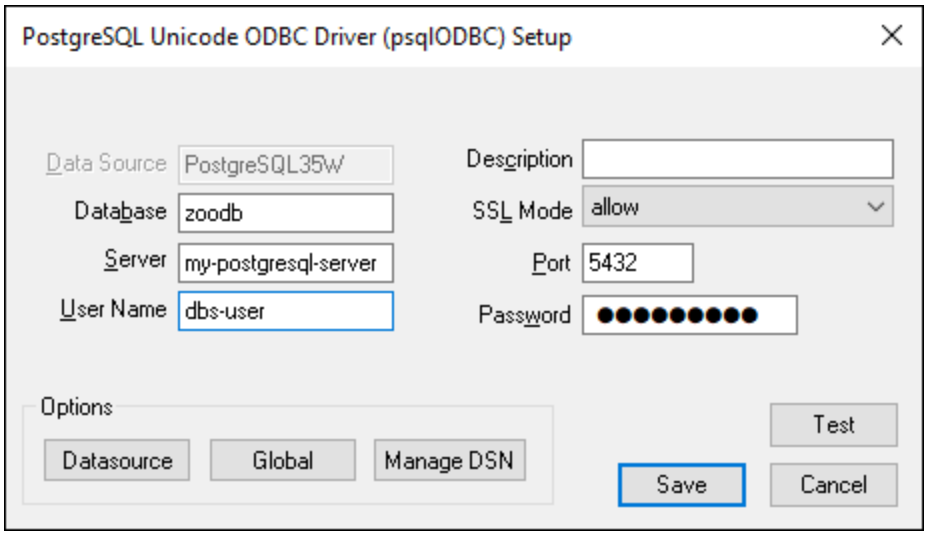
Para conectarse a PostgreSQL por ODBC:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en **Conexiones ODBC**.
3. Seleccione la opción *DNS de usuario*.
4. Haga clic en **Crear DNS nuevo** y seleccione el controlador en la lista desplegable. Si no hay ningún controlador PostgreSQL en la lista, asegúrese de que el controlador PostgreSQL está instalado en su sistema operativo, tal y como se indica más arriba en los requisitos.



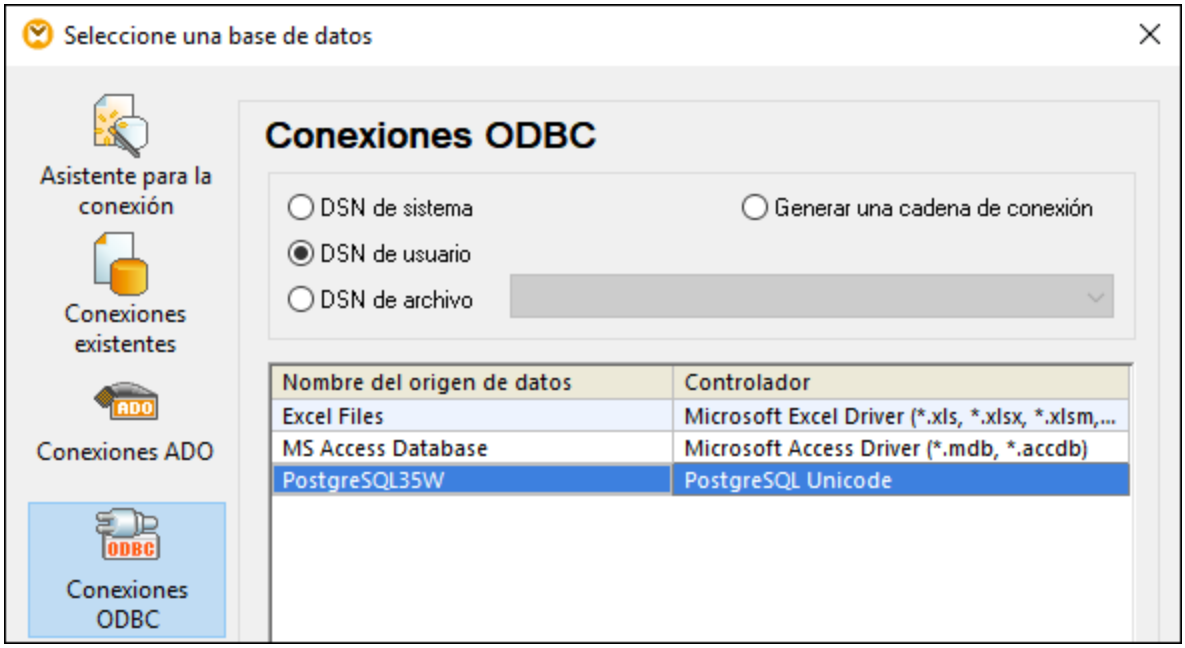
5. Haga clic en **DNS de usuario**.

 © 2016-2022 Altova GmbH



6. Rellene las credenciales de la conexión de base de datos (las debe proporcionar el propietario de la base de datos) y haga clic en **Aceptar**.

Ahora la conexión está disponible en la lista de conexiones ODBC. Para conectarse a la base de datos puede hacer doble clic en la conexión o seleccionarla y hacer clic en **Conectarse**.



19.1.10.16 Progress OpenEdge (JDBC)

En este apartado encontrará instrucciones para conectarse a un servidor de base de datos Progress OpenEdge 11.6 por JDBC.

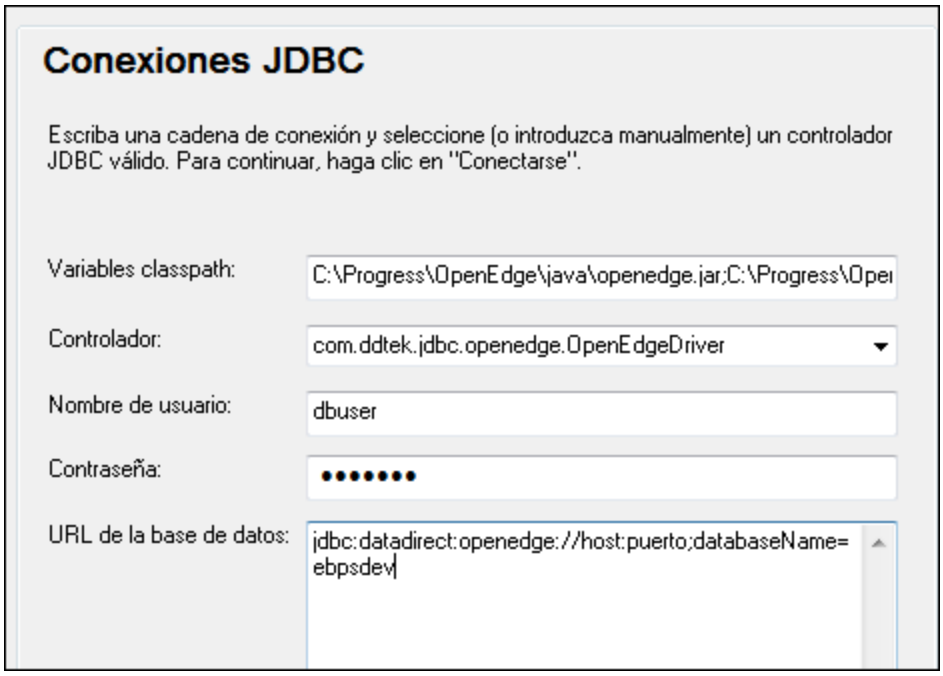
Requisitos:

- Debe tener instalado Java Runtime Environment (JRE) o Java Development Kit (JDK). Este último puede ser el JDK de Oracle o uno de código abierto, como Oracle OpenJDK. XMLSpy identifica la ruta de acceso al equipo virtual Java (JVM) a partir de estas ubicaciones, en este orden: a) la ruta personal de acceso al JVM que puede indicar en en las **Opciones** ; b) la ruta de acceso al JVM que se encuentra en el registro; c) la variable de entorno `JAVA_HOME`.
- Asegúrese de que la plataforma de XMLSpy (32 o 64 bits) es la misma que la de JRE/JDK.
- La variable de entorno `PATH` del sistema incluye la ruta al directorio `bin` del directorio de instalación de JRE o JDK (p. ej. `C:\Archivos de programa (x86)\Java\jre1.8.0_51\bin`).
- El controlador JDBC Progress OpenEdge está disponible en el sistema operativo. En este ejemplo la conexión por JDBC se consigue con los archivos de controlador **openedge.jar** y **pool.jar** disponibles en `C:\Progress\OpenEdge\java` y que se instalan con OpenEdge SDK.
- Disponer de los datos de conexión: host, puerto, nombre de la base de datos, nombre de usuario y contraseña.

Para conectarse a Progress OpenEdge por JDBC:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en **Conexiones JDBC**.
3. En el campo *Variables classpath* debe introducir la ruta de acceso del archivo .jar que ofrece conectividad con la base de datos. Si fuera necesario, también puede introducir una lista de rutas de archivo .jar separadas por caracteres de punto y coma. En este ejemplo, los archivos .jar están ubicados en esta ruta de acceso: **`C:\Progress\OpenEdge\java\openedge.jar;C:\Progress\OpenEdge\java\pool.jar;`**. Tenga en cuenta que este campo puede dejarse en blanco si añadió la ruta de acceso de los archivos .jar a la variable de entorno CLASSPATH del sistema operativo (*véase Configurar la variable CLASSPATH* 977).
4. En el campo *Controlador* seleccione **com.ddtek.jdbc.openedge.OpenEdgeDriver**. Recuerde que esta entrada solo estará disponible si se encuentra una ruta de archivo .jar válida en el campo Variables *classpath* o en la variable de entorno CLASSPATH del sistema operativo.

© 2016-2022 Altova GmbH



5. Introduzca el nombre de usuario y la contraseña de la base de datos.
6. Introduzca la cadena de conexión para el servidor de BD en el cuadro de texto *URL de la base de datos* (reemplace lo valores resaltados con los de su base de datos).

```
jdbc:datadirect:openedge://host:puerto;databaseName=nombre_BD
```

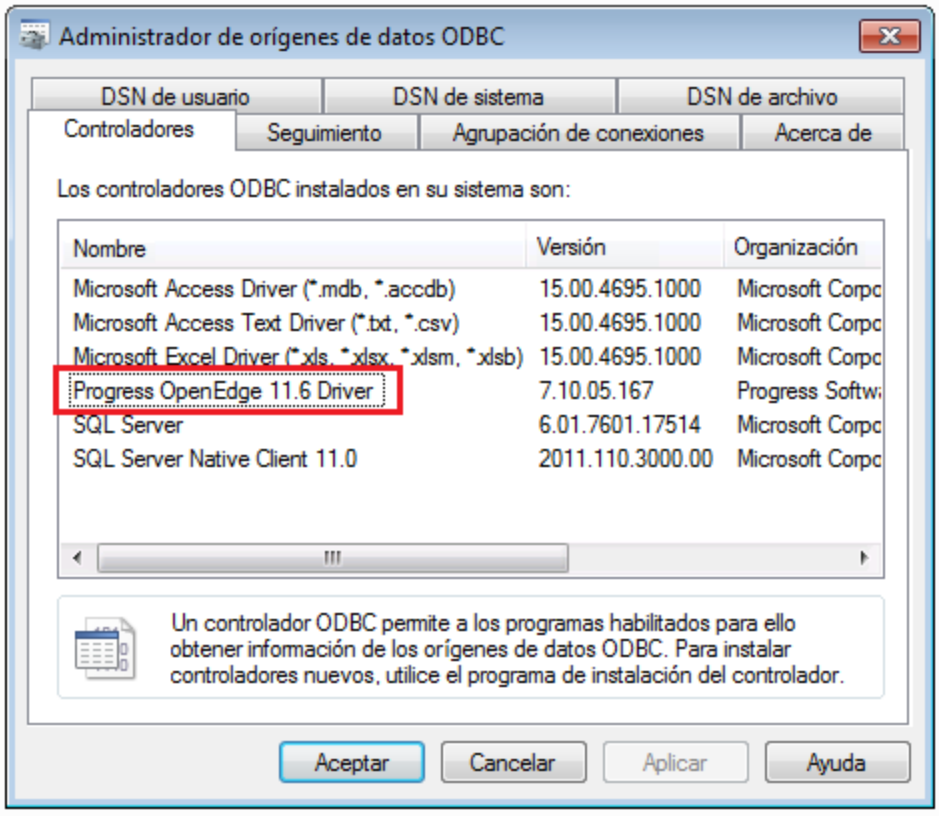
7. Haga clic en **Conectarse**.

19.1.10.17 Progress OpenEdge (ODBC)

En este apartado encontrará instrucciones para conectarse a un servidor de base de datos Progress OpenEdge por medio del controlador ODBC Progress OpenEdge 11.6.

Requisitos:

- El controlador *ODBC Connector for Progress OpenEdge* está instalado en el sistema operativo. Este controlador se puede descargar del sitio web del proveedor (consulte la lista del apartado Resumen de controladores de base de datos 955). Si trabaja con la versión de 32 bits de XMLSpy, descargue el controlador de 32 bits. Por el contrario, si usa la versión de 64 bits, descargue el controlador de 64 bits. Una vez finalizada la instalación, compruebe que el controlador ODBC está disponible en el equipo (*véase* *Ver los controladores ODBC disponibles* 973).



- Disponer de los datos de conexión: nombre de host, número de puerto, nombre de la base de datos, id. de usuario y contraseña.

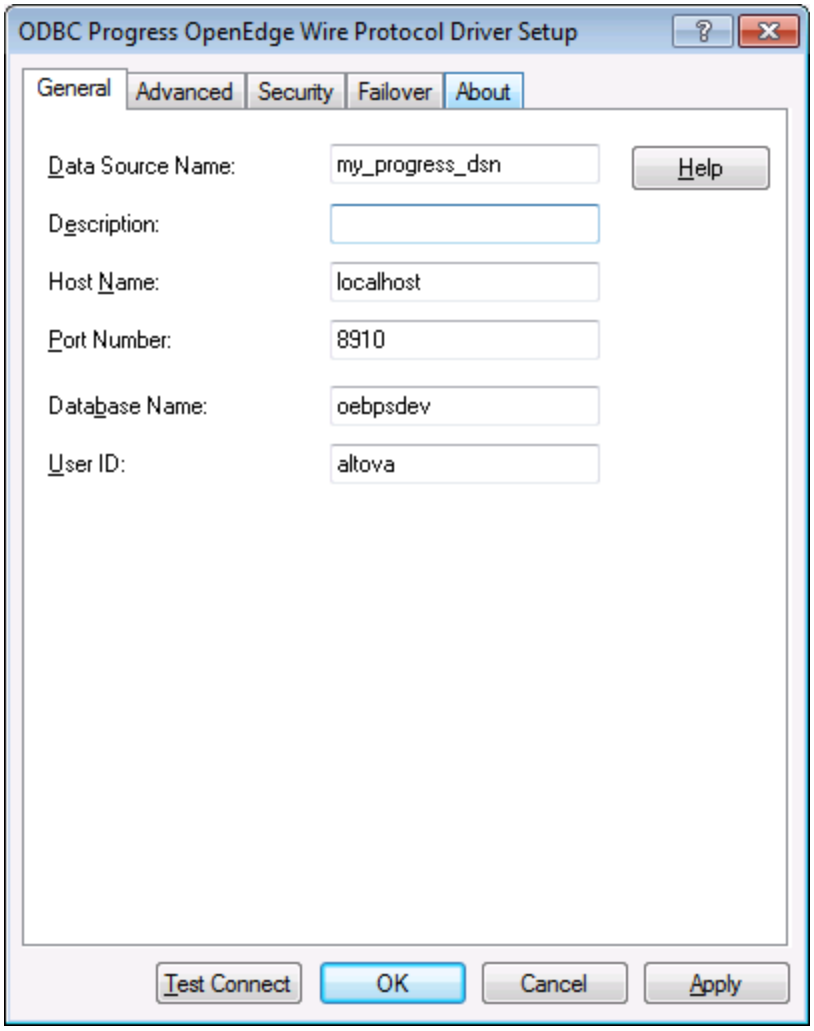
Para conectarse a Progress OpenEdge por ODBC:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en el botón Conexiones ODBC.
3. Haga clic en *DSN de usuario* (o en *DSN de sistema* o *DSN de archivo*).
4. Ahora haga clic en el icono **Agregar** .
5. Seleccione el controlador **Progress OpenEdge** de la lista y haga clic en el botón **DSN de usuario** (o **DSN de sistema** según corresponda).

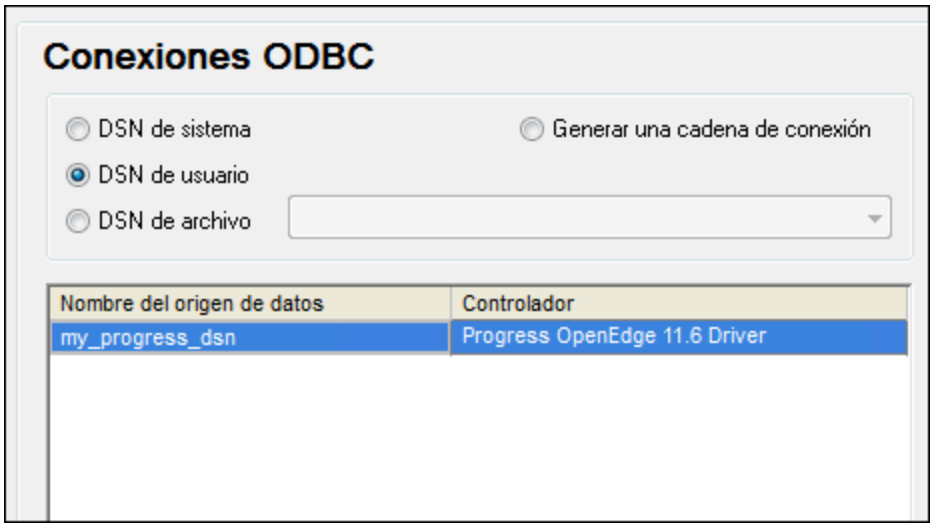


 © 2016-2022 Altova GmbH

6. Rellene las credenciales para la conexión con la base de datos (base de datos, servidor, puerto, nombre de usuario y contraseña) y haga clic en **Aceptar**. Para probar la conexión antes de guardar los datos introducidos haga clic en el botón **Probar conexión**.



7. Haga clic en **Aceptar**. El origen de datos nuevo aparece ahora en la lista de orígenes de datos ODBC.



8. Para terminar haga clic en **Conectarse**.

19.1.10.18 Sybase (JDBC)

Este ejemplo explica cómo conectarse a un servidor de base de datos Sybase por JDBC.

Requisitos:

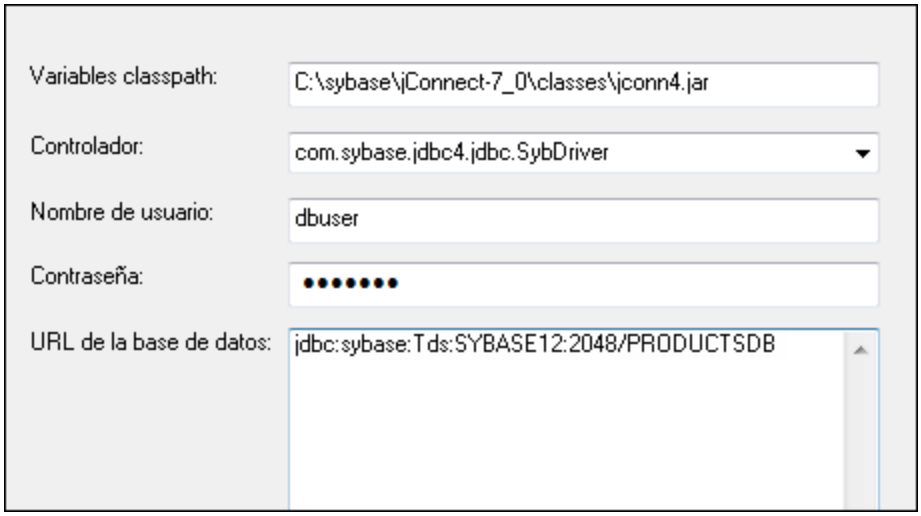
- Debe tener instalado Java Runtime Environment (JRE) o Java Development Kit (JDK). Este último puede ser el JDK de Oracle o uno de código abierto, como Oracle OpenJDK. XMLSpy identifica la ruta de acceso al equipo virtual Java (JVM) a partir de estas ubicaciones, en este orden: a) la ruta personal de acceso al JVM que puede indicar en en las **Opciones** ; b) la ruta de acceso al JVM que se encuentra en el registro; c) la variable de entorno `JAVA_HOME`.
- Asegúrese de que la plataforma de XMLSpy (32 o 64 bits) es la misma que la de JRE/JDK.
- El componente Sybase *jConnect* está instalado en el sistema (en este ejemplo usamos *jConnect 7.0* que se instala con el cliente *Sybase Adaptive Server Enterprise PC Client*. Consulte la documentación de Sybase para obtener más información.
- Disponer de los datos de conexión: host, puerto, nombre de la base de datos, nombre de usuario y contraseña.

Para conectarse a Sybase por JDBC:

1. Inicie el asistente para la conexión de base de datos 953.
2. Haga clic en **Conexiones JDBC**.
3. En el campo *Variables classpath* debe introducir la ruta de acceso del archivo .jar que ofrece conectividad con la base de datos. Si fuera necesario, también puede introducir una lista de rutas de archivo .jar separadas por caracteres de punto y coma. En este ejemplo, el archivo .jar está ubicado en esta ruta de acceso: `C:\sybase\jConnect-7_0\classes\jconn4.jar`. Tenga en cuenta que este campo puede dejarse en blanco si añadió la ruta de acceso de los archivos .jar a la variable de entorno CLASSPATH del sistema operativo (*véase Configurar la variable CLASSPATH* 977).

 © 2016-2022 Altova GmbH

4. En el campo *Controlador* seleccione **com.sybase.jdbc4.jdbc.SybDriver**. Recuerde que esta entrada solo estará disponible si se encuentra una ruta de archivo .jar válida en el campo Variables *classpath* o en la variable de entorno CLASSPATH del sistema operativo.



5. Introduzca el nombre de usuario y la contraseña de la base de datos.
6. Introduzca la cadena de conexión para el servidor de BD en el cuadro de texto *URL de la base de datos* (reemplace lo valores resaltados con los de su base de datos).

```
jdbc:sybase:Tds:nombreHost:puerto/nombreBD
```

7. Haga clic en **Conectarse**.

19.1.10.19 Teradata (JDBC)

Este ejemplo explica cómo conectarse a un servidor de base de datos Teradata por JDBC.

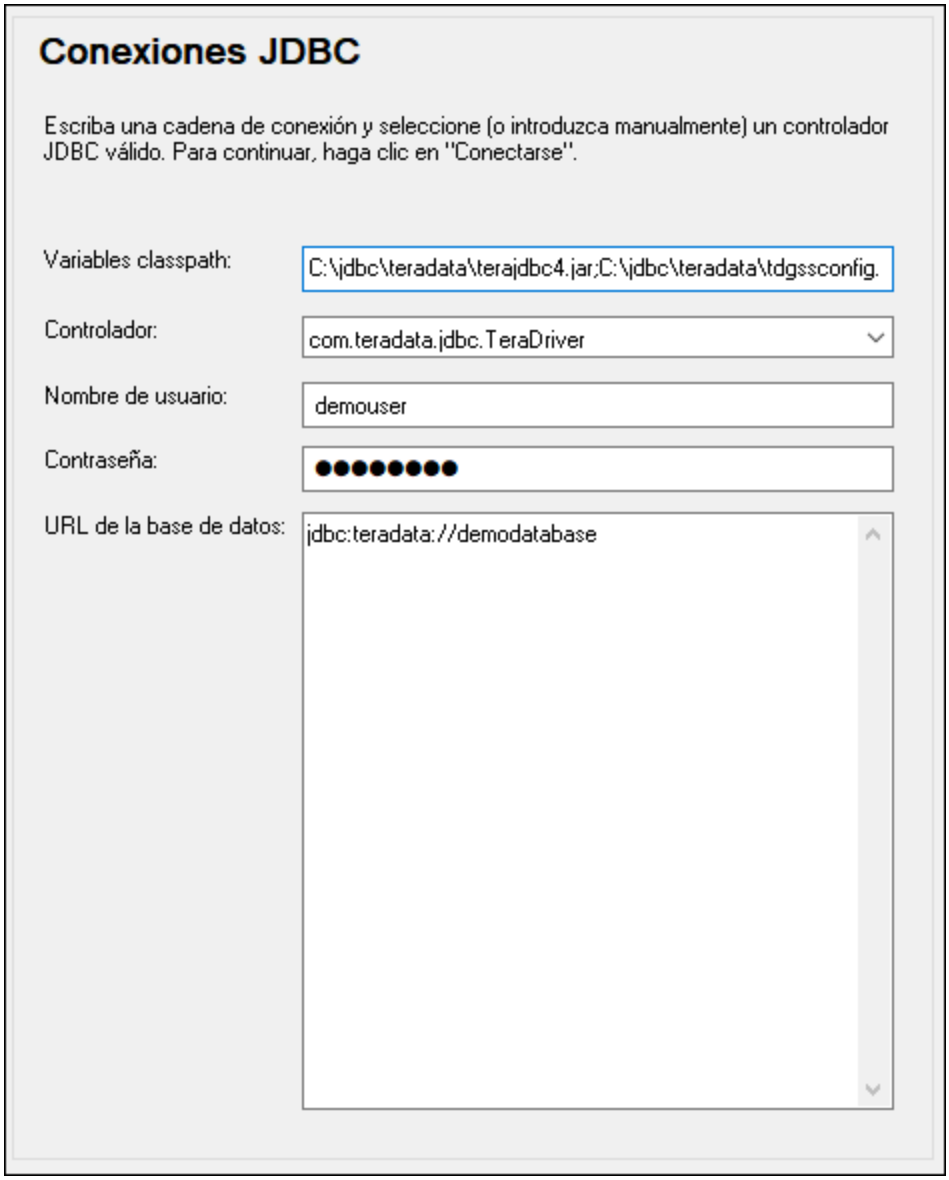
Requisitos:

- Debe tener instalado Java Runtime Environment (JRE) o Java Development Kit (JDK). Este último puede ser el JDK de Oracle o uno de código abierto, como Oracle OpenJDK. XMLSpy identifica la ruta de acceso al equipo virtual Java (JVM) a partir de estas ubicaciones, en este orden: a) la ruta personal de acceso al JVM que puede indicar en en las **Opciones** ; b) la ruta de acceso al JVM que se encuentra en el registro; c) la variable de entorno `JAVA_HOME`.
- Asegúrese de que la plataforma de XMLSpy (32 o 64 bits) es la misma que la de JRE/JDK.
- Debe contar con el controlador JDBC (archivos .jar que permiten conectarse a la base de datos) en el sistema operativo. Para este ejemplo usamos el controlador JDBC de Teradata 16.20.00.02 (https://downloads.teradata.com/download/connectivity/jdbc-driver).
- Disponer de los datos de conexión: host, base de datos, puerto, nombre de usuario y contraseña.

Para conectarse a Teradata por JDBC:

1. Inicie el asistente para la conexión de base de datos [953].

2. Haga clic en **Conexiones JDBC**.
3. Junto a *Variables Classpath* introduzca la ruta de acceso del archivo .jar que permite conectarse a la base de datos. Si es necesario, también puede introducir una lista de rutas de archivo .jar separadas por punto y comas. Para este ejemplo los archivos .jar están ubicados en esta ruta de acceso: **C:\jdbc\teradata**. Recuerde que puede dejar vacío el cuadro *Variables Classpath* si añadió las rutas de archivo .jar a la variable de entorno CLASSPATH del sistema operativo (véase Configurar la variable CLASSPATH 977).
4. En el cuadro *Controlador* seleccione **com.teradata.jdbc.TeraDriver**. Recuerde que esta entrada está disponible si en el cuadro *Variables Classpath* o en la variable de entorno CLASSPATH del sistema operativo se encuentra una ruta de archivo .jar válida (ver paso anterior).



5. Introduzca el nombre de usuario y la contraseña de la base de datos en las casillas correspondientes.
6. Introduzca la cadena de conexión en el cuadro de texto *URL de la base de datos* (reemplazando el valor resaltado con el valor correspondiente).

 © 2016-2022 Altova GmbH

```
jdbc:teradata://nombreServidorBaseDatos
```

7. Por último, haga clic en **Conectarse**.

19.1.10.20 Teradata (ODBC)

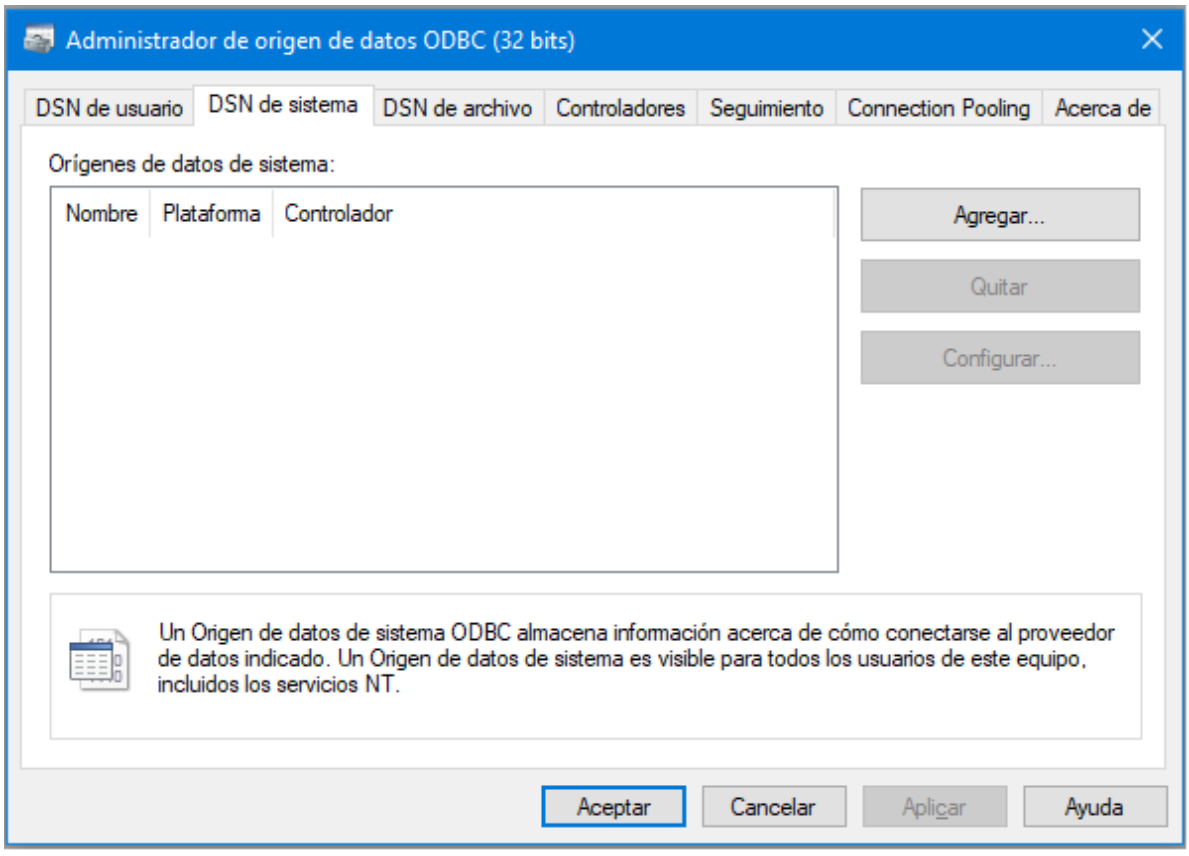
Este ejemplo explica cómo conectarse a un servidor de base de datos Teradata por ODBC.

Requisitos:

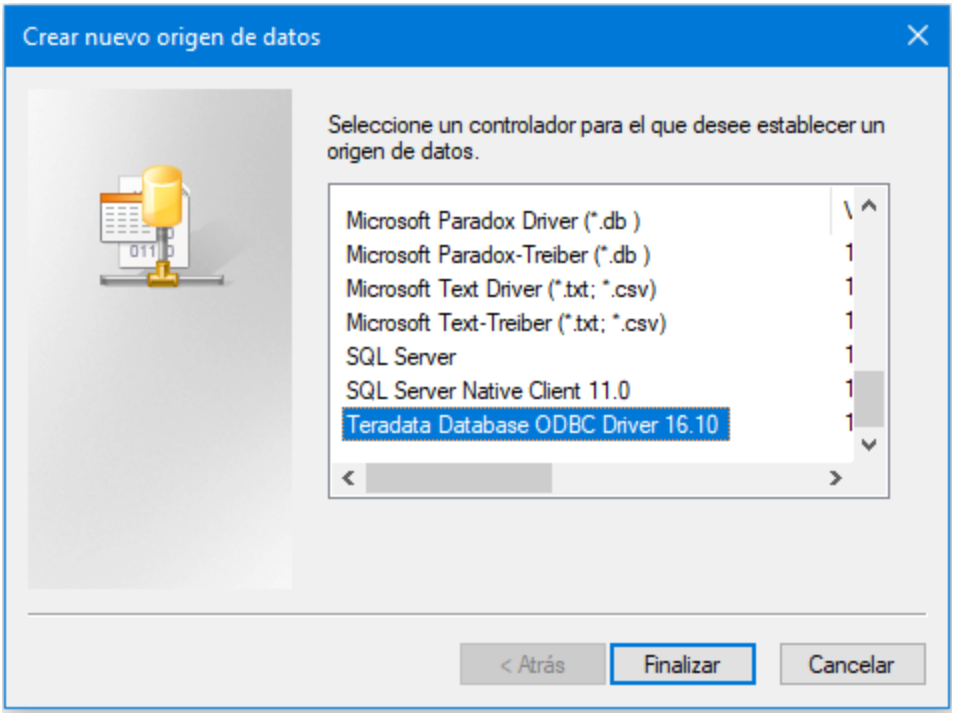
- Debe tener instalado el controlador ODBC de Teradata (https://downloads.teradata.com/download/connectivity/odbc-driver/windows). Para este ejemplo se utiliza el controlador ODBC de Teradata para Windows versión 16.20.00.
- Disponer de los datos de conexión: host, nombre de usuario y contraseña.

Para conectarse a Teradata por ODBC:

1. Pulse la tecla Windows, teclee "ODBC" y seleccione **Configurar orígenes de datos ODBC (32 bits)** en la lista de sugerencias. Si tiene un controlador ODBC de 64 bits, seleccione la opción **Configurar orígenes de datos ODBC (64 bits)** y utilice XMLSpy de 64 bits a partir de ese momento.

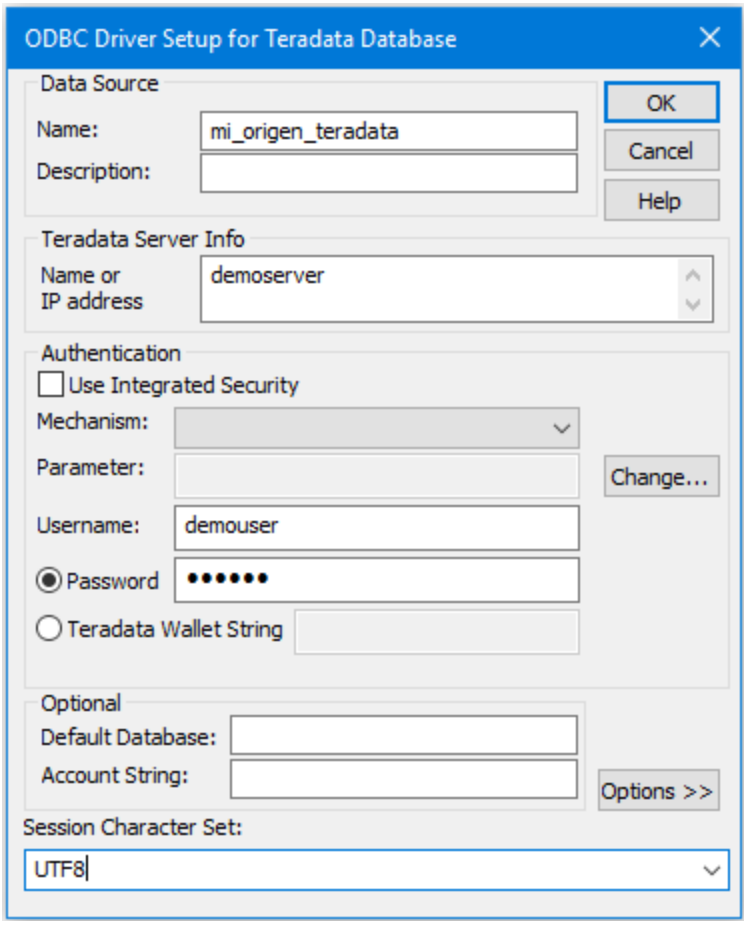


2. Haga clic en la pestaña *DSN de sistema* y después haga clic en **Agregar**.

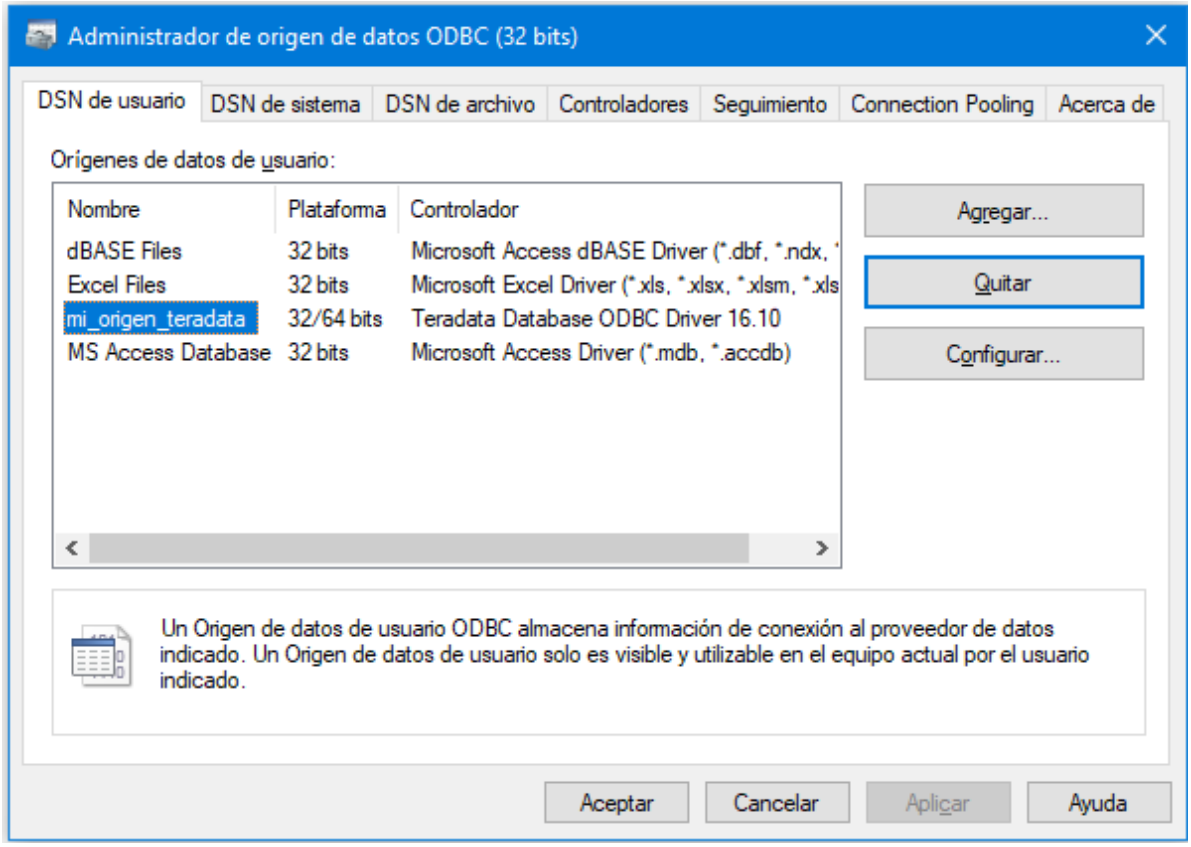


3. Seleccione **Teradata Database ODBC Driver** y después haga clic en **Finalizar**.

© 2016-2022 Altova GmbH

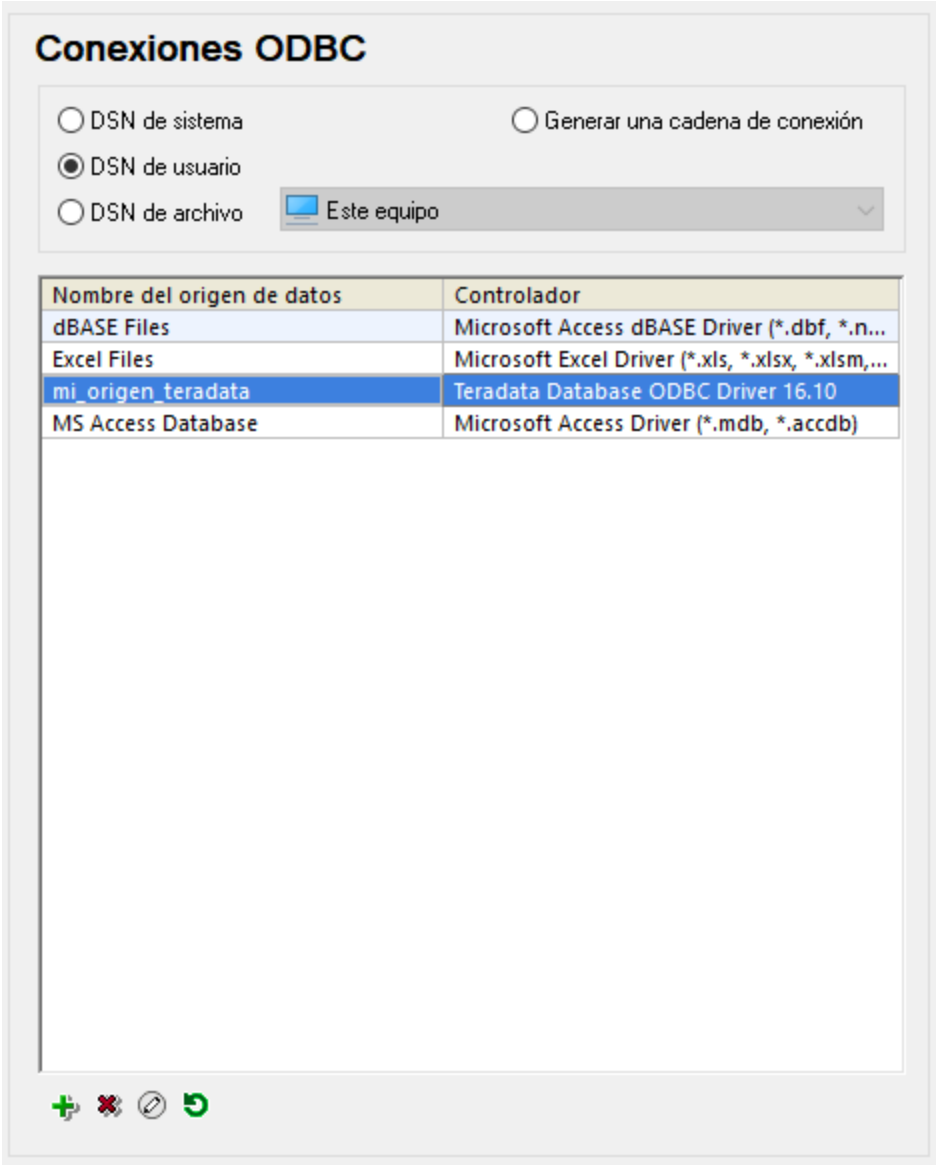


4. Introduzca el nombre y, si quiere, una descripción que le ayude a identificar este origen de datos ODBC más adelante. Además, deberá introducir las credenciales de la conexión de base de datos (servidor de BD, usuario y contraseña) y, si quiere, seleccione una base de datos.
5. Haga clic en **Aceptar**. El origen de datos aparecerá en la lista.



6. Ejecute XMLSpy e inicie el asistente para la conexión de base de datos 953.
7. Haga clic en **Conexiones ODBC**.

© 2016-2022 Altova GmbH



8. Haga clic en *DSN de sistema,* seleccione el origen de datos creado en los pasos anteriores y después haga clic en el botón **Conectarse**.

Nota: si recibe el mensaje de error "Controlador devuelto no válido (o error en la devolución) SQL_DRIVER_ODBC_VER: 03.80", compruebe que la ruta de acceso del cliente ODBC (p. ej. **C:\Archivos de programa\Teradata\Client\16.10\bin**) existe en su variable de entorno PATH del sistema. Si falta esta ruta de acceso, entonces deberá añadirla a mano.

19.2 Bases de datos compatibles

Las bases de datos compatibles y sus objetos raíz aparecen a continuación. Altova procura ofrecer compatibilidad con otras bases de datos, pero sólo se garantiza una conexión y un procesamiento correctos con las bases de datos de la lista. Si usa la versión de 64 bits de XMLSpy, compruebe que tiene acceso a los controladores de BD de 64 bits de la BD a la que quiere conectarse.

Base de datos	Observaciones
Firebird 2.x, 3.x	
IBM DB2 8.x, 9.x, 10.x, 11.x	
IBM Db2 for i 6.x, 7.4	Los archivos lógicos son compatibles y se muestran en vistas.
IBM Informix 11.70 y superior	
MariaDB 10 y superior	
Microsoft Access 2003 y superior	En el momento de escribir esta documentación (principios de septiembre de 2019) no hay ningún Microsoft Access Runtime disponible para Access 2019. Solo puede conectarse a la BD de Access 2019 con productos de Altova si tiene instalado Microsoft Access 2016 Runtime y solamente si la BD no usa el tipo de datos "Large Number" (número grande).
Microsoft Azure SQL Database	SQL Server 2016 codebase
Microsoft SQL Server 2005 y superior Microsoft SQL Server para Linux	
MySQL 5 y superior	
Oracle 9i y superior	
PostgreSQL 8 y superior	Son compatibles todas las conexiones PostgreSQL, tanto nativas como basadas en controladores, a través de interfaces como ODBC o JDBC. Las conexiones nativas no necesitan controladores.
Progress OpenEdge 11.6	
SQLite 3.x	Las conexiones SQLite son conexiones nativas y directas compatibles con el archivo de base de datos de SQLite. No se precisan controladores separados. En la vista Authentic, los datos que provienen de una base de datos SQLite no se pueden editar. Cuando intente guardar los datos de

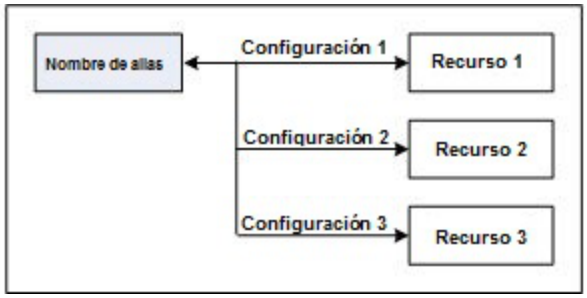
© 2016-2022 Altova GmbH

	SQLite desde la vista Authentic, un cuadro de mensaje le informará de esta limitación conocida.
Sybase ASE 15, 16	
Teradata 16	

20 Recursos globales de Altova

Los recursos globales de Altova son un conjunto de alias para recursos de archivo, carpeta y base de datos. Cada alias puede tener varias configuraciones y cada configuración se corresponde con un solo recurso (*imagen siguiente*). Por tanto, cuando un recurso global se utiliza como entrada de datos, se puede alternar la configuración del recurso global utilizado. La interfaz gráfica de usuario ofrece sencillos controles para cambiar de configuración activa.

Por ejemplo, si se asigna una hoja de estilos XSLT mediante un recurso global para transformar un documento XML, podemos crear varias configuraciones del recurso global, cada una apuntando a un archivo XSLT diferente. Una vez creadas estas configuraciones del recurso global, al cambiar de configuración alternamos el archivo XSLT usado para la transformación.



Los recursos globales no sólo sirven para cambiar de recurso dentro de una aplicación de Altova. También puede usarlos para generar y utilizar recursos de otras aplicaciones de Altova. Así podrá generar archivos automáticamente en una aplicación de Altova y usarlos en otra. Esto facilita y agiliza enormemente las tareas de desarrollo y de pruebas. Por ejemplo, en XMLSpy puede usar una hoja de estilos XSLT para transformar un archivo XML generado automáticamente a partir de una asignación de datos de Altova MapForce.

El uso de los recursos globales de Altova conlleva dos procesos:

- Definir los recursos globales 1043: los recursos se definen y sus definiciones se almacenan en un archivo XML. Estos recursos se pueden compartir con varias aplicaciones de Altova.
- Usar los recursos globales 1054: dentro de una aplicación de Altova puede localizar archivos con un recurso global en lugar de con una ruta de acceso. La ventaja es que puede cambiar de recurso con solo cambiar de configuración activa en XMLSpy.

Recursos globales en otros productos de Altova

Por el momento, puede definir y usar recursos globales en estos productos de Altova: XMLSpy, StyleVision, MapForce, Authentic Desktop, MobileTogether Designer y DatabaseSpy.

© 2016-2022 Altova GmbH

20.1 Definir recursos globales

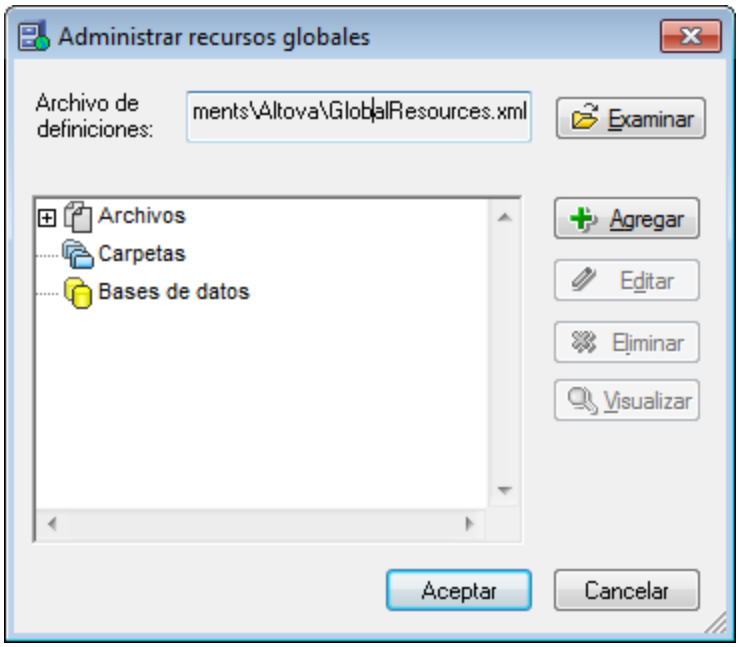
Los recursos globales de Altova se definen en el cuadro de diálogo "Administrar recursos globales". Hay dos maneras de abrir este cuadro de diálogo:

- Seleccionando el comando de menú **Herramientas | Recursos globales**.
- Haciendo clic en el icono **Administrar recursos globales** de la barra de herramientas Recursos globales (*imagen siguiente*).



El archivo de definiciones de recursos globales

La información sobre los recursos globales se almacena en un archivo XML llamado archivo de definiciones de recursos globales. Este archivo se crea y guarda cuando se define el primer recurso global en el cuadro de diálogo "Administrar recursos globales" (*imagen siguiente*).



Cuando se abre el cuadro de diálogo "Administrar recursos globales" por primera vez, el nombre y la ubicación predeterminados del archivo de definiciones de recursos globales aparece en el cuadro de texto *Archivo de definiciones* (*imagen anterior*):

```
C:\Users\<nombreUsuario>\Mis Documentos\Altova\GlobalResources.xml
```

Este archivo se define como archivo predeterminado de definiciones de recursos globales para todas las aplicaciones de Altova. Así puede guardar en este archivo recursos globales desde cualquier aplicación de Altova y el recurso global estará a disposición de todas las aplicaciones de Altova. Para definir y guardar un recurso global en el archivo de definiciones, añada el recurso global en el cuadro de diálogo "Administrar recursos globales" y haga clic en **Aceptar**.

Para seleccionar un archivo de definiciones de recursos globales ya existente como archivo de definiciones activo, búsquelo con el botón **Examinar** del cuadro de texto *Archivo de definiciones* (*imagen anterior*).

Nota: puede dar el nombre que quiera al archivo de definiciones y guardarlo en cualquier ubicación. Después en las aplicaciones de Altova sólo tiene que definir este archivo como archivo de definiciones de recursos globales (en el cuadro de texto *Archivo de definiciones*). Si usa un solo archivo de definiciones para todas las aplicaciones de Altova, los recursos definidos en el archivo estarán a disposición de todas las aplicaciones de Altova.

Nota: si lo prefiere, también puede crear varios archivos de definiciones de recursos globales. Sin embargo, en cada aplicación de Altova sólo puede estar activo un archivo de definiciones y, por tanto, sólo las definiciones de ese archivo estarán a disposición de la aplicación.

Administración de recursos globales: cómo añadir, editar, eliminar y guardar recursos

En el cuadro de diálogo "Administrar recursos globales" (*imagen anterior*) puede añadir un recurso global al archivo de definiciones seleccionado o editar/eliminar un recurso global. El archivo de definiciones de recursos globales organiza los recursos en tres grupos: archivos, carpetas y bases de datos (*ver imagen anterior*).

Para **añadir un recurso global**, haga clic en el botón **Agregar** y defina el recurso global en el cuadro de diálogo que aparece (*consulte los subapartados archivos[1045], carpetas[1050] y bases de datos[1043] para más información*). Después de definir el recurso global y guardarlo (haciendo clic en **Aceptar** en el cuadro de diálogo "Administrar recursos globales"), el recurso global se añade a la biblioteca de definiciones globales del archivo de definiciones seleccionado. El recurso global se identificará por medio de un alias.

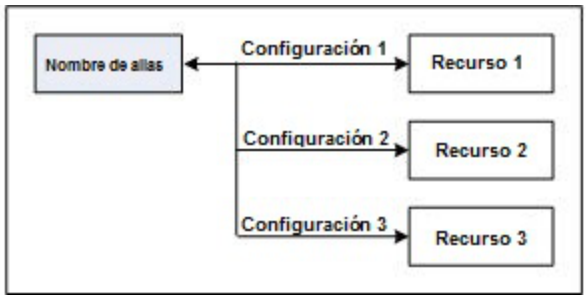
Para **editar un recurso global**, selecciónelo y haga clic en **Editar**. En el cuadro de diálogo que aparece puede realizar todos los cambios necesarios (*consulte los subapartados archivos[1045], carpetas[1050] y bases de datos[1043] para más información*).

Para **eliminar un recurso global**, selecciónelo y haga clic en **Eliminar**.

Cuando termine de añadir, editar y eliminar recursos globales, es necesario pulsar el botón **Aceptar** del cuadro de diálogo "Administrar recursos globales" para que **los cambios se guarden** en el archivo de definiciones de recursos globales.

Relacionar recursos globales con nombres de alias mediante configuraciones

Definir un recurso global implica asignar un nombre de alias a un recurso (a un archivo, una carpeta o una base de datos). Además se puede asignar el mismo nombre de alias a varios recursos. Cada asignación de nombre de alias se denomina configuración. Por tanto, el mismo nombre de alias se puede asociar a varios recursos mediante diferentes configuraciones (*ver imagen siguiente*).



 © 2016-2022 Altova GmbH

Después en la aplicación de Altova puede asignar alias en lugar de archivos y cambiar de recursos (los recursos asignados a cada alias) con solo cambiar de configuración de recursos globales (configuración activa). Por ejemplo, imagine que está trabajando con Altova XMLSpy y desea ejecutar una transformación en el documento XML `MiXML.xml`. Una opción es asignarle el alias de recurso global `MiXSLT` para poder usarla en transformaciones XSLT. En XMLSpy puede cambiar la configuración activa para usar archivos XSLT distintos. Si `Configuración 1` asigna `Primer.xslt` a `MiXSLT` y selecciona `Configuración 1` como configuración activa, entonces se usará `Primer.xslt` para la transformación. Es decir, puede usar configuraciones distintas para acceder a diferentes recursos a través de un solo alias. Este mecanismo puede ser de utilidad a la hora de probar y comparar recursos. Además, como los recursos globales son una característica compartida por varios productos de Altova, también podrá probar y comparar recursos en productos diferentes.

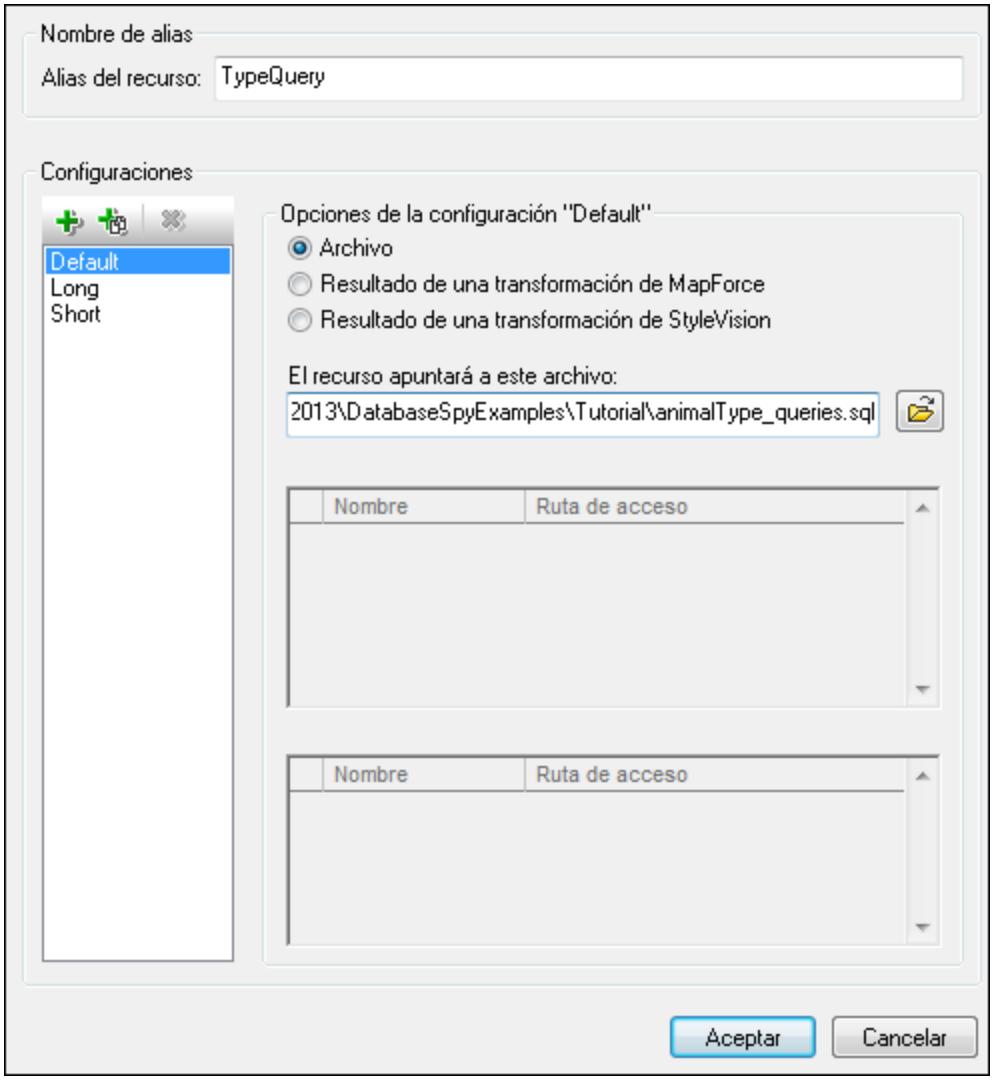
20.1.1 Archivos

El cuadro de diálogo "Recurso global" para archivos (*imagen siguiente*) se abre con el comando **Agregar | Archivos** del cuadro de diálogo Administrar recursos globales 1043. En este cuadro de diálogo puede indicar las configuraciones del alias que aparece en el cuadro de texto *Alias del recurso*. Después de definir las propiedades de las configuraciones (tal y como se explica más abajo), guarde la definición del alias haciendo clic en **Aceptar**.

Tras guardar una definición de alias puede agregar otro alias repitiendo los pasos que acabamos de describir (empezando por el comando **Agregar | Archivos** del cuadro de diálogo Administrar recursos globales) 1043.

Cuadro de diálogo "Recursos globales"

Los alias se definen en el cuadro de diálogo de recursos globales (*imagen siguiente*).



Iconos del cuadro de diálogo de recursos globales

Agregar configuración: abre el cuadro de diálogo "Agregar configuración", donde puede introducir el nombre de la configuración que se debe agregar.

Agregar una copia de configuración: abre el cuadro de diálogo "Agregar configuración", donde puede escribir el nombre de la configuración que debe crearse como copia de la configuración seleccionada.

Eliminar: elimina la configuración seleccionada.

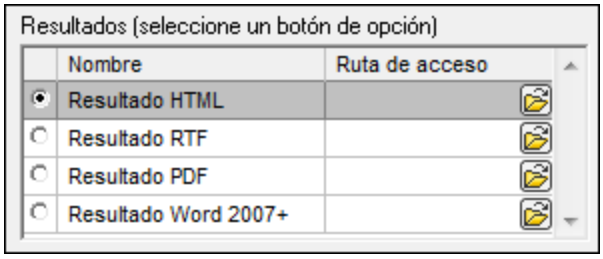
Abrir: examina el sistema de archivos para buscar el archivo que se debe usar para crear el recurso global.

Definir el alias

Siga estos pasos para definir el alias (el nombre de alias y sus configuraciones):

© 2016-2022 Altova GmbH

1. *Póngale un nombre al alias:* escriba el nombre del alias en el cuadro de texto *Alias del recurso.*
2. *Añada configuraciones:* el panel *Configuraciones* tiene por defecto una configuración llamada `Default` (*ver imagen anterior*), que no se puede editar ni eliminar. Puede agregar más configuraciones de este modo: (i) haga clic en los iconos **Agregar configuración** o **Agregar copia de configuración** y (ii) póngale un nombre a la configuración en el cuadro de diálogo que aparece. Cada configuración que añada aparecerá en la lista *Configuraciones*. En la imagen anterior, por ejemplo, se añadieron las configuraciones `Long` y `Short`. El comando **Agregar copia de configuración** permite copiar la configuración seleccionada y después modificarla.
3. *Seleccione un tipo de recurso para cada configuración:* seleccione una configuración de la lista *Configuraciones* y, en el panel *Opciones de la configuración*, especifique el tipo de recurso para esa configuración: puede escoger entre (i) un archivo, (ii) el resultado de una transformación de Altova MapForce y (iii) el resultado de una transformación de Altova StyleVision. Si selecciona una transformación de MapForce o de StyleVision, entonces la transformación se lleva a cabo con MapForce o StyleVision usando el archivo `.mfd`/`.sps` y el archivo de entrada correspondiente.
4. *Seleccione un archivo para el tipo de recurso:* pulse el icono Examinar del cuadro *El recurso apuntará a...* y seleccione el archivo, el archivo .mfd (para transformaciones de MapForce) o el archivo .sps (para transformaciones de StyleVision). Si la transformación permite usar varias entradas y varios resultados, podrá elegir cuál de ellos se utiliza. Por ejemplo, dependiendo de qué edición de StyleVision tenga instalada, podrá elegir entre unos resultados u otros (en la imagen siguiente se ofrecen documentos de resultados de la edición Enterprise Edition).



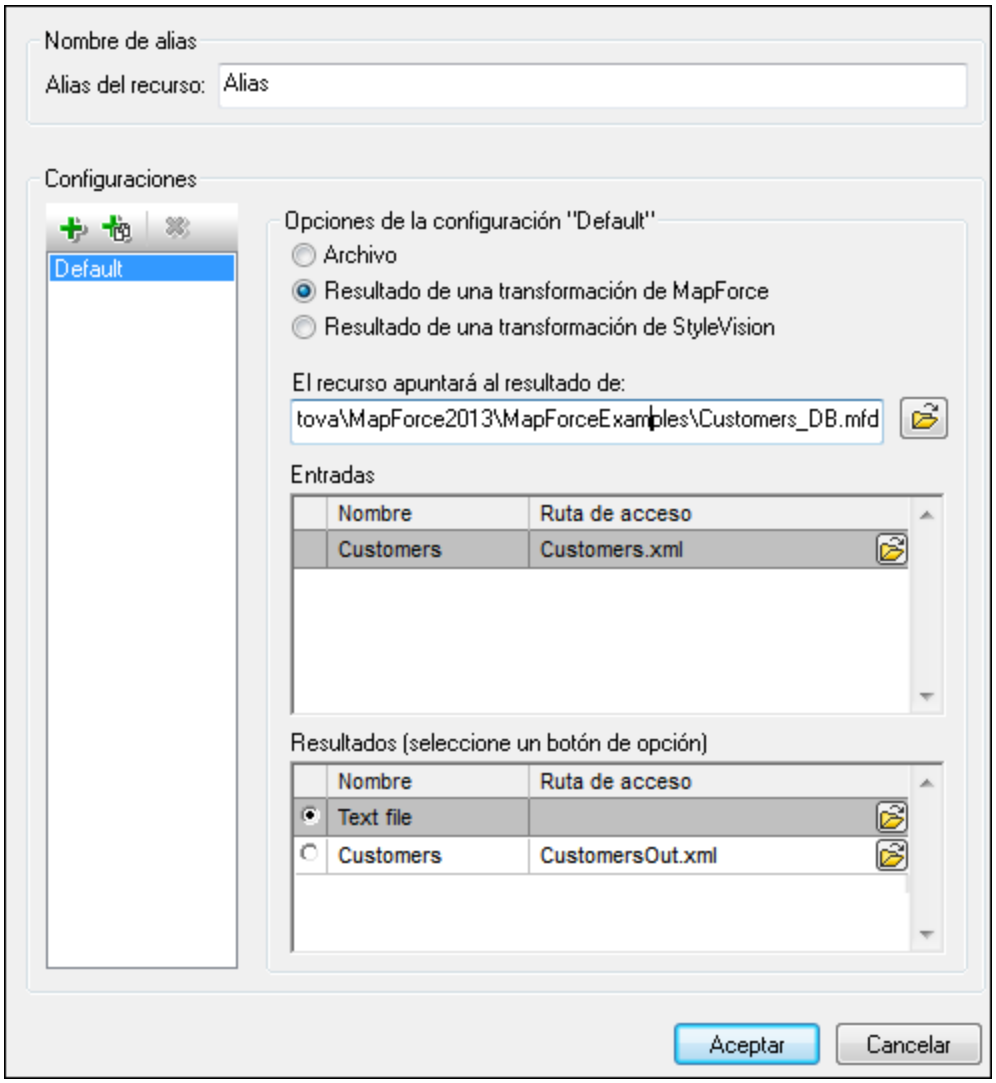
   Seleccione una opción (por ejemplo, en la imagen anterior se seleccionó el resultado en formato HTML). Si el recurso es el resultado de una transformación, el resultado se puede guardar como archivo o como recurso global. Haga clic en el icono y seleccione **Examinar...** (si desea guardar el resultado como archivo) o **Elegir otro recurso global...** (si desea guardar el resultado como recurso global). Si no selecciona ninguna de estas dos opciones, el resultado de la transformación se cargará como archivo temporal cuando se invoque el recurso global.

5. *Si es necesario, defina más configuraciones*: si quiere, puede añadir más configuraciones y especificar un recurso para cada una de ellas. Para ello repita los pasos 3 y 4. Puede agregar una configuración nueva a la definición de alias en todo momento.
6. *Guarde la definición de alias:* haga clic en **Aceptar** para guardar el alias y todas sus configuraciones como recurso global. El recurso global nuevo aparecerá bajo el nodo *Archivos* del cuadro de diálogo Administrar recursos globales[1043].

Resultados de transformaciones de MapForce

Altova MapForce realiza asignaciones de datos entre esquemas de entrada (ya existentes) y esquemas de salida (nuevos). Esta asignación, creada por el usuario de MapForce, se conoce como diseño de MapForce (o archivo MFD). Los archivos XML, de texto, bases de datos, etc. correspondientes a los esquemas de entrada se pueden usar como orígenes de datos. MapForce genera archivos de datos de resultados que se corresponden con el esquema de salida. Este documento de salida es el *resultado de una transformación de MapForce* que se convertirá en recurso global.

Si desea definir un archivo de datos generado con MapForce como recurso global, seleccione la opción *Resultado de una transformación de MapForce* en este cuadro de diálogo:



- ***Un archivo de diseño de MapForce (archivo `.mfd`).*** Es necesario especificar este archivo en el cuadro de texto *El recurso apuntará al resultado de* del cuadro de diálogo (*ver imagen anterior*).
- ***Uno o más archivos de datos de entrada.*** El archivo MFD seleccionado se analiza y, en función de la información que contenga sobre los esquemas de entrada, el panel *Entradas* muestra una lista de los archivos de datos de entrada (*ver imagen anterior*). Si prefiere usar otro archivo, haga clic en el icono **Examinar** y seleccione un archivo diferente.
- ***Un archivo de resultados.*** Si el archivo MFD tiene varios esquemas de salida, estos aparecen en la lista del panel *Resultados* (*imagen anterior*) y es necesario seleccionar uno. Si la ubicación del archivo de resultados de un esquema de salida está definida en el archivo MFD, esta ubicación aparecerá en el panel *Resultados.* Por ejemplo, en la imagen anterior puede ver que el archivo MFD dice que el esquema de salida `Customers` tiene un archivo de datos XML predeterminado (llamado `CustomersOut.xml`), mientras que el esquema de salida `Text file` no tiene ninguna asociación de archivo en el archivo MFD. Puede usar la ubicación de archivo predeterminada que aparece en el panel *Resultados* o especificar una ubicación nueva. El resultado de la transformación de MapForce se

 © 2016-2022 Altova GmbH

guardará en la ubicación de archivo del esquema de salida seleccionado. Este es el archivo que se usará como recurso global.

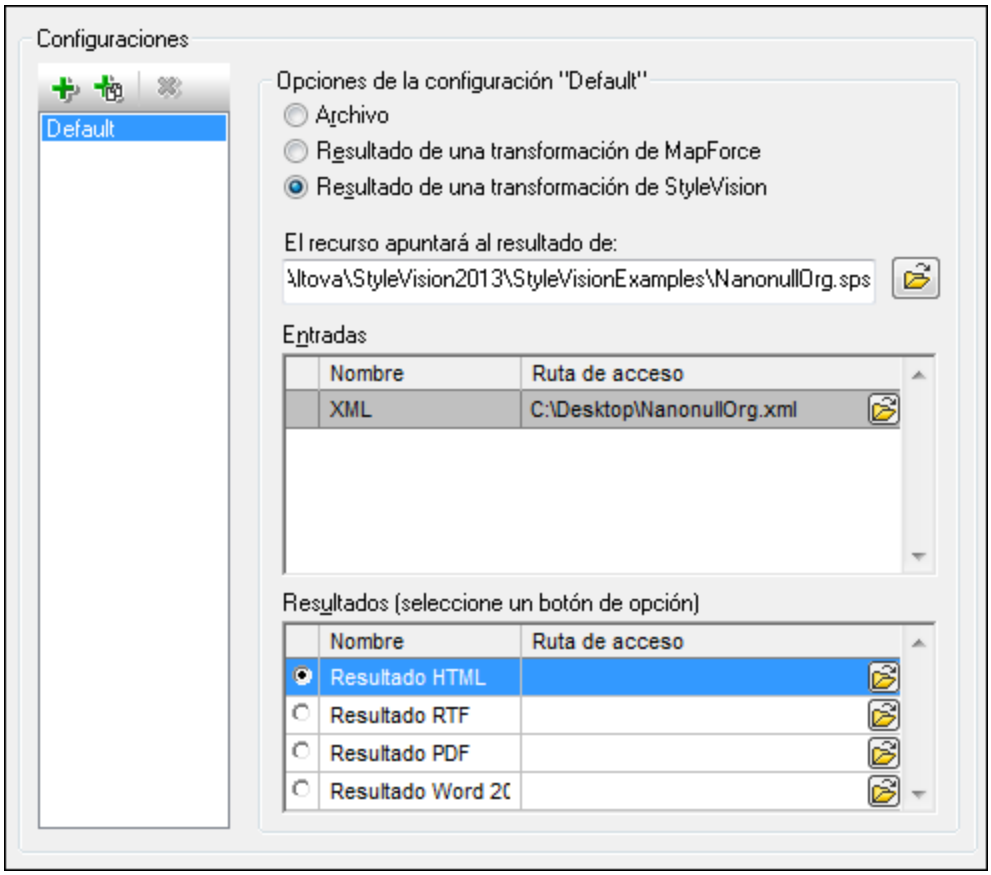
Nota: la ventaja de usar esta opción (resultado de una transformación de MapForce) es que la transformación se ejecuta al invocarse el recurso global. Esto significa que el recurso global contendrá los datos más recientes (de los archivos de entrada).

Nota: ya que para ejecutar la transformación se utiliza MapForce, es necesario tener instalado este producto de Altova. De lo contrario esta característica no funcionará.

Resultados de transformaciones de StyleVision

Altova StyleVision se utiliza para crear archivos SPS (StyleVision Power Stylesheet). Estos archivos generan hojas de estilos XSLT que se usan para transformar documentos XML en documentos de salida en varios formatos (HTML, PDF, RTF, Word 2007+, etc.) Si selecciona la opción *Resultado de una transformación de StyleVision*, el documento de resultados creado por StyleVision será el recurso global asociado a la configuración seleccionada.

Si selecciona la opción Transformación de StyleVision (*imagen siguiente*), es necesario especificar también estos archivos:



- ***Un archivo .sps (SPS).*** Seleccione el archivo SPS en el cuadro de texto *El recurso apuntará al resultado de* (*ver imagen anterior*).

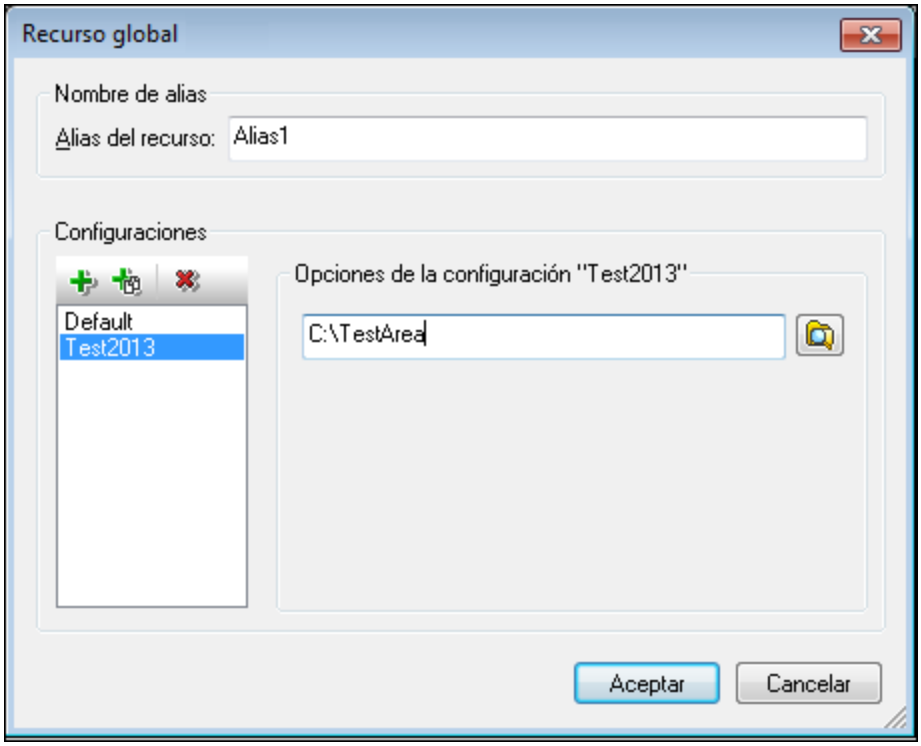
- ***Archivos de entrada***. El archivo de entrada puede venir dado por el archivo SPS. Si es así, el archivo de entrada aparecerá automáticamente en el panel *Inputs*. En este panel puede elegir el archivo de entrada que quiera. Si el panel está vacío, entonces añada un archivo.
- ***Archivos de resultados.*** Seleccione el formato de salida en el panel *Resultados* y especifique la ubicación del archivo.

Nota: la ventaja de seleccionar esta opción (resultado de la transformación de StyleVision) es que la transformación se ejecuta al invocarse el recurso global. Esto significa que el recurso global contendrá los datos más recientes (de los archivos de entrada).

Nota: ya que para ejecutar la transformación se utiliza StyleVision, es necesario tener instalado este producto de Altova. De lo contrario esta característica no funcionará.

20.1.2 Carpetas

En el cuadro de diálogo "Recurso global" para carpetas (*imagen siguiente*), siga los pasos que aparecen más abajo para agregar un recurso de carpeta.



Iconos del cuadro de diálogo de recursos globales

Agregar configuración: abre el cuadro de diálogo "Agregar configuración", donde puede introducir el nombre de la configuración que se debe agregar.

Agregar una copia de configuración: abre el cuadro de diálogo "Agregar configuración", donde puede escribir el nombre de la configuración que debe crearse como copia de la configuración seleccionada.

© 2016-2022 Altova GmbH


Eliminar: elimina la configuración seleccionada.


Abrir: examina el sistema de archivos para buscar el archivo que se debe usar para crear el recurso global.

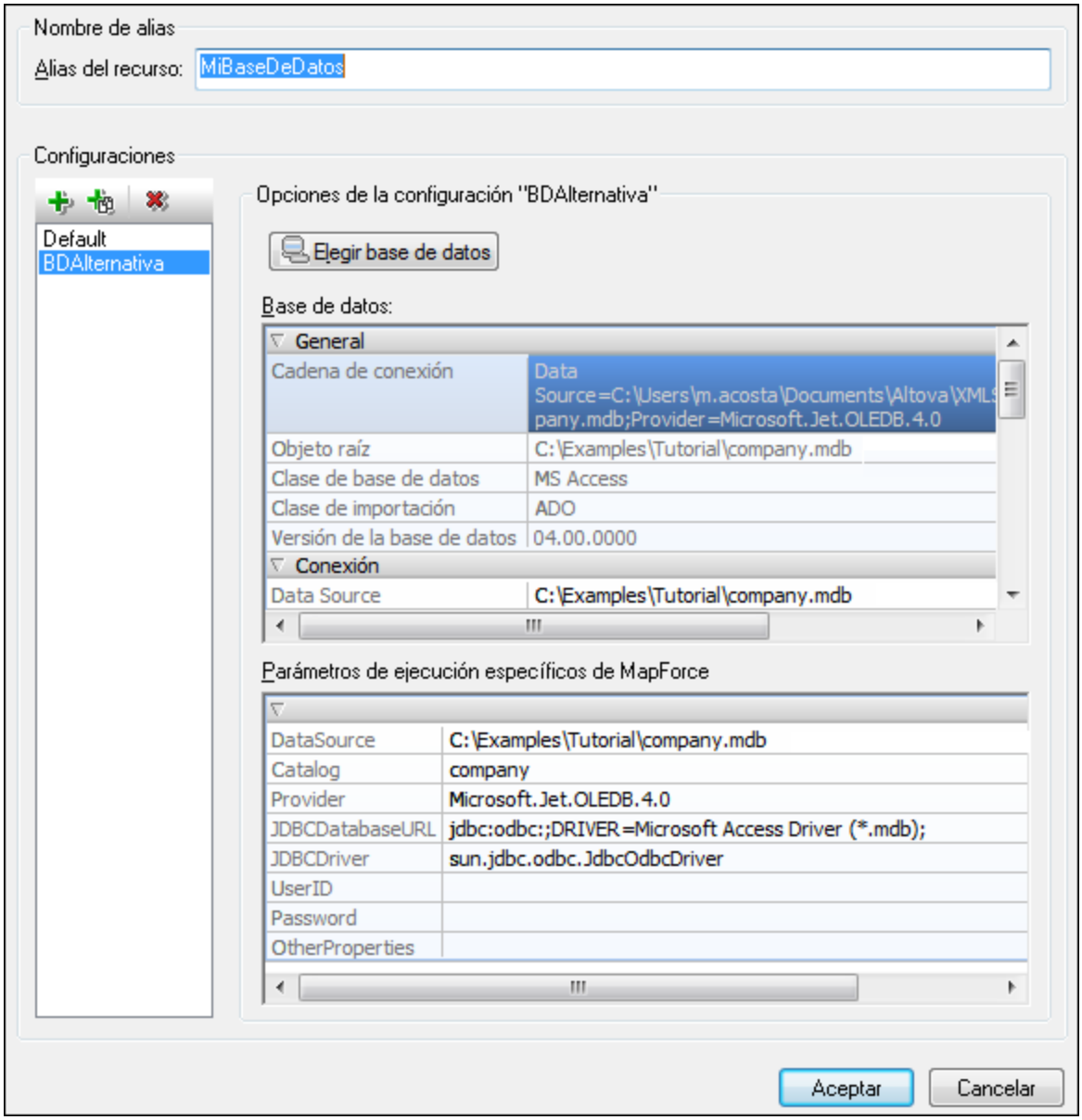
Definir el alias

Siga estos pasos para definir el alias (su nombre y su configuración):

1. *Póngale un nombre al alias:* escriba un nombre de alias en el cuadro de texto *Alias del recurso*.
2. *Añada configuraciones:* el panel Configuraciones, situado a la izquierda, tendrá una configuración predeterminada llamada *Default* (*imagen anterior*). Esta configuración Default no se puede eliminar ni cambiar de nombre. Puede introducir todas las configuraciones que quiera para el alias seleccionado. Para agregar una configuración, haga clic en el icono **Agregar configuración** o en el icono **Agregar copia de configuración**. Aparece el cuadro de diálogo "Agregar configuración". Escriba el nombre de la configuración y haga clic en **Aceptar**. La configuración nueva aparece en la lista del panel Configuraciones. Repita estos pasos para crear más configuraciones.
3. *Seleccione una carpeta como recurso de la configuración:* seleccione una de las configuraciones del panel Configuraciones y busque la carpeta que desea crear como recurso global. Si se necesitan credenciales de seguridad para acceder a una carpeta, introdúzcalos en los campos *Nombre de usuario* y *Contraseña*.
4. *Si es necesario, defina más configuraciones*: indique un recurso de carpeta por cada configuración (es decir, repita el paso 3 tantas veces como configuraciones). Puede agregar configuraciones nuevas a la definición de alias en todo momento.
5. *Guarde la definición de alias:* en el cuadro de diálogo "Recurso global" haga clic en el botón **Aceptar** para guardar como recurso global el alias y todas sus configuraciones. El recurso global aparecerá en el cuadro de diálogo de recursos globales, bajo el nodo *Carpetas*.

20.1.3 Bases de datos

En el cuadro de diálogo "Recurso global" para bases de datos (*imagen siguiente*), siga los pasos que aparecen más abajo para agregar un recurso de base de datos.



Iconos del cuadro de diálogo de recursos globales

Agregar configuración: abre el cuadro de diálogo "Agregar configuración", donde puede introducir el nombre de la configuración que se debe agregar.

Agregar una copia de configuración: abre el cuadro de diálogo "Agregar configuración", donde puede escribir el nombre de la configuración que debe crearse como copia de la configuración seleccionada.

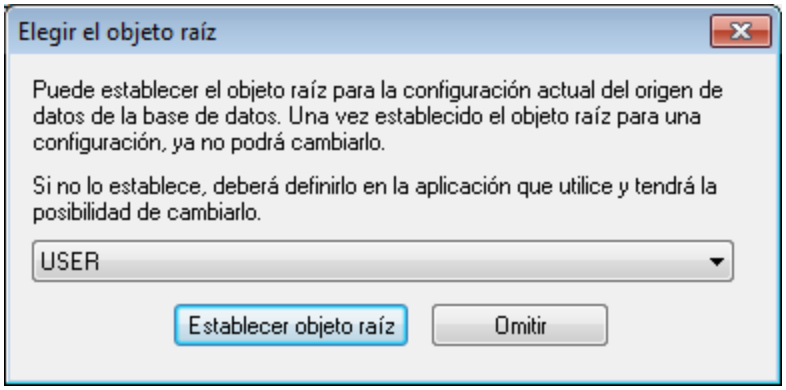
Eliminar: elimina la configuración seleccionada.

Definir el alias

Siga estos pasos para definir el alias (su nombre y su configuración):

© 2016-2022 Altova GmbH

1. *Póngale un nombre al alias:* escriba el nombre del alias en el cuadro de texto *Alias del recurso*.
2. *Agregar configuraciones:* el panel Configuraciones, situado a la izquierda, tendrá una configuración predeterminada llamada *Default* (*imagen anterior*). Esta configuración Default no se puede eliminar ni cambiar de nombre. Puede introducir todas las configuraciones que quiera para el alias seleccionado. Para agregar una configuración, haga clic en el icono **Agregar configuración** o en el icono **Agregar copia de configuración**. Aparece el cuadro de diálogo "Agregar configuración". Escriba el nombre de la configuración y haga clic en **Aceptar**. La configuración nueva aparece en la lista del panel Configuraciones. Repita estos pasos para crear las configuraciones que necesite.
3. *Inicie la selección de una base de datos como recurso de la configuración:* seleccione una de las configuraciones de panel Configuraciones y pulse el botón **Elegir base de datos**. Aparece el cuadro de diálogo "Crear conexión para recursos globales".
4. *Conéctese a la base de datos:* seleccione si desea crear una conexión a la base de datos con el asistente para la conexión, usar una conexión existente, una conexión ADO, ODBC o JDBC. Termine de definir el método de conexión tal y como se indica en la sección Conectarse a una base de datos 952. Si ya se estableció una conexión a una base de datos desde XMLSpy, puede hacer clic en el icono **Conexiones existentes** y seleccionar la BD de la lista de conexiones que aparece.
5. *Seleccione el objeto raíz:* si se conecta a un servidor de base de datos en el que se puede seleccionar un objeto raíz, entonces aparece el cuadro de diálogo "Elegir el objeto raíz" (*imagen siguiente*). En este diálogo puede seleccionar el objeto raíz del servidor, que es el objeto raíz que se cargará cuando se utilice esta configuración. Para terminar haga clic en **Establecer objeto raíz**.



   Si prefiere no seleccionar ningún objeto raíz (haciendo clic en el botón **Omitir**), podrá seleccionar el objeto raíz en el momento en que se cargue el recurso global.

6. *Si es necesario, defina más configuraciones:* especifique un recurso de base de datos por cada configuración que necesite (es decir, repita los pasos 3, 4 y 5 tantas veces como configuraciones necesite). Puede añadir una configuración nueva a la definición de alias en todo momento.
7. *Guarde la definición de alias:* en el cuadro de diálogo "Recurso global" haga clic en **Aceptar** para guardar como recurso global el alias y todas sus configuraciones. El recurso global aparecerá en el cuadro de diálogo "Administrar recursos globales".

20.2 Usar recursos globales

Hay varios tipos de recursos globales (de tipo archivo, carpeta y base de datos). Algunos casos de uso de los recursos globales en XMLSpy se describen aquí: Archivos y carpetas[1054] y Bases de datos[1057].

Selecciones que determinan el recurso que se utiliza

Hay dos selecciones generales que determinan qué recursos globales se pueden usar en general y cuáles se pueden usar en cada momento:

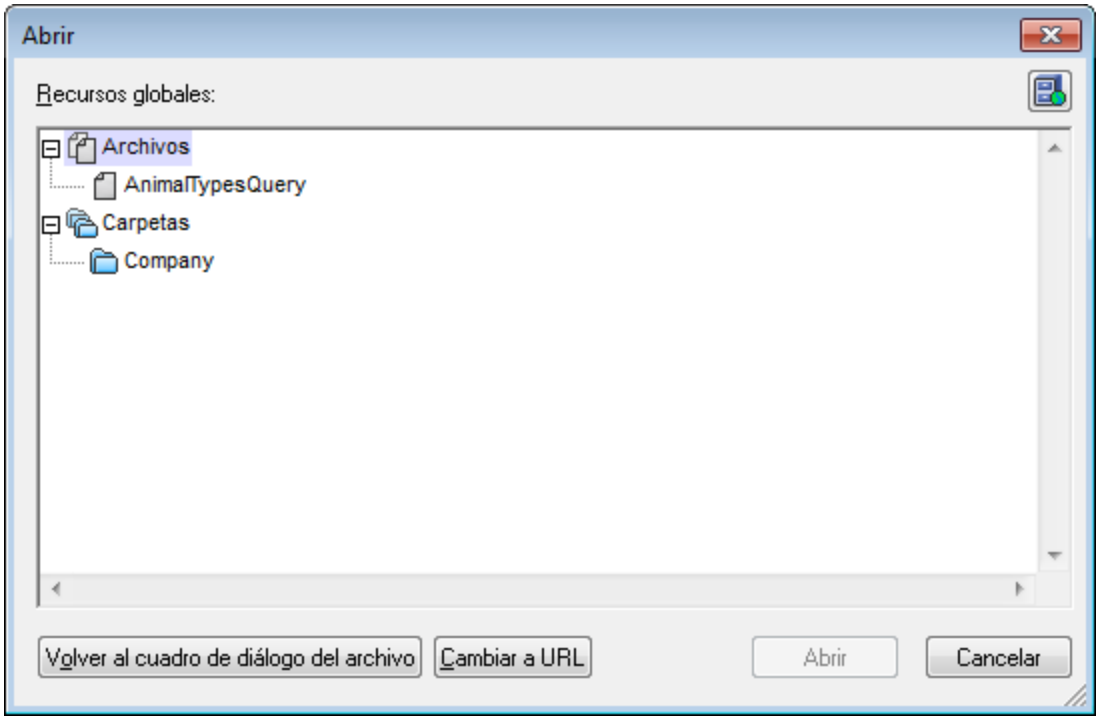
- El *archivo XML de recursos globales* seleccionado en el cuadro de diálogo Recurso global[1043]. Las definiciones de recursos globales presentes en el archivo XML de recursos globales están a disposición de todos los archivos que estén abiertos en la aplicación. Las únicas definiciones disponibles son las del archivo XML de recursos globales. Puede cambiar de archivo XML de recursos globales cuando quiera y las definiciones del nuevo archivo activo sustituyen inmediatamente a las del archivo activo anterior. Por tanto, el archivo XML de recursos globales activo determina: (i) qué recursos globales se pueden asignar y (ii) qué recursos globales se pueden buscar (p. ej. si se asigna un recurso global de un archivo XML de recursos globales, pero en el archivo XML de recursos globales activo no hay ningún recurso global que se llame así, entonces no se puede buscar el recurso global, o alias, asignado).

- La *configuración activa* seleccionada en la opción de menú **Herramientas | Configuración activa**[1558] o en la barra de herramientas Recursos globales. Al hacer clic en este comando (o en la lista desplegable de la barra de herramientas) aparecen las configuraciones de todos los alias. Si selecciona una configuración, la configuración es la configuración activa para toda la aplicación. Es decir, cuando se usa un recurso global (o alias), se carga el recurso correspondiente a la configuración activa del alias usado. La configuración activa se aplica a todos los alias usados. Si un alias no tiene una configuración que se llame como la configuración activa, entonces se usa la configuración predeterminada de ese alias. La configuración activa no es relevante a la hora de asignar recursos. Sólo es importante a la hora de utilizar los recursos.

20.2.1 Asignar archivos y carpetas

Los recursos globales de tipo archivo no se asignan igual que los recursos globales de tipo carpeta. En los casos de uso[1055] que damos como ejemplo y que aparecen en este apartado, al hacer clic en el botón **Cambiar a recursos globales** aparece el cuadro de diálogo "Abrir recurso global" (*imagen siguiente*).

© 2016-2022 Altova GmbH



Administrar recursos globales: abre el cuadro de diálogo Administrar recursos globales[1043].

Si selecciona un archivo en este cuadro de diálogo, se asigna el archivo seleccionado. Si selecciona una carpeta, aparece un cuadro de diálogo "Abrir" donde puede examinar la carpeta para buscar el archivo deseado. La ruta de acceso al archivo seleccionado es relativa al recurso de carpeta. Por tanto, si un recurso global de tipo carpeta tuviese dos configuraciones, cada una apuntando a una carpeta diferente, se podrían seleccionar dos archivos con el mismo nombre pero ubicados en carpetas diferentes. Esto puede ser muy útil durante la fase de pruebas, por ejemplo.

Puede cambiar al cuadro de diálogo del archivo o al de la URL con solo pulsar el botón correspondiente. El icono **Administrar recursos globales** situado en la esquina superior derecha abre el cuadro de diálogo Administrar recursos globales[1043].

Casos de uso

Puede usar recursos globales de tipo archivo y carpeta en estos casos:

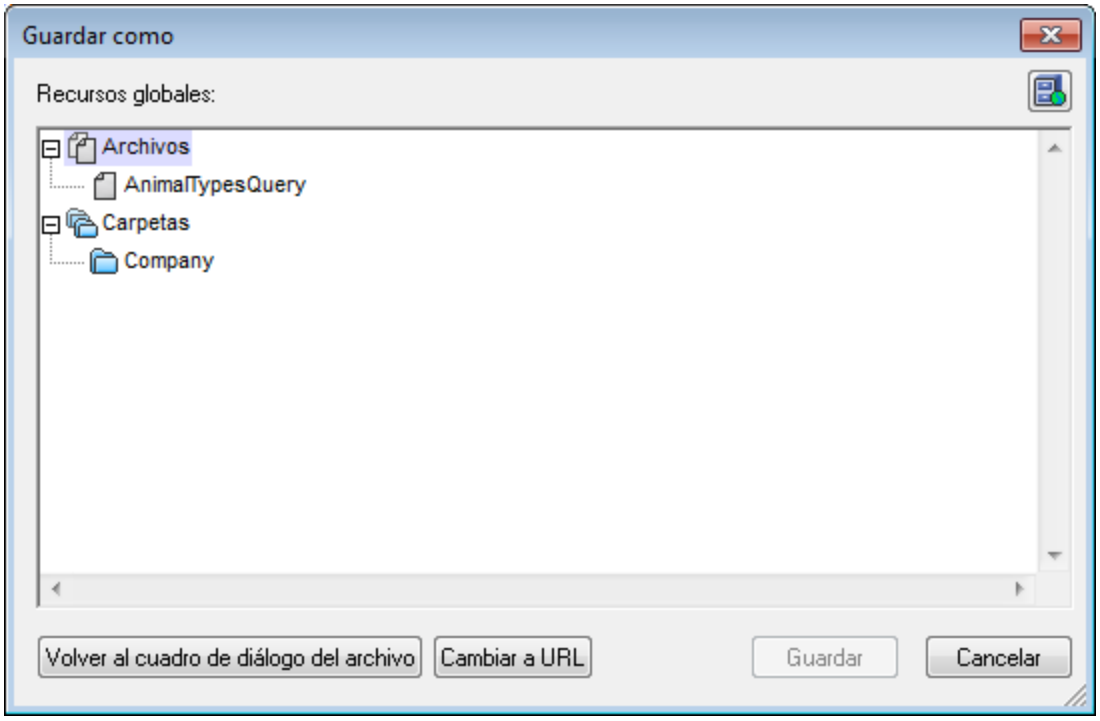
- Al abrir recursos globales[1056]
- Al guardar un archivo como recurso global[1056]
- Al asignar archivos para transformaciones XSLT[1057]
- En transformaciones XSLT[1057]
- Ejecuciones XQuery[1057]
- Al asignar una SPS[1057]

Abrir recursos globales

Puede abrir recursos globales en XMLSpy con el comando **Archivo | Abrir (Cambiar a recursos globales)**[1262]. Si se trata de un recurso global de tipo archivo, el archivo se abre directamente. Si se trata de un recurso global de tipo carpeta, entonces aparece un cuadro de diálogo "Abrir" donde puede examinar la carpeta seleccionada y buscar el archivo deseado. Una ventaja de editar archivos a través de recursos globales es que los archivos relacionados se pueden guardar en configuraciones de un recurso global y para abrirlos basta con cambiar de configuración. Recuerde guardar los cambios realizados en el archivo antes de cambiar de configuración.

Guardar un archivo como recurso global

Los archivos recién creados se pueden guardar como recurso global. También puede abrir un archivo ya existente y guardarlo como recurso global. Al hacer clic en los comandos **Archivo | Guardar** o **Archivo | Guardar como** aparece el cuadro de diálogo "Guardar". Pulse el botón **Cambiar a recursos globales** para acceder a los recursos globales disponibles (*imagen siguiente*), es decir, los alias definidos en el archivo XML de recursos globales actual.



Seleccione un alias y pulse el botón **Guardar**. Si el alias es un alias de archivo[1045] el archivo se guarda directamente. Si el alias es un alias de carpeta[1050], entonces aparece un cuadro de diálogo donde debe escribir el nombre con el que desea guardar el archivo. En ambos casos, el archivo se guarda en la ubicación que se definió para la configuración activa actual[1058].

Nota: cada configuración apunta a una ubicación de archivo concreta, que se indica en la definición de la configuración. Si el archivo que desea guardar como recurso global no tiene la misma extensión de archivo que el archivo guardado en la ubicación de archivo de la configuración, entonces puede haber errores de edición y validación al abrir este recurso global en XMLSpy. El motivo es que, al abrir el archivo, XMLSpy asume el tipo de archivo indicado en la definición de la configuración.

© 2016-2022 Altova GmbH

Asignar archivos para transformaciones XSLT

Puede asignar archivos XSLT a documentos XML y viceversa a través de recursos globales. Al hacer clic en los comandos para asignar archivos XSLT (**XSL/XQuery | Asignar XSL**[1403] y **XSL/XQuery | Asignar XSL-FO**[1403]) y archivos XML (**XSL/XQuery | Asignar archivo XML de muestra**[1404]) aparece el cuadro de diálogo de asignación. Al hacer clic en el botón **Examinar** aparece el diálogo "Abrir", donde puede hacer clic en el botón **Cambiar a recursos globales** y seleccionar un recurso global. Una gran ventaja de usar un recurso global para especificar archivos para transformaciones XSLT es que puede cambiar el archivo XSLT (o XML) con solo cambiar de configuración activa en XMLSpy, sin necesidad de asignar un archivo nuevo cada vez que desee transformar otro archivo. De este modo la transformación XSLT utiliza el archivo asociado a la configuración activa.

Transformaciones XSLT y ejecuciones de XQuery

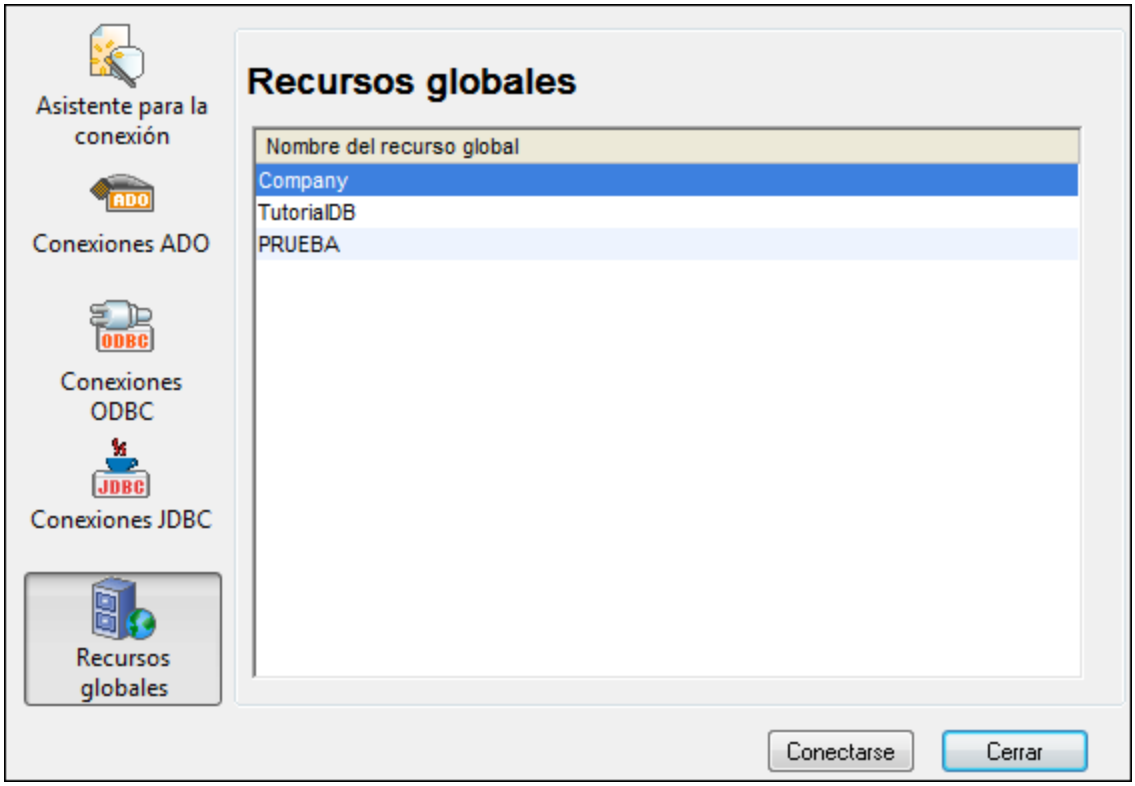
Al hacer clic en los comandos **XSL/XQuery | Transformación XSL**[1394] **XSL/XQuery | Transformación XSL-FO**[1395] o **XSL/XQuery | Ejecutar XQuery**[1400] aparece un cuadro de diálogo en el que puede buscar el archivo XSLT, XQuery y XML pertinentes. Haga clic en el botón **Examinar** y después en el botón **Cambiar a recursos globales**. Aparece el cuadro de diálogo "Abrir" (*ver primera imagen de este apartado*[1054]). Para la transformación se usa el archivo asociado a la configuración activa del recurso global seleccionado en este diálogo.

Asignar una SPS

Cuando asigne una hoja de estilos de StyleVision a un archivo XML (**Authentic | Asignar hoja de estilos de StyleVision**), puede usar un recurso global para localizar la hoja de estilos. Haga clic en el botón **Examinar** y después en el botón **Cambiar a recursos globales** y aparece el diálogo "Abrir recurso global" (*ver primera imagen de este apartado*[1054]). Si se selecciona un recurso global, puede cambiar el documento XML de la vista Authentic con solo cambiar la configuración activa en XMLSpy.

20.2.2 Asignar bases de datos

Cuando ejecute un comando que importa datos o una estructura de datos (p. ej. un esquema XML) desde una BD a XMLSpy (p. ej. el comando **Convert | Importar datos de una base de datos**), tiene la opción de usar un recurso global (*imagen siguiente*).



Al hacer clic en el icono **Recursos globales** del cuadro de diálogo "Conectarse a un origen de datos" se enumeran todos los recursos de base de datos definidos en el archivo XML de recursos globales[1043]. Seleccione el recurso global que desea usar y haga clic en **Conectarse**. Si el recurso global seleccionado tiene más de una configuración, se usa el recurso para la configuración activa (consulte la opción de menú **Herramientas | Configuración activa** o la barra de herramientas de recursos globales) y a continuación se establece la conexión. Ahora debe seleccionar las estructuras de datos y los datos que se deben usar siguiendo las instrucciones de los apartados Crear un esquema XML a partir de una BD[1467] y Importar datos de BD[1455].

20.2.3 Cambiar de configuración

En la aplicación no puede haber más de una configuración activa al mismo tiempo y la configuración activa afecta a toda la aplicación. Esto quiere decir que la configuración activa está activa para todos los alias en todos los archivos que estén abiertos en la aplicación y en todas las conexiones a orígenes de datos. Si un alias no tiene una configuración con el nombre de la configuración activa, entonces se usa la configuración predeterminada (*Default*) de dicho alias.

Hay dos maneras de cambiar de configuración activa:

- Con el comando de menú **Herramientas | Configuración Activa**, donde puede seleccionar qué configuración que desea usar.
- En el cuadro combinado de la barra de herramientas Recursos globales (*imagen siguiente*), donde puede seleccionar la configuración deseada.

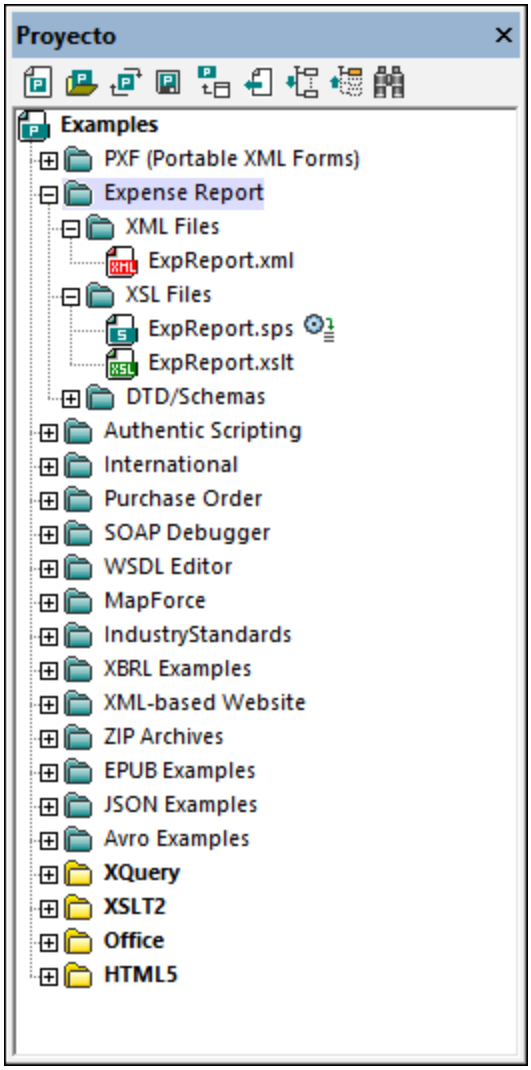
© 2016-2022 Altova GmbH



De este manera, al cambiar de configuración activa, puede cambiar los archivos de origen que se asignaron a través de recursos globales.

21 Proyectos

Un proyecto es un conjunto de archivos relacionados entre sí, de acuerdo con el criterio del usuario. Por ejemplo, en la imagen siguiente, el proyecto `Examples` reúne varios archivos de ejemplo y los agrupa en carpetas de ejemplos, cada una de las cuales incluye varias subcarpetas. Dentro del proyecto `Examples`, la carpeta de ejemplos `Expense Report` incluye una subcarpeta para archivos XML, otra para archivos XSL y otra para archivos de esquema.



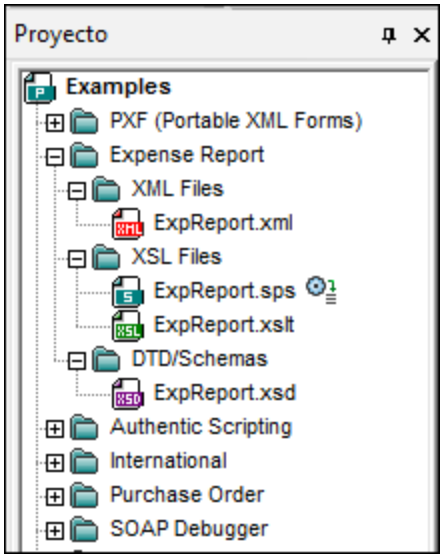
Por tanto, los proyectos permiten reunir varios archivos que se usan en conjunto y acceder a ellos más rápido. Además, puede definir esquemas y archivos XSLT para cada carpeta, lo cual permite procesar por lotes los archivos guardados en esa carpeta.

Esta sección incluye instrucciones para crear y editar proyectos[1061] y explica cómo usarlos[1066].

© 2016-2022 Altova GmbH

21.1 Crear y editar proyectos

Los proyectos se gestionan en la ventana Proyecto 122 (*imagen siguiente*) y desde el menú Proyecto 1298. Sólo se puede abrir un proyecto a la vez en la aplicación y el proyecto que está abierto aparece en la ventana Proyecto 122.



Crear proyectos nuevos y abrir proyectos

Para crear un proyecto nuevo utilice el comando de menú **Proyecto | Proyecto Nuevo**. Para abrir un proyecto utilice el comando de menú **Proyecto | Abrir proyecto**. Si crea un proyecto nuevo o abre un proyecto, este sustituye al proyecto que estaba abierto en la ventana Proyecto. Si el proyecto que estaba abierto contiene cambios sin guardar (lo cual se indica por medio de un asterisco junto al nombre del proyecto, como en la imagen siguiente), la aplicación le pregunta si desea guardar los cambios.

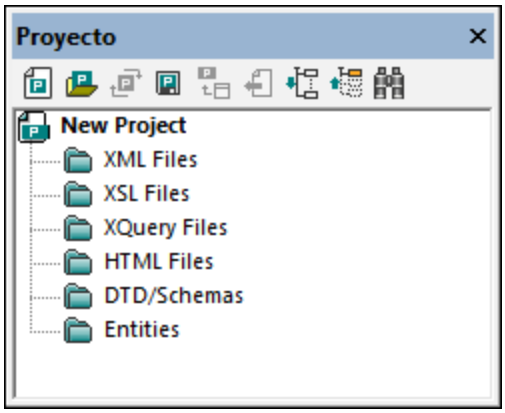
Guardar proyectos y darles un nombre

Para dar un nombre al proyecto es necesario guardarlo. Para guardar el proyecto utilice el comando **Proyecto | Guardar el proyecto** y utilice la extensión de archivo `.spp`. Cuando realice cambios en un proyecto es necesario guardar el proyecto para que los cambios se guarden. Tenga en cuenta que el nombre del proyecto (el de la carpeta de nivel superior de la ventana Proyecto) no se puede cambiar desde la IGU, sino que debe cambiarse en el explorador de archivos de Windows. El nombre de las demás carpetas del proyecto sí se pueden guardar desde la IGU.

Estructura de los proyectos

Los proyectos tienen una estructura en forma de árbol, compuesta por carpetas y archivos. Las carpetas y archivos se pueden crear en cualquier nivel de la estructura y pueden tener un número ilimitado de subniveles. Para agregar carpetas, archivos o recursos al proyecto, seleccione una carpeta de la ventana Proyecto y use los comandos del menú **Proyecto**. También puede hacer clic con el botón derecho en la carpeta y elegir un comando del menú contextual. Las carpetas, los archivos y los recursos del proyecto se pueden eliminar o arrastrar y colocar en otra posición del proyecto.

Al crear un proyecto nuevo, la estructura predeterminada del proyecto organiza el contenido del proyecto según el tipo de archivo (XML, XSL, etc.) (*imagen siguiente*).



Las extensiones de archivo asociadas a una carpeta vienen definidas por las propiedades de la carpeta. Cuando se añade un archivo a una carpeta, este se añade a la subcarpeta pertinente en función a su extensión de archivo. Puede definir en cada carpeta las extensiones de archivo asociadas a ella.

¿Qué se puede agregar a un proyecto?

Puede agregar carpetas, archivos y otros recursos tanto a la carpeta de proyecto de nivel superior como a las otras carpetas del proyecto. Hay tres tipos de carpetas: (i) carpetas de proyecto, (ii) carpetas externas y (iii) carpetas web externas.

Para agregar un objeto, seleccione la carpeta pertinente y el comando correspondiente del menú **Proyecto** o del menú contextual de la carpeta seleccionada. Estos son los objetos que puede agregar a una carpeta de proyecto:

- *Carpetas de proyecto* (en verde): estas carpetas sirven para ordenar la estructura del contenido de un proyecto. Puede definir las extensiones de archivo asociadas a la carpeta de proyecto (en las propiedades de la carpeta). Cuando se agregan archivos a una carpeta, estos se agregan automáticamente a la subcarpeta que tiene asociada la extensión de ese tipo de archivo. Por consiguiente, cuando añada varios archivos a una carpeta, estos se distribuyen en función de su tipo de extensión en las subcarpetas correspondientes.
- *Carpetas externas* (en amarillo): son carpetas ubicadas en un sistema de archivos. Cuando se agrega una carpeta externa a una carpeta, se añaden al proyecto la carpeta externa y todos los archivos, subcarpetas y archivos de subcarpeta que contiene. Puede definir extensiones de archivo en una carpeta externa para filtrar los archivos disponibles en el proyecto.
- *Carpetas web externas*: son carpetas externas ubicadas en un servidor web y requieren que el usuario se identifique para poder acceder a su contenido. Puede definir extensiones de archivo en carpetas web externas para filtrar los archivos disponibles en el proyecto.
- *Archivos*: para agregar archivos a una carpeta seleccione la carpeta y seleccione uno de estos tres comandos: (i) **Agregar archivos**, para seleccionar los archivos desde el cuadro de diálogo "Abrir"; (ii) **Agregar archivo activo**, para agregar el archivo que está activo en la ventana principal; o (iii) **Agregar archivos activos y relacionados**, para agregar el archivo activo y todos los archivos relacionados, como su esquema XML o su DTD. Recuerde que los archivos asociados por medio de instrucciones de procesamiento (como archivos XSLT, por ejemplo) no se consideran archivos relacionados en este caso.

© 2016-2022 Altova GmbH

- *Recursos globales*: son alias para recursos de archivo, carpeta y base de datos. Para más información sobre cómo definir estos recursos y cómo usarlos consulte la sección Recursos globales[1042].
- *Direcciones URL*: identifican un objeto mediante una dirección URL.
- *Un proyecto de script de Altova*: un archivo `.asprj` que se puede asignar a un proyecto de XMLSpy. Esto pone a disposición del proyecto macros y otros scripts. Para más información sobre cómo crear proyectos de script y asignarlos a los proyectos de XMLSpy consulte la sección Scripting[1641].

Propiedades de los proyectos y las carpetas

Las propiedades (como el esquema de validación y el archivo XSLT para las transformaciones) se pueden definir tanto para el proyecto completo como para carpetas individuales. Una vez definidas puede realizar las acciones de validar o transformar a nivel del proyecto o de una sola carpeta. Para ejecutar una acción haga clic con el botón derecho en el proyecto o la carpeta y seleccione la acción que prefiera en el menú contextual que aparece.

Las carpetas guardan sus propiedades en el cuadro de diálogo "Propiedades", al que puede acceder seleccionando la carpeta y usando el comando de menú **Proyecto | Propiedades** (o desde el comando Propiedades del menú contextual de esa carpeta). Estas son las propiedades que se pueden definir y editar en el cuadro de diálogo "Propiedades":

- *Nombre de la carpeta*: el nombre de la carpeta de nivel superior no se puede editar (en su caso aparece la ruta de acceso del archivo y no el nombre de la carpeta).
- *Extensiones de archivo*: las extensiones de archivo de la carpeta de nivel superior no se pueden editar. Se trata de una lista de extensiones separadas por punto y coma (por ejemplo **`xml;svg;wml`**). Esta lista determina qué archivos se añaden a esa carpeta cuando se añaden archivos al proyecto. Por ejemplo, si añade archivos activos y relacionados a un proyecto, su extensión de archivo determina en qué carpeta debe ir cada uno de ellos.
- *Validación*: indica qué archivo XML o DTD se debe usar para validar los archivos XML de una carpeta.
- *Transformaciones*: indica (i) qué archivos XSLT se deben usar para transformar los archivos XML de la carpeta y (ii) qué archivos XML se deben transformar con los archivos XSLT de la carpeta.
- *Archivos de destino*: para los documentos de salida de las transformaciones. Indica la extensión de archivo y la carpeta donde se deben guardar los documentos de salida.
- *Archivos SPS para la vista Authentic*: indica qué archivos SPS se deben usar para que los archivos XML de la carpeta se puedan ver y editar en la vista Authentic.

Es importante tener en cuenta que:

- Al definir una propiedad en una carpeta se sobrescribe la misma propiedad en el proyecto.
- Al definir una propiedad en el proyecto, esta se aplica a todas las carpetas para las que no se haya definido esa misma propiedad.
- Si se ejecuta una acción en un proyecto, esta se aplica a todos los tipos de archivo pertinentes en todas las carpetas del proyecto. Por ejemplo, si valida un proyecto, esa validación se aplica a todos los archivos XML que haya en todas las carpetas del proyecto. En este caso, el esquema que se ha definido para el proyecto se usa para validar todos los archivos excepto para los archivos XML que se encuentren en carpetas cuyas propiedades se hayan configurado para que usen otro esquema de validación.

Consulte la descripción del comando **Proyecto | Propiedades**[1328] para obtener más información.

Control de código fuente en los proyectos

En los proyectos se pueden usar los sistemas de control de código fuente que son compatibles con Microsoft Visual Source-Safe. Para más información sobre esta característica consulte la *Comandos de menú*[1302] del manual.

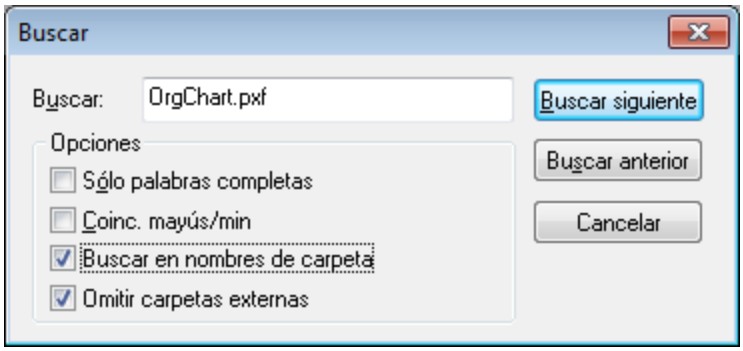
Guardar los proyectos

Si realiza cambios en el proyecto (p. ej. si añade o elimina un archivo o si cambia una propiedad del proyecto), estos cambios se deben guardar haciendo clic en el comando **Guardar el proyecto**.

Buscar en los proyectos

Puede buscar carpetas y archivos de proyectos usando su nombre completo o solo una parte. Si la búsqueda obtiene resultados, los archivos y carpetas encontrados se resaltan uno por uno.

Para iniciar una búsqueda, haga clic en la ventana Proyecto y seleccione el comando **Edición | Buscar** (o pulse **Ctrl+F**). Aparece el cuadro de diálogo "Buscar" (*imagen siguiente*) donde puede escribir la cadena de texto que desea buscar y donde puede seleccionar otras opciones de búsqueda (explicadas más abajo) dependiendo de lo que busque.



Estas son las opciones de búsqueda del cuadro de diálogo:

- *Sólo palabras completas*: limita la búsqueda porque la cadena debe coincidir con una palabra completa en el nombre del archivo o de la carpeta. En el nombre de archivo, el texto situado antes y después del punto (el punto excluido) se considera una palabra completa.
- *Coincidencia de mayúsculas y minúsculas*: puede especificar que las mayúsculas y minúsculas de la cadena de búsqueda coincidan exactamente con las del nombre del archivo o de la carpeta.
- *Buscar en nombres de carpeta:* puede realizar la búsqueda en nombres de archivo y carpetas o sólo en nombres de archivo.
- Puede *omitir las carpetas externas*[1319] de la búsqueda. Las carpetas externas son carpetas ubicadas en el sistema de archivos o en la red, a diferencia de las carpetas de proyecto, que se crean en el proyecto y no en el sistema.

Si la búsqueda obtiene resultados, el primer resultado de la búsqueda se resalta en la ventana Proyecto. Para ver el resto de resultados, haga clic en los botones **Buscar siguiente** y **Buscar anterior** del cuadro de diálogo "Buscar".

© 2016-2022 Altova GmbH

Volver a cargar los proyectos

Si se realizan cambios en una carpeta externa, estos cambios no se reflejan en la ventana Proyecto a no ser que vuelva a cargar el proyecto.

21.2 Usar proyectos

Los proyectos son muy útiles porque sirven para organizar su espacio de trabajo, aplicar opciones de configuración a varios archivos y ejecutar comandos de procesos por lotes. Por tanto, los proyectos pueden ayudarle a hacer su trabajo más rápido y con mayor eficacia. Para más información consulte Crear y editar proyectos[1061] y la descripción de la ventana Proyecto[122].

Ventajas de usar proyectos

Estas son algunas ventajas de usar proyectos:

- Los archivos y las carpetas se pueden agrupar en carpetas según la extensión de archivo o cualquier otro criterio.
- Puede asignar esquemas y archivos XSLT a una carpeta. Así podrá validar o transformar un archivo XML usando esquemas y archivos XSLT diferentes. Para ello basta con añadir el mismo archivo XML a carpetas diferentes y definir un esquema y archivo XSLT diferente para cada carpeta.
- Puede procesar por lotes las carpetas. Más abajo puede ver la lista de comandos que se pueden procesar por lotes.
- Puede indicar carpetas de destino para los documentos de salida de las transformaciones.

Organizar recursos para acceder a ellos con rapidez

Puede organizar sus recursos de archivo y carpeta en una estructura en forma de árbol para tener una visión de conjunto de las carpetas y los archivos de su proyecto y poder acceder a ellos de forma rápida y sencilla. Para abrir un archivo basta con hacer doble clic en el nombre del archivo en la ventana Proyecto. También puede agregar y eliminar carpetas y archivos igualmente rápido. Si desea trabajar con otro proyecto, cierre el proyecto que está abierto en la ventana Proyecto y abra el otro proyecto.

Procesamiento por lotes

Los comandos para procesar por lotes los archivos de una carpeta, ya sea la carpeta de nivel superior o una subcarpeta, **están en el menú contextual de la carpeta** (clic con el botón derecho en la carpeta). Los pasos para ejecutar comandos de procesos por lotes son:

1. Defina los archivos que desea usar para la validación y transformación de los archivos en el cuadro de diálogo "Propiedades" de la carpeta.
2. Indique en qué carpeta se deben guardar los resultados de la transformación. Si no se indica una carpeta de destino, entonces se usa la carpeta de destino de la carpeta del nivel inmediatamente superior.
3. Use los comandos del menú contextual para la ejecución de lotes. Si por el contrario selecciona el mismo comando en los menús **XML**, **DTD/Esquema** y **XSL/XQuery**, el comando se ejecuta sólo en el documento que está activo en la ventana principal y no en todos los archivos de la carpeta de proyecto.

Los comandos del menú contextual de las carpetas de proyecto que se pueden ejecutar por lotes son:

- *Comprobar formato XML*: si se detecta un error durante la ejecución por lotes, este se notifica en la ventana Mensajes.
- *Validar el documento XML:* si se detecta un error durante la ejecución por lotes, este se notifica en la ventana Mensajes.

© 2016-2022 Altova GmbH

- *Transformación*: los resultados de la transformación se guardan en la carpeta de destino indicada en el cuadro de diálogo "Propiedades" de la carpeta. Si no se indicó ninguna carpeta de destino, entonces se usa la carpeta de destino de la carpeta del nivel inmediatamente superior. Si ninguna de las carpetas de niveles superiores tiene definida una carpeta de destino, se abre una ventana de documento en la que van apareciendo los resultados de cada transformación. La transformación XSL:FO transforma documentos XML o FO en PDF.
- *Generar DTD o esquema*: antes de generar los esquemas la aplicación solicita una carpeta de destino. Los esquemas se generan en la carpeta de destino y se muestran en ventanas de documento nuevas.

Nota: para ejecutar comandos de procesos por lotes use el menú de contexto de la carpeta en la ventana Proyecto. No use los comandos de los menús **XML**, **DTD/Esquema** y **XSL/XQuery**. Estos comando se ejecutan en el documento que está activo en la ventana principal.

Validación y transformaciones XSLT/XQuery con RaptorXML Server

Los menús contextuales de la carpeta de proyecto incluyen comandos para trabajar con RaptorXML y realizar validaciones de datos XML y transformaciones XSLT/XQuery de alto rendimiento. Consulte la sección RaptorXML Server[1068] para obtener más información.

22 RaptorXML(+XBRL) Server

Si tiene Altova RaptorXML(+XBRL) Server (en adelante también llamado RaptorXML Server, RaptorXML o Raptor) instalado en la red, cuenta con una licencia válida y XMLSpy tiene acceso a esta aplicación servidor, podrá usar RaptorXML Server para validar documentos XML y XBRL* y ejecutar transformaciones XSLT/XQuery[1091]. Puede validar el documento activo o todos los documentos de la carpeta de proyecto de XMLSpy. Los resultados de la validación aparecen en la ventana Mensajes de la interfaz gráfica.

En XMLSpy se pueden (i) validar documentos, (ii) ejecutar transformaciones XSLT/XQuery o (iii) ejecutar documentos XULE (o conjuntos de documentos) en una instancia XBRL. La principal ventaja de usar RaptorXML para estas tareas es que permite configurar las validaciones o ejecuciones una a una con diferentes opciones de validación. Además, en XMLSpy puede almacenar sus opciones para RaptorXML en una *configuración* y seleccionarla más adelante para realizar validaciones con RaptorXML. Por último, recomendamos utilizar RaptorXML para validar colecciones de datos de gran tamaño.

Nota: el rendimiento real depende del número de núcleos de procesador utilizados por RaptorXML para la validación. Cuanto mayor sea el número de núcleos, más rápido será el procesamiento.

***Nota:** existen dos ediciones de Raptor: *RaptorXML Server* (para validaciones XML) y *RaptorXML+XBRL Server* (para validaciones XML y XBRL). Si quiere validar documentos XBRL, debe utilizar RaptorXML+XBRL Server. Encontrará más información sobre esta edición en el sitio web de Altova y en los manuales de usuario de *RaptorXML Server* y *RaptorXML+XBRL Server*.

Nota: RaptorXML Server no puede utilizarse con la mayoría de los proxies HTTP porque no admite actualizaciones de websocket. Si tiene este problema, añada el host de RaptorXML Server a la lista de omitidos del proxy si todavía no se omite.

Pasos para validar o transformar datos con RaptorXML Server

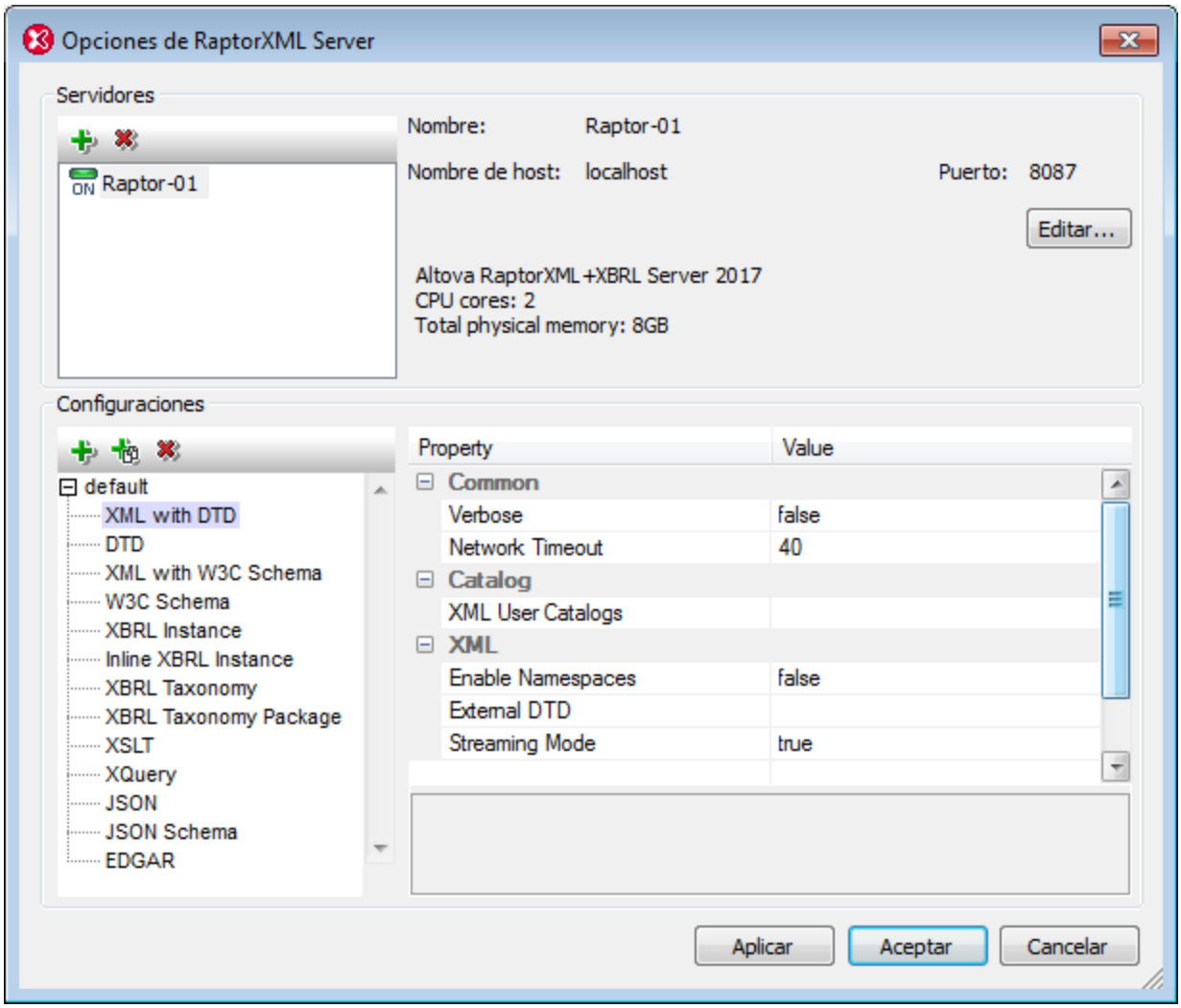
Para validar documentos XML/XBRL o ejecutar transformaciones XSLT/XQuery con RaptorXML, XMLSpy debe conocer qué servidor RaptorXML debe utilizar, cómo tener acceso al servidor y qué opciones de validación/transformación debe pasarle. Esta información se gestiona así:

1. *Primero se añade un servidor al repertorio de servidores RaptorXML.*[1069] Es decir, se añaden servidores RaptorXML a un repertorio y los datos de acceso para cada uno de ellos se almacenan en XMLSpy. Cada servidor tiene un nombre distinto.
2. *Después se definen configuraciones para cada servidor.*[1070] Una configuración es un conjunto de opciones de validación de Raptor y un servidor puede tener varias configuraciones distintas. Para cada sesión de validación el usuario elige una configuración, que es la configuración activa.
3. *Después se selecciona la configuración de servidor para la validación.*[1073] El usuario selecciona un servidor y una de sus configuración. La configuración activa se utiliza en todas las validaciones siguientes que usen Raptor.
4. *Por último se ejecuta la validación*[1073] *o se ejecuta la transformación XSLT/XQuery*[1091] *con Raptor*.

© 2016-2022 Altova GmbH

22.1 Agregar sevidores y configuraciones de servidor

En el cuadro de diálogo "Opciones de RaptorXML Server" (que se abre con **Herramientas | Gestionar servidores Raptor**[1559]) puede agregar servidores Raptor[1069] al repertorio de servidores Raptor disponibles y después definir configuraciones[1070] para cada servidor. Los servidores que se añadan desde este cuadro de diálogo, así como sus configuraciones, aparecerán en el submenú **Herramientas | Servidores Raptor y configuraciones**[1563]. En este submenú puede seleccionar la configuración de servidor que se debe usar para la validación con Raptor.



Agregar servidores Raptor

En el panel *Servidores* del cuadro de diálogo (*imagen siguiente*) haga clic en el icono **Agregar servidor** y escriba el nombre con el quiere identificar al servidor Raptor, el nombre de red del equipo donde está instalado Raptor (el nombre de host) y el puerto. Haga clic en **Aceptar** para guardar la configuración.



- *Nombre:* cualquier cadena de texto. Se usa en XMLSpy para identificar un servidor Raptor concreto.
- *Nombre de host:* el nombre o dirección IP del equipo de red donde está instalado el servidor Raptor. El procesamiento será más rápido si utiliza una dirección IP en lugar de un nombre de host. La dirección IP que corresponde a `localhost` (el equipo local) es **`127.0.0.1`**.
- *Puerto:* el puerto por el que se accede al servidor Raptor. Este puerto se especifica en el archivo de configuración de Raptor (llamado `server_config.xml`). El puerto debe ser fijo y debe conocerse para que las solicitudes se puedan dirigir correctamente al servicio. Para más información consulte los manuales del usuario de *RaptorXML Server* y *RaptorXML+XBRL Server*.

Haga clic en **Aceptar** cuando termine de introducir la información del servidor. El nombre de servidor aparecerá en la lista de servidores (en el panel izquierdo). Además, el icono verde situado junto al nombre del servidor indica que ese servidor Raptor está encendido y en ejecución. Los datos del servidor aparecen en el panel. Si el icono es de color rojo, esto significa que el servidor no está conectado. Por otra parte, si no se encuentra el servidor, entonces aparece un mensaje de error.

Nota: el servidor Raptor debe estar en ejecución cuando se añada. De lo contrario, XMLSpy no podrá obtener ni almacenar información sobre el servidor. Si, tras ser añadido, el servidor se desconecta o no se encuentra, el icono situado junto al nombre del servidor cambiará de color y aparecerá un mensaje de error.

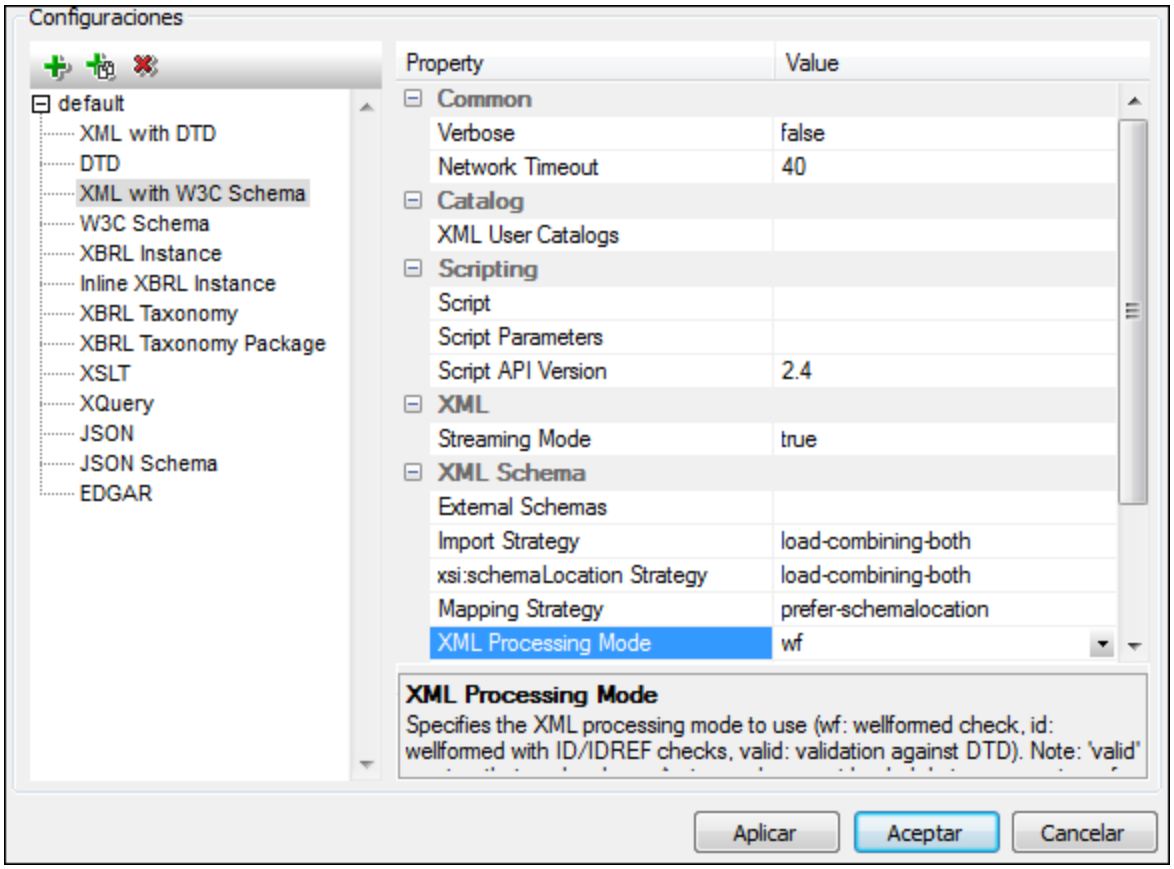
Para editar el nombre, nombre de host o puerto de un servidor basta con seleccionar el servidor en el panel izquierdo y hacer clic en el botón **Editar**. Esto abre un cuadro de diálogo donde puede editar los datos. Para quitar un servidor del repertorio basta con seleccionarlo y hacer clic en el icono **Quitar servidor seleccionado**.

Configuraciones de servidor

Una configuración es un conjunto de opciones de validación de RaptorXML. Cuando se añade un servidor, se añade con la configuración predeterminada `default`. Se trata de un conjunto de opciones RaptorXML que tienen valores predeterminados. Estos valores pueden editarse y también se pueden añadir nuevas configuraciones con otros valores. Tras definir las configuraciones de servidor podrá seleccionar una configuración como configuración activa. Esta configuración activa es la que se utilizará cuando se ejecute el comando **Validar en servidor**.

El panel *Configuraciones* está dividido en dos partes: (i) el lateral izquierdo muestra las configuraciones y los tipos de documentos que se pueden validar y (ii) el lateral derecho muestra todas las opciones del tipo de validación que está seleccionada en el lateral izquierdo. En la parte inferior del cuadro de diálogo encontrará una descripción de la opción seleccionada en el lateral derecho.

 © 2016-2022 Altova GmbH



Agregar una configuración
Haga clic en **Agregar una configuración** en el panel *Configuraciones* del cuadro de diálogo "Opciones de RaptorXML Server". Esto añade una configuración nueva cuyas opciones tienen valores predeterminados. También puede crear una configuración nueva haciendo clic en Copiar configuración seleccionada. Esto crea una configuración nueva cuyas opciones tienen los mismos valores que la configuración que se copió. Las configuraciones nuevas se crean con nombres predeterminados como `config<X>`. Para cambiar el nombre de una configuración haga doble clic en ella y escriba el nombre nuevo. Después puede editar los valores de las opciones de la configuración.

Editar valores de las opciones de una configuración
Primero seleccione el documento que se debe validar en el panel izquierdo. Esto muestra las opciones para ese tipo de validación en el panel derecho. Siga estos pasos para editar el valor de una opción:

- Si el valor puede ser un valor de un conjunto de valores predefinidos, seleccione el valor en el cuadro combinado de esa opción.
- Si el valor no está restringido, haga clic en el campo de la opción y escriba el valor que desea utilizar.
- Si el valor es una ruta de acceso de archivo, no solo puede introducir el valor sino que además puede buscar el archivo que desea usar haciendo clic en el botón **Examinar** de la opción.

Cuando seleccione una opción su descripción aparecerá en el recuadro situado al final del cuadro de diálogo. Consulte los manuales de RaptorXML Server y RaptorXML(+XBRL) Server para obtener más información sobre las diferentes opciones de RaptorXML.

Quitar una configuración
Seleccione la configuración en el panel izquierdo y después haga clic en **Quitar configuración seleccionada**.

XMLSpy en Visual Studio y Eclipse

Si se integra XMLSpy en Visual Studio[1123] y Eclipse[1129], la configuración activa en estos entornos IDE será la configuración que esté activa en la versión independiente de XMLSpy.

© 2016-2022 Altova GmbH

22.2 Validación de datos con RaptorXML Server

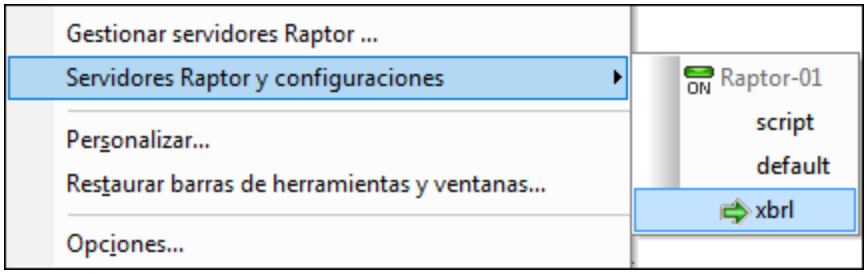
Con RaptorXML Server puede validar documentos XML y XBRL*. La validación tiene lugar en dos pasos:

- Primero se selecciona el servidor y la configuración que se deben usar para la validación.
- Después se ejecuta la validación (con uno de los comandos **Validar en el servidor**).

***Nota:** existen dos ediciones de Raptor: *RaptorXML Server* (para validaciones XML) y *RaptorXML+XBRL Server* (para validaciones XML y XBRL). Si quiere validar documentos XBRL, debe utilizar RaptorXML+XBRL Server. Encontrará más información sobre esta edición en el sitio web de Altova y en los manuales de usuario de *RaptorXML Server* y *RaptorXML+XBRL Server*.

Seleccionar la configuración de servidor

Si definió varias configuraciones en varios servidores, podrá seleccionar un servidor y una de sus configuraciones como configuración activa. La configuración activa se usará para todas las validaciones siguientes. Pase el puntero por encima del comando **Herramientas | Servidores Raptor y configuraciones** (*imagen siguiente*) para abrir el submenú que contiene todos los servidores que se añadieron, así como sus configuraciones. Seleccione la configuración de servidor que desea usar como configuración activa. Por ejemplo, en la imagen siguiente, se seleccionó la configuración `xbrl` del servidor `Raptor-01`. La fecha verde indica que se trata de la configuración activa.



Validar documentos con RaptorXML Server

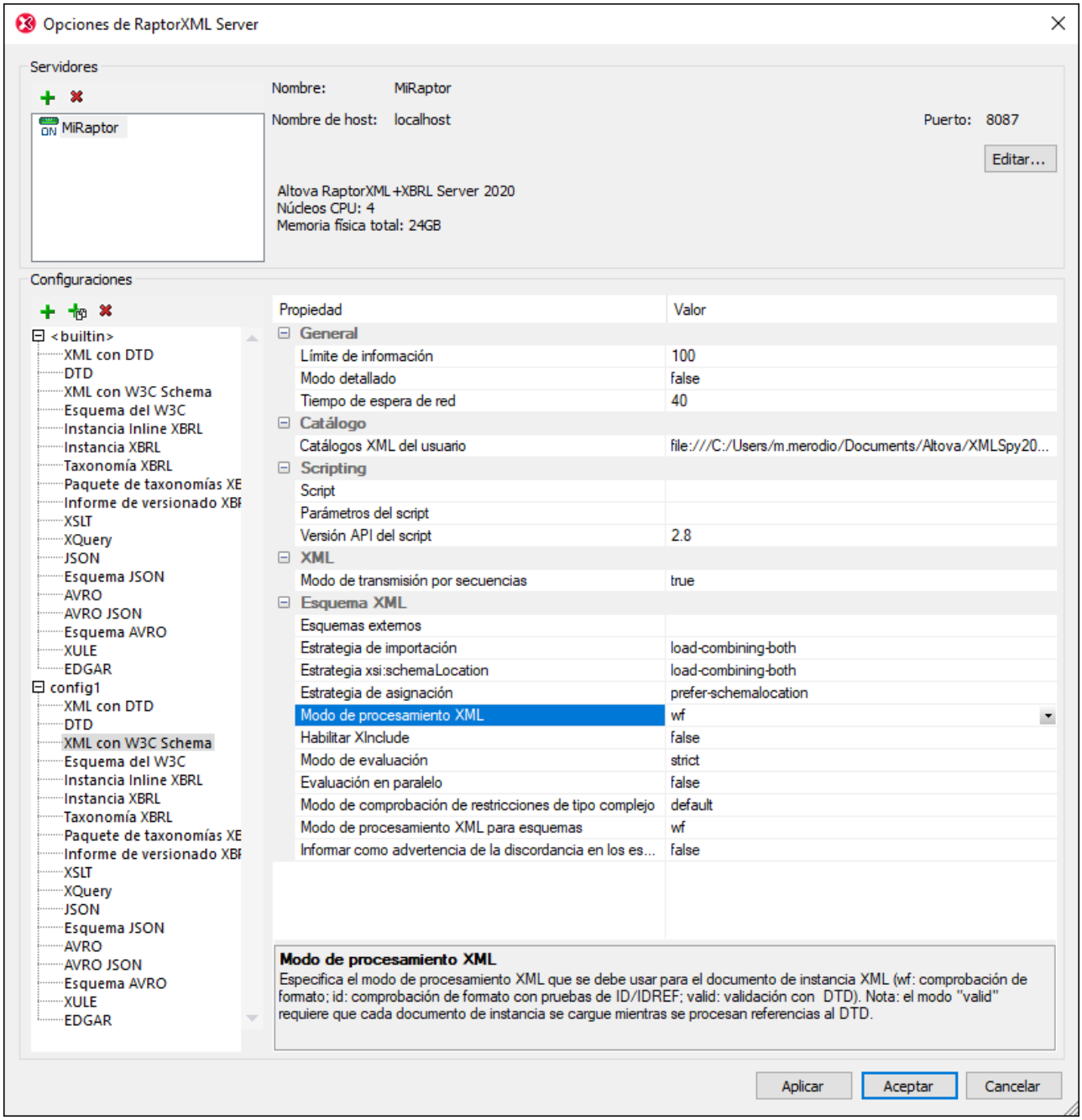
Puede validar documentos XML y XBRL con los motores de validación integrados de XMLSpy o con RaptorXML Server. Hay varias maneras de validar documentos con RaptorXML:

- Con el icono **Validar en el servidor** de la barra de herramientas.
- Con el comando **XML | Validar XML en el servidor (alto rendimiento) (Ctrl+F8)**
- En el ayudante de entrada Proyecto haga clic con el botón derecho en el proyecto, en una carpeta o en un archivo y después seleccione **Validar XML en el servidor (alto rendimiento)** en el menú contextual para validar los datos XML o XBRL del objeto seleccionado.

Nota: la validación con RaptorXML se puede iniciar desde las vistas Texto, Esquema, XBRL y Cuadrícula.

22.3 Opciones de validación

Los subapartados siguientes corresponden a los diferentes tipos de documentos que se pueden validar. Por ejemplo, *XML con W3C Schema* valida un documento XML con un esquema XML del W3C. Cuando seleccione un tipo de validación en el panel izquierdo, las opciones de validación de RaptorXML que corresponden a ese tipo de validación aparecen en el panel derecho, agrupadas en categorías. Tenga en cuenta que no todos los grupos que aparecen en la imagen (como los grupos XBRL) están disponibles en la edición Professional.



 © 2016-2022 Altova GmbH

Los subapartados siguientes incluyen enlaces a la documentación de las opciones de validación de RaptorXML Server.

22.3.1 Opciones comunes

Estas opciones son compartidas por todos los tipos de validación.

- General

  Límite de información
  Modo detallado
  Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

22.3.2 XML con DTD

Opciones para validar datos XML con una DTD.

- General

  Límite de información
  Modo detallado
  Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

- XML

  Habilitar espacios de nombres
  DTD externos
  Modo de transmisión por secuencias

22.3.3 DTD

Opciones para validar documentos DTD.

- General

Límite de información
Modo detallado
Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

- Scripting

  Script
  Parámetros del script
  Versión API del script

22.3.4 XML con W3C Schema

Opciones para validar datos XML con esquemas XML.

- General

  Límite de información
  Modo detallado
  Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

- Scripting

  Script
  Parámetros del script
  Versión API del script

- XML

  Modo de transmisión por secuencias

- Esquema XML

  Esquemas externos (xsd)
  Estrategia de importación (schema-imports)
  Estrategia xsi:schemaLocation (schemalocation-hints)
  Estrategia de asignación (schema-mapping)

© 2016-2022 Altova GmbH

Modo de procesamiento XML (xmlmode)
Habilitar XInclude (xinclude)
Modo de evaluación
Evaluación en paralelo
Complex Type Restriction Check Mode
XML Processing Mode for Schemas
Report Import Namespace Mismatch as Warning

22.3.5 W3C Schema

Opciones para validar esquemas XML.

- General

  Límite de información
  Modo detallado
  Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

- Scripting

  Script
  Parámetros del script
  Versión API del script

- Esquema XML

  Estrategia de importación (schema-imports)
  Estrategia xsi:schemaLocation (schemalocation-hints)
  Estrategia de asignación (schema-mapping)
  Modo de procesamiento XML (xmlmode)
  Habilitar XInclude (xinclude)
  Modo de comprobación de restricciones de tipo complejo
  Modo de procesamiento XML para esquemas
  Informar como advertencia de espacios de nombres que no coinciden en la importación

22.3.6 Instancia inline XBRL

Opciones para validar documentos de Inline XBRL.

- General
  - Límite de información
  - Modo detallado
  - Tiempo de espera de red
- Catálogo
  - Catálogo XML del usuario
- Scripting
  - Script
  - Parámetros del script
  - Versión API del script
- Esquema XML
  - Estrategia de importación (schema-imports)
  - Estrategia xsi:schemaLocation (schemalocation-hints)
  - Estrategia de asignación (schema-mapping)
  - Habilitar XInclude (xinclude)
  - Evaluación en paralelo
  - Modo de comprobación de restricciones de tipo complejo
  - Informar como advertencia de espacios de nombres que no coinciden en la importación
- XBRL
  - Habilitar extensión XBRL Dimensions (dimensions)
  - Habilitar extensión Extensible Enumerations (extensible-enumerations)
  - Habilitar extensión Unit Registry
  - Carga previa de los esquemas de la especificación XBRL (preload-xbrl-schemas)
  - Paquetes de taxonomía
  - Archivo de configuración de los paquetes de taxonomía
  - Validar solo DTS referenciado
  - Tratar incoherencias XBRL como errores (treat-inconsistencies-as-errors)
  - Archivo del registro de unidades
  - Compatibilidad con el estado dado en el registro de unidades
  - Punto de entrada DTS añadido
  - Estrategia de transformación URI (en documentos de salida)
  - Informar de incoherencias relacionadas con el ítem de suma
  - Informar de incoherencias relacionadas con essence-alias
  - Informar de incoherencias relacionadas con requires-element
  - Habilitar la extensión de etiqueta genérica preferida
  - Habilitar la extensión Generic Links

© 2016-2022 Altova GmbH

Deduplicar
Informar de duplicados
Informar de la gravedad de los duplicados

- Inline XBRL

Versión de Inline XBRL (ixbrl-version)
Registro de transformaciones de Inline XBRL (transformation-registry)
Tratar argumentos como un conjunto de documentos de Inline XBRL (document-set)
Habilitar la validación del documento de destino (validate-xbrl)
Archivo de salida del documento de destino (xbrl-output)
Normalización de espacios en blanco no numéricos
Normalización extendida de espacios en blanco

- Fórmulas XBRL

Habilitar extensión XBRL Formula (formula)
Habilitar extensión Assertion Severity (assertion-severity)
Carga previa de los esquemas de la especificación Formula (preload-formula-schemas)
Informar de evaluaciones de aserción no cumplidas
Idioma del mensaje de validación (message-lang)
Rol del mensaje de validación (message-role)
Archivo Fórmulas-que-ignorar
Archivo Fórmulas-que-procesar
Archivo Aserciones-que-ignorar
Archivo Aserciones-que-procesar
Fórmulas-que-ignorar
Aserciones-que-ignorar
Resultado de la evaluación de la fórmula
Habilitar optimización de fórmula

- Tablas XBRL

Habilitar extensión XBRL Table (table)
Carga previa de los esquemas de la especificación XBRL Table (preload-table-schemas)
Espacio de nombres de la base de enlaces de tabla
Orden de tabla nodoAspecto

- XULE para XBRL

XULE
Tamaño de la pila XULE
Enlaces de espacios de nombres de instancia XULE
Reglas XULE que procesar
Informar de evaluaciones de reglas XULE
Archivo de salida XULE

22.3.7 Instancia XBRL

Opciones para validar documentos de instancia XBRL.

- General
  - Límite de información
  - Modo detallado
  - Tiempo de espera de red
- Catálogo
  - Catálogo XML del usuario
- Scripting
  - Script
  - Parámetros del script
  - Versión API del script
- Esquema XML
  - Estrategia de importación (schema-imports)
  - Estrategia xsi:schemaLocation (schemalocation-hints)
  - Estrategia de asignación (schema-mapping)
  - Habilitar XInclude (xinclude)
  - Evaluación en paralelo
  - Modo de comprobación de restricciones de tipo complejo
  - Informar como advertencia de espacios de nombres que no coinciden en la importación
- XBRL
  - Habilitar extensión XBRL Dimensions (dimensions)
  - Habilitar extensión Extensible Enumerations (extensible-enumerations)
  - Habilitar extensión Unit Registry (utr)
  - Carga previa de los esquemas de la especificación XBRL (preload-xbrl-schemas)
  - Paquetes de la taxonomía
  - Validar solamente la DTS referenciada (validate-dts-only)
  - Tratar las incoherencias de XBRL como errores (treat-inconsistencies-as-errors)
  - Archivo del registro de unidades
  - Compatibilidad con el estado dado en el registro de unidades
  - Punto de entrada DTS añadido
  - Estrategia de transformación URI (en documentos de salida)
  - Informar de incoherencias relacionadas con el ítem de suma
  - Informar de incoherencias relacionadas con essence-alias
  - Informar de incoherencias relacionadas con requires-element
  - Habilitar la extensión de etiqueta genérica preferida
  - Habilitar la extensión Generic Links
  - Deduplicar

© 2016-2022 Altova GmbH

Informar de duplicados
Informar de la gravedad de los duplicados

- Fórmulas XBRL

Habilitar extensión XBRL Formula (formula)
Habilitar extensión Assertion Severity (assertion-severity)
Carga previa de los esquemas de la especificación Formula (preload-formula-schemas)
Informar de evaluaciones de aserción no cumplidas
Idioma del mensaje de validación (message-lang)
Rol del mensaje de validación (message-role)
Archivo Fórmulas-que-ignorar
Archivo Fórmulas-que-procesar
Archivo Aserciones-que-ignorar
Archivo Aserciones-que-procesar
Fórmulas-que-ignorar
Aserciones-que-ignorar
Resultado de la evaluación de la fórmula
Habilitar optimización de fórmula

- Tablas XBRL

Habilitar extensión XBRL Table (table)
Carga previa de los esquemas de la especificación XBRL Table (preload-table-schemas)
Espacio de nombres de la base de enlaces de tabla
Orden de tabla nodoAspecto

- XULE para XBRL

XULE
Tamaño de la pila XULE
Enlaces de espacios de nombres de instancia XULE
Reglas XULE que procesar
Informar de evaluaciones de reglas XULE
Archivo de salida XULE

22.3.8 Taxonomía XBRL

Opciones para validar taxonomías XBRL.

- General

Límite de información
Modo detallado
Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

- Scripting

  Script
  Parámetros del script
  Versión API del script

- Esquema XML

  Estrategia de importación (schema-imports)
  Estrategia xsi:schemaLocation (schemalocation-hints)
  Estrategia de asignación (schema-mapping)
  Habilitar XInclude (xinclude)
  Modo de comprobación de restricciones de tipo complejo
  Informar como advertencia de espacios de nombres que no coinciden en la importación

- XBRL

  Habilitar extensión XBRL Dimensions (dimensions)
  Habilitar extensión Extensible Enumerations (extensible-enumerations)
  Carga previa de los esquemas de la especificación XBRL (preload-xbrl-schemas)
  Paquetes de la taxonomía
  Tratar las incoherencias de XBRL como errores (treat-inconsistencies-as-errors)
  Habilitar la extensión de etiqueta genérica preferida
  Habilitar la extensión de enlaces genéricos

- Fórmulas XBRL

  Habilitar extensión XBRL Formula (formula)
  Habilitar extensión Assertion Severity (assertion-severity)
  Carga previa de los esquemas de la especificación Formula (preload-formula-schemas)

- Tablas XBRL

  Habilitar extensión XBRL Table (table)
  Carga previa de los esquemas de la especificación XBRL Table (preload-table-schemas)
  Espacio de nombres de la base de enlaces de tabla
  Orden de tabla nodoAspecto

© 2016-2022 Altova GmbH

22.3.9 Paquete de taxonomía XBRL

Opciones para validar paquetes de taxonomía XBRL.

- General

  Límite de información
  Modo detallado
  Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

- Scripting

  Script
  Parámetros del script
  Versión API del script

- Esquema XML

  Estrategia de importación (schema-imports)
  Estrategia xsi:schemaLocation (schemalocation-hints)
  Estrategia de asignación (schema-mapping)

22.3.10 Informe de versiones XBRL

Opciones para validar versiones XBRL.

- General

  Límite de información
  Modo detallado
  Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

- Esquema XML

  Estrategia de importación (schema-imports)
  Estrategia xsi:schemaLocation (schemalocation-hints)
  Estrategia de asignación (schema-mapping)

Habilitar XInclude (xinclude)

22.3.11 XSLT

Opciones para validar documentos XSLT.

- General

  Límite de información
  Modo detallado
  Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

- Esquema XML

  Estrategia de importación (schema-imports)
  Estrategia xsi:schemaLocation (schemalocation-hints)
  Estrategia de asignación (schema-mapping)
  Modo de procesamiento XML (xmlmode)
  Habilitar XInclude (xinclude)

- Extensión Java

  Deshabilitar extensiones Java (javaext-disable)
  Ubicación de la extensión para códigos de barras (javaext-barcode-location)

- Extensión para gráficos

  Deshabilitar extensiones para gráficos (chartext-disable)

- Extensiones .NET

  Deshabilitar extensiones .NET (dotnetext-disable)

- XEngines en general

  Cargar XML con PSVI (load-xml-with-psvi)

- XSLT

  Versión del motor XSLT (xslt-version)

© 2016-2022 Altova GmbH

Modo de la plantilla
Punto de entrada de la plantilla

22.3.12 XQuery

Opciones para validar documentos XQuery.

- General

  Límite de información
  Modo detallado
  Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

- Esquema XML

  Estrategia de importación (schema-imports)
  Estrategia xsi:schemaLocation (schemalocation-hints)
  Estrategia de asignación (schema-mapping)
  Modo de procesamiento XML (xml-mode)
  Habilitar XInclude (xinclude)

- Extensión Java

  Deshabilitar extensiones Java (javaext-disable)
  Ubicación de la extensión para códigos de barras (javaext-barcode-location)

- Extensión para gráficos

  Deshabilitar extensiones para gráficos (chartext-disable)

- Extensiones .NET

  Deshabilitar extensiones .NET (dotnetext-disable)

- XEngines en general

  Cargar XML con PSVI (load-xml-with-psvi)

- XQuery

Versión del motor XQuery (xquery-version)
Omitir declaración XML

22.3.13 JSON

Opciones para validar documentos de instancia JSON.

- General

  Límite de información
  Modo detallado
  Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

- Validación JSON

  Deshabilitar la comprobación del formato
  Líneas JSON
  JSON con comentarios

22.3.14 JSON Schema

Opciones para validar documentos de esquema JSON.

- General

  Límite de información
  Modo detallado
  Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

- Validación JSON

  Deshabilitar la comprobación del formato

© 2016-2022 Altova GmbH

22.3.15 AVRO

Opciones para validar un bloque de datos en uno o más archivos binarios Avro con respecto al esquema Avro correspondiente de cada archivo binario.

- General

  Límite de información
  Modo detallado
  Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

22.3.16 AVRO JSON

Opciones para validar un documento JSON con respecto a un esquema AVRO.

- General

  Límite de información
  Modo detallado
  Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

22.3.17 AVRO Schema

Opciones para validar uno o más documentos de esquema Avro con respecto a la especificación del esquema Avro.

- General

  Límite de información
  Modo detallado
  Tiempo de espera de red

- Catálogo

  Catálogo XML del usuario

22.3.18 EDGAR

EDGAR (Recopilación, análisis y recuperación de datos electrónicos) es un sistema que realiza tareas automatizadas de recopilación, validación e indización de informes financieros presentados por compañías a la comisión estadounidense de bolsas y valores. Cuando se validan datos con EDGAR, RaptorXML valida el documento de instancia XBRL mediante un script EDGAR interno. Estas son las opciones de validación EDGAR.

Parámetros de script EDGAR

El script EDGAR realiza comprobaciones adicionales tal y como se establece en el manual EDGAR Filing Manual Volume II: EDGAR Filing. El script permite además especificar estos parámetros:

`CIK`	La clave CIK del solicitante
`submissionType`	El tipo de presentación EDGAR (por ejemplo: `'10-K'`)
`cikList`	Una lista de CIK separadas por comas.
`cikNameList`	Una lista de nombres oficiales de solicitantes para cada CIK de `cikList` separados por `'|Edgar|'`
`forceUtrValidation`	Si tiene el valor `true`, se fuerza la validación UTR
`edbody-url`	La ruta de acceso de `edbody.dtd` que se debe usar para validar los fragmentos HTML incrustados
`edgar-taxonomies-url`	La ruta de acceso de `edgartaxonomies.xml`, que contiene una lista de archivos de taxonomía a los que se puede hacer referencia desde la taxonomía de extensión de la compañía.

General

Límite de información
Modo detallado
Tiempo de espera de red

Catálogo

Catálogo XML del usuario

Esquema XML

Estrategia de importación (schema-imports)
Estrategia xsi:schemaLocation (schemalocation-hints)
Estrategia de asignación (schema-mapping)
Habilitar XInclude (xinclude)

© 2016-2022 Altova GmbH

Evaluación en paralelo
Modo de comprobación de restricciones de tipo complejo
Informar como advertencia de espacios de nombres que no coinciden en la importación

- XBRL

Habilitar extensión XBRL Dimensions (dimensions)
Habilitar extensión Extensible Enumerations (extensible-enumerations)
Carga previa de los esquemas de la especificación XBRL (preload-xbrl-schemas)
Paquetes de la taxonomía
Tratar las incoherencias de XBRL como errores (treat-inconsistencies-as-errors)
Archivo del registro de unidades
Compatibilidad con el estado dado en el registro de unidades
Punto de entrada DTS añadido
Estrategia de transformación URI (en documentos de salida)
Informar de incoherencias relacionadas con el ítem de suma
Informar de incoherencias relacionadas con essence-alias
Informar de incoherencias relacionadas con requires-element
Habilitar la extensión de etiqueta genérica preferida
Habilitar la extensión Generic Links
Deduplicar
Informar de duplicados
Informar de la gravedad de los duplicados

- Fórmulas XBRL

Habilitar extensión XBRL Formula (formula)
Habilitar extensión Assertion Severity (assertion-severity)
Carga previa de los esquemas de la especificación Formula (preload-formula-schemas)
Informar de evaluaciones de aserción no cumplidas
Idioma del mensaje de validación (message-lang)
Rol del mensaje de validación (message-role)
Archivo Fórmulas-que-ignorar
Archivo Fórmulas-que-procesar
Archivo Aserciones-que-ignorar
Archivo Aserciones-que-procesar
Fórmulas-que-ignorar
Aserciones-que-ignorar
Resultado de la evaluación de la fórmula
Habilitar optimización de fórmula

- Tablas XBRL

Habilitar extensión XBRL Table (table)
Carga previa de los esquemas de la especificación XBRL Table (preload-table-schemas)
Espacio de nombres de la base de enlaces de tabla
Orden de tabla nodoAspecto

- XULE para XBRL

XULE
Tamaño de la pila XULE
Enlaces de espacios de nombres de instancia XULE
Reglas XULE que procesar
Informar de evaluaciones de reglas XULE
Archivo de salida XULE

© 2016-2022 Altova GmbH

22.4 Transformaciones XSLT/XQuery con RaptorXML Server

También puede usar RaptorXML Server para ejecutar (i) transformaciones XSLT (ii) y ejecutar XQuery o actualizaciones XQuery en documentos XML. Estas acciones solamente se pueden realizar desde el proyecto[1060] y tienen varios pasos:

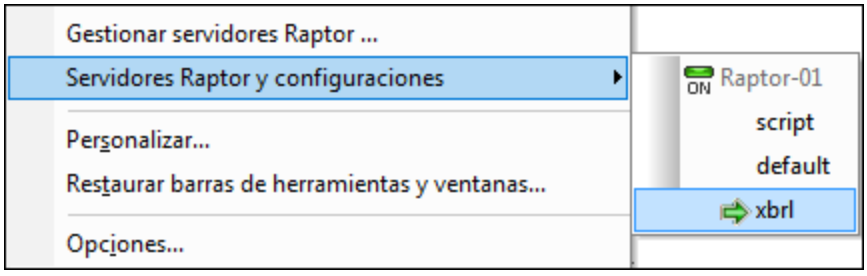
- Primero debemos seleccionar el servidor y la configuración de servidor que deseamos utilizar para la tarea.
- Después configuramos la carpeta de proyecto[1061] y especificamos qué archivos XSLT/XQuery se deben usar (en el cuadro de diálogo Propiedades del proyecto[1061]). Los archivos XSLT/XQuery que se asignan en el cuadro de diálogo Propiedades del proyecto[1061] de una carpeta son los archivos que se utilizarán para las transformaciones XSLT/XQuery de todos los archivos XML que estén en esa carpeta de proyecto. En otras palabras, no se pueden asignar archivos XSLT/XQuery a cada archivo XML de la carpeta de proyecto por separado, sino que los archivos XSLT/XQuery se asignan a toda una carpeta.
- Por último, ejecutamos la transformación XSLT o ejecutamos el código XQuery o la actualización XQuery.

Nota: si el documento XSLT o XQuery utiliza funciones de extensión Java o .NET, se usarán rutas de acceso de archivo para encontrar los archivos JAR (Java) o archivos de ensamblado externos sin registrar (.NET). Esto significa que, si se usa el mismo documento XSLT/XQuery para transformaciones/ejecuciones desde XMLSpy y desde RaptorXML, entonces las rutas de acceso de los archivos JAR o de ensamblado que están en el documento deben ser correctas.

Note: si RaptorXML está en el mismo equipo que XMLSpy, se recomienda especificar el valor true `para` la opción de servidor **`server.unrestricted-filesystem-access`**. Consulte la documentación de RaptorXML Server y el apartado sobre el archivo de configuración de RaptorXML Server.

Seleccionar la configuración de servidor que se debe usar

Si definió varias configuraciones en varios servidores, podrá seleccionar un servidor y una de sus configuraciones como configuración activa. La configuración activa se usará para todas las validaciones siguientes. Pase el puntero por encima del comando **Herramientas | Servidores Raptor y configuraciones** (*imagen siguiente*) para abrir el submenú que contiene todos los servidores que se añadieron, así como sus configuraciones. Seleccione la configuración de servidor que desea usar como configuración activa. Por ejemplo, en la imagen siguiente, se seleccionó la configuración `xbrl` del servidor `Raptor-01`. La fecha verde indica que se trata de la configuración activa.



Ejecutar transformaciones XSLT

La transformación XSLT se puede realizar con los motores XSLT de XMLSpy o con RaptorXML. Para ejecutar transformaciones XSLT con RaptorXML:

- Haga clic con el botón derecho en la carpeta de proyecto donde están los archivos XML que desea transformar. Puede ser la carpeta de proyecto entera o una carpeta del proyecto.
- En el menú contextual seleccione el comando **Transformación XSL en el servidor (alto rendimiento)**.

Nota: recuerde que no se pueden asignar archivos XSLT a cada archivo XML de la carpeta de proyecto por separado, sino que los archivos XSLT/XQuery se asignan a toda una carpeta.

Para más información consulte las secciones XSLT 490 y Transformación XSLT 1394 de este manual.

Ejecución de código XQuery y actualizaciones XQuery

Las transformaciones y actualizaciones XQuery se pueden realizar con los motores XQuery de XMLSpy o con RaptorXML. Para ejecutar transformaciones y actualizaciones XQuery con RaptorXML:

- Haga clic con el botón derecho en la carpeta de proyecto donde están los archivos XML que desea transformar o los archivos XQuery que desea actualizar. Puede ser la carpeta de proyecto entera o una carpeta del proyecto.
- En el menú contextual seleccione el comando **Ejecutar XQuery/XQuery Update en el servidor (alto rendimiento)**.

Nota: recuerde que no se pueden asignar archivos XQuery a cada archivo XML de la carpeta de proyecto por separado, sino que los archivos XSLT/XQuery se asignan a toda una carpeta.

Para más información consulte las secciones XQuery 505 y Ejecutar XQuery/XQuery Update 1400 de este manual.

© 2016-2022 Altova GmbH

23 Comparación de archivos y directorios

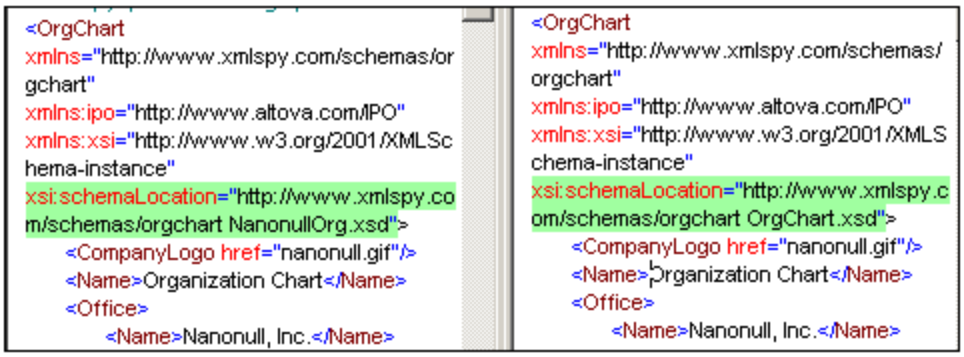
XMLSpy ofrece una función de comparación de archivos y otra de comparación de directorios. Para usar estas funciones seleccione los comandos **Comparar el archivo con...** y **Comparar directorios** del menú **Herramientas**. Las opciones de comparación de archivos se pueden definir en el cuadro de diálogo "Configuración", que aparece al hacer clic en el comando **Opciones de comparación** del menú **Herramientas**.

Todos estos comandos se describen detalladamente en la *Comandos de menú*[1547]. En los diferentes apartados de esta sección ofrecemos una introducción a los mecanismos de comparación de archivos[1094] y comparación de directorios[1095].

23.1 Comparación de archivos

Con la función de comparación de archivos[1547] puede comparar el archivo activo con otro archivo, que se selecciona desde el cuadro de diálogo "Abrir archivo" o a través de un recurso global[1042]. A continuación resumimos cómo funciona este mecanismo de comparación. Para más información, consulte la sección pertinente de la *Comandos de menú*[1547].

- La configuración elegida en el diálogo Opciones de comparación[1555] al iniciar la sesión de comparación de archivos es la configuración de la sesión.
- Puede decidir si los archivos se comparan como archivos XML (donde también se compara la estructura del documento) o como archivos de texto. Esto se configura en el cuadro de diálogo Configuración[1555], seleccionando (i) la opción *Vista Cuadrícula* o *Vista Texto* y dejando sin marcar la casilla *Comparación textual* (para comparaciones XML) o seleccionando (ii) la opción *Vista Texto* y marcando la casilla *Comparación textual* (para comparaciones de texto).
- Los dos archivos comparados aparecen en paneles contiguos en la vista seleccionada (*Vista Texto* o *Vista Cuadrícula*) y las diferencias encontradas en los dos archivos aparecen resaltadas en color (*imagen siguiente*).



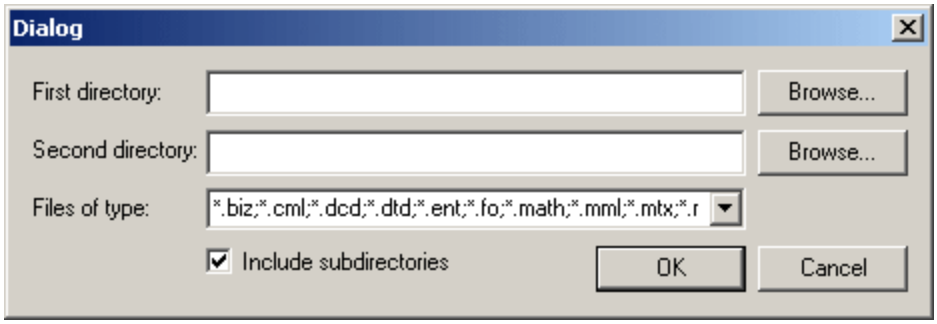
La ventana de control emergente llamada "Comparar archivos" aparece para que pueda navegar por las diferencias y combinarlas.

El cuadro de diálogo Configuración[1555] incluye varias opciones con las que puede indicar qué aspectos de los documentos XML deben tenerse en cuenta a la hora de compararlos y qué aspectos se deben ignorar. Para más información, consulte la sección Opciones de comparación[1547] de la *Comandos de menú*[1555].

© 2016-2022 Altova GmbH

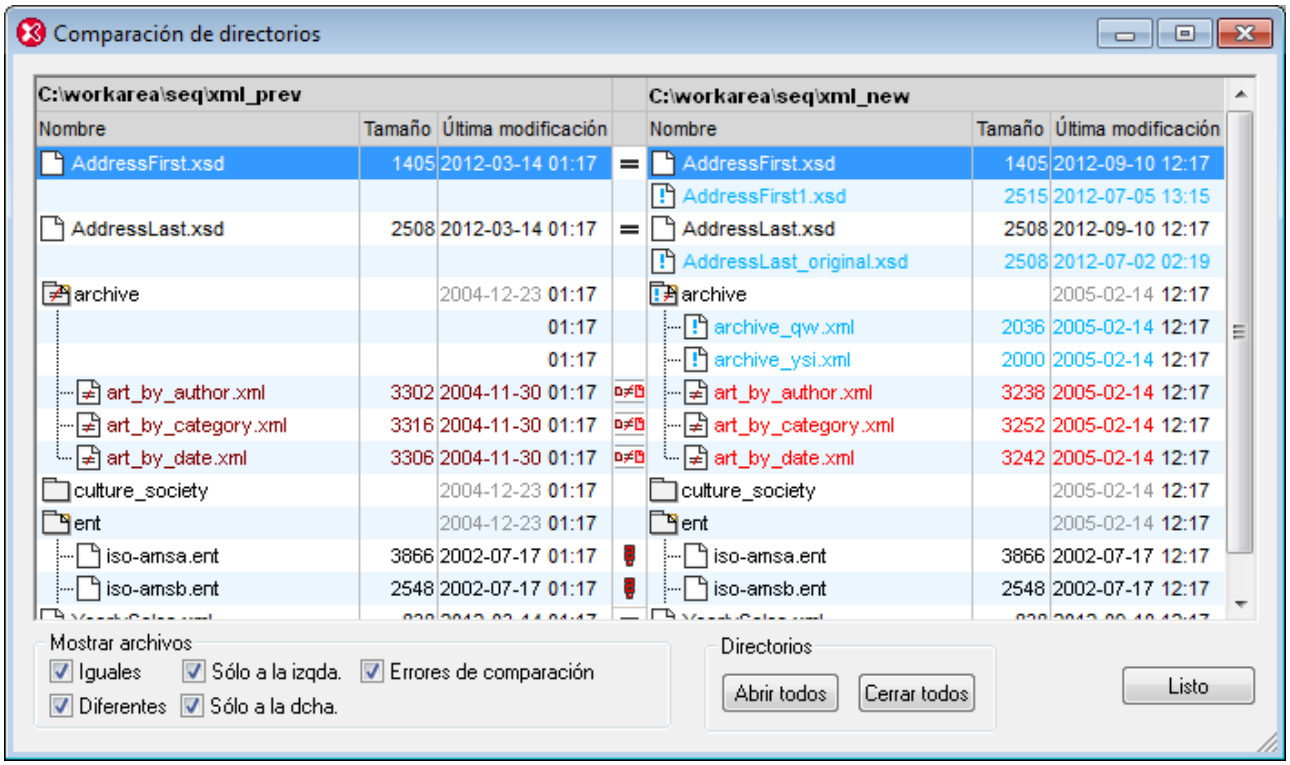
23.2 Comparación de directorios

Con la función de Comparación de directorios[1551] puede comparar dos directorios, que se seleccionan en sendos diálogos "Buscar carpetas". Si lo desea, también puede comparar los subdirectorios y especificar qué tipos de archivos se deben comparar.



Para añadir archivos ZIP a la comparación, incluya la extensión de archivo `.zip` en la lista de tipos de archivo.

La comparación de directorios indica si faltan archivos en los directorios o si los archivos que se llaman igual son idénticos o no. La comparación de los archivos se hace en función de las opciones elegidas en el cuadro de diálogo Configuración[1555]. Los resultados de la comparación de directorios aparece en una ventana distinta (*imagen siguiente*).



Para más información sobre los iconos de la ventana de comparación y sobre cómo administrar la vista, consulte la descripción del comando **Comparar directorios** de la *Comandos de menú*[1547]. Para iniciar una comparación de archivos basta con hacer doble clic en un par de archivos.

© 2016-2022 Altova GmbH

24 Control de código fuente

La función de control de código fuente de XMLSpy funciona con la API del complemento Microsoft Source Control (antes conocido como MSSCCI), versiones 1.1, 1.2 y 1.3. Gracias a esta API podrá ejecutar comandos de control de código fuente como **Proteger** y **Desproteger** desde XMLSpy directamente con prácticamente cualquier control de código fuente que permita la conexión a clientes nativos o de terceros a través de la API del complemento Microsoft Source Control.

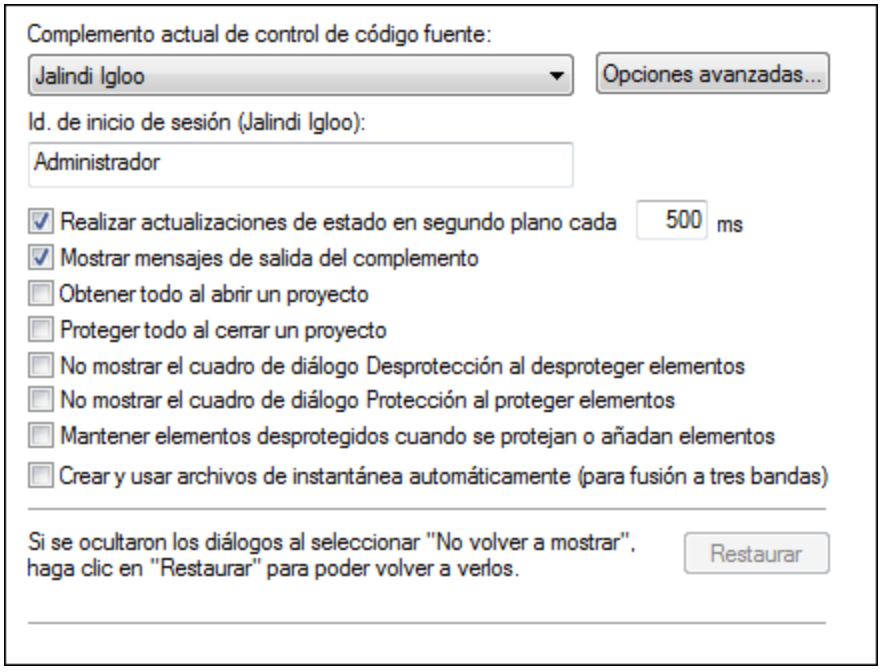
Puede usar cualquier complemento comercial o libre que sea compatible con la API del complemento Microsoft Source Control y puede conectarse a todos los sistemas de control de versiones compatibles (*ver lista de sistemas de control de código fuente compatibles* 1100).

Instalar y configurar el proveedor de control de código fuente

Para ver los proveedores de control de código fuente que están disponibles en el sistema:

1. Haga clic en **Opciones** en el menú **Herramientas**.
2. Haga clic en la sección *Control de código fuente*.

Los complementos de control de código fuente que sean compatibles con la API del complemento Microsoft Source Control aparecerán en la lista desplegable *Complemento actual de control de código fuente*.



Si no se encuentra ningún complemento compatible en el sistema, aparece este mensaje:

```
"No se Registration of installed source control providers could not be found or is
incomplete."
```

Algunos sistemas de control de código fuente no instalan el complemento de control de código fuente automáticamente. En este caso deberá instalar el complemento por separado. XMLSpy espera que los

complementos compatibles con la API del complemento Microsoft Source Code Control estén instalados bajo esta entrada del registro del sistema operativo:

```
HKEY_LOCAL_MACHINE\SOFTWARE\SourceCodeControlProvider\InstalledSCCProviders
```

Tras la instalación, el complemento aparecerá automáticamente en la lista de complementos disponibles de XMLSpy.

Acceso a los comandos de control de código fuente

Los comandos para trabajar con el control de código fuente están en el menú **Proyecto | Control de código fuente**.

Problemas de rendimiento y recursos

Algunas bases de datos de control de código fuente de gran tamaño pueden crear problemas con recursos y de rendimiento cuando realicen actualizaciones automáticas de estado en segundo plano.

Para aumentar la velocidad del sistema puede deshabilitar (o aumentar el intervalo de) la opción *Realizar actualizaciones de estado en segundo plano cada....segundos* de la sección *Control de código fuente* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).

Nota: la versión de 64 bits de XMLSpy es compatible automáticamente con todos los programas de control de código fuente de 32 bits que se enumeran en esta documentación. Cuando usa una versión de 64 bits de XMLSpy con un programa de control de código fuente de 32 bits, la opción *Realizar actualizaciones de estado en segundo plano cada....segundos* se deshabilita automáticamente y no se puede seleccionar.

Comparación de datos con Altova DiffDog

Muchos sistemas de control de código fuente (como Git y TortoiseSVN) se pueden configurar para usar la herramienta de comparación Altova DiffDog. Para más información consulte la documentación de Altova DiffDog y el sitio web de Altova.

© 2016-2022 Altova GmbH

24.1 Configurar el control de código fuente

El mecanismo para configurar el control de código fuente y poner archivos de un proyecto de XMLSpy bajo control de código fuente se describe a continuación:

1. Instale un programa de control de código fuente compatible, si todavía no tiene uno instalado (puede consultar la lista de Sistemas de control de código fuente compatibles[1100]). Configure la BD del control de código fuente (repositorio) donde desea guardar su trabajo.
2. Cree una carpeta local de la copia de trabajo donde se guardarán los archivos de trabajo que desea poner bajo control de código fuente. La carpeta que contiene todas sus carpetas y archivos de la copia de trabajo se llama *carpeta local*. La ruta de acceso de la carpeta local se denomina *ruta de acceso local*. La carpeta se enlazará a una carpeta determinada del repositorio.
3. En su aplicación de Altova cree una carpeta de proyecto de la aplicación donde debe añadir los archivos que desea poner bajo control de código fuente. El proyecto de la aplicación es una organización abstracta de los archivos. Los archivos del proyecto hacen referencia a archivos físicos guardados en la carpeta local..
4. En la BD del sistema de control de código fuente (conocida también como repositorio) se crea una carpeta que está enlazada a la carpeta local. Esta carpeta (denominada *carpeta enlazada*) es una réplica de la estructura de la carpeta local. De este modo los archivos que se ponen bajo control de código fuente se colocan en el puesto correcto dentro de la jerarquía de archivos. La carpeta enlazada se suele crear cuando se agrega por primera vez un archivo o un proyecto de la aplicación al control de código fuente. Para más información sobre la estructura de carpetas del repositorio, consulte el apartado Proyecto de la aplicación[1103].
5. Los archivos del proyecto se añaden al control de código fuente con el comando **Proyecto | Control de código fuente | Agregar al control de código fuente**. Cuando se agrega por primera vez un proyecto o un archivo del proyecto por primera vez, se crea la carpeta enlazada.
6. Las operaciones de control de código fuente, como la protección o desprotección de archivos, se realizan con los comandos del submenú **Proyecto | Control de código fuente**. Estos comandos se describen en el apartado Menú Proyecto[1302] de la *Referencia del usuario*.

Nota: si desea cambiar de proveedor de control de código fuente, puede hacerlo de dos maneras diferentes: (i) en el cuadro de diálogo "Opciones" (**Herramientas | Opciones | Control de código fuente**[1625]) o (ii) en el cuadro de diálogo "Cambiar control de código fuente" (**Proyecto | Control de código fuente | Cambiar control de código fuente**).

24.2 Sistemas de control de código fuente compatibles

A continuación puede ver una lista con todos los servidores de control de código fuente compatibles con XMLSpy, junto con sus correspondientes clientes de control de código fuente. La lista está ordenada alfabéticamente.

Notas:

- Altova ha implementado la API del complemento Microsoft Source Control (versiones 1.1, 1.2 y 1.3) en XMLSpy y ha probado la compatibilidad con los controladores y sistemas de control de versiones de la lista que aparece a continuación. Altova seguirá ofreciendo compatibilidad con estos productos cuando se actualicen.
- Los clientes de control de código fuente que no aparecen en la lista pero que implementan la API del complemento Microsoft Source Control también deberían funcionar con XMLSpy.

Sistema de control de código fuente	Clientes de control de código fuente
AccuRev 4.7.0 Windows	AccuBridge for Microsoft SCC 2008.2
Bazaar 1.9 Windows	Aigenta Unified SCC 1.0.6
Borland StarTeam 2008	Borland StarTeam Cross-Platform Client 2008 R2
Codice Software Plastic SCM Professional 2.7.127.10 (Server)	Codice Software Plastic SCM Professional 2.7.127.10 (SCC Plugin)
Collabnet Subversion 1.5.4	• Aigenta Unified SCC 1.0.6 • PushOK SVN SCC 1.5.1.1 • PushOK SVN SCC x64 versión 1.6.3.1 • TamTam SVN SCC 1.2.24
ComponentSoftware CS-RCS (PRO) 5.1	ComponentSoftware CS-RCS (PRO) 5.1
Dynamsoft SourceAnywhere for VSS 5.3.2 Standard/Professional Server	Dynamsoft SourceAnywhere for VSS 5.3.2 Client
Dynamsoft SourceAnywhere Hosted	Dynamsoft SourceAnywhere Hosted Client (22252)
Dynamsoft SourceAnywhere Standalone 2.2 Server	Dynamsoft SourceAnywhere Standalone 2.2 Client
Git	PushOK GIT SCC plug-in (consulte Control de código fuente con Git 1118)
IBM Rational ClearCase 7.0.1 (LT)	IBM Rational ClearCase 7.0.1 (LT)
March-Hare CVSNT 2.5 (2.5.03.2382)	Aigenta Unified SCC 1.0.6
March-Hare CVS Suite 2008	• Jalindi Igloo 1.0.3 • March-Hare CVS Suite Client 2008 (3321) • PushOK CVS SCC NT 2.1.2.5 • PushOK CVS SCC x64 versión 2.2.0.4 • TamTam CVS SCC 1.2.40

© 2016-2022 Altova GmbH

Sistema de control de código fuente	Clientes de control de código fuente
Mercurial 1.0.2 for Windows	Sergey Antonov HgSCC 1.0.1
Microsoft SourceSafe 2005 with CTP	Microsoft SourceSafe 2005 with CTP
Microsoft Visual Studio Team System 2008/2010 Team Foundation Server	Microsoft Team Foundation Server 2008/2010 MSSCCI Provider
Perforce 2008 P4S 2008.1	Perforce P4V 2008.1
PureCM Server 2008/3a	PureCM Client 2008/3a
QSC Team Coherence Server 7.2.1.35	QSC Team Coherence Client 7.2.1.35
Reliable Software Code Co-Op 5.1a	Reliable Software Code Co-Op 5.1a
Seapine Surround SCM Client/Server for Windows 2009.0.0	Seapine Surround SCM Client 2009.0.0
Serena Dimensions Express/CM 10.1.3 for Win32 Server	Serena Dimensions 10.1.3 for Win32 Client
Softimage Alienbrain Server 8.1.0.7300	Softimage Alienbrain Essentials/Advanced Client 8.1.0.7300
SourceGear Fortress 1.1.4 Server	SourceGear Fortress 1.1.4 Client
SourceGear SourceOffsite Server 4.2.0	SourceGear SourceOffsite Client 4.2.0 (Windows)
SourceGear Vault 4.1.4 Server	SourceGear Vault 4.1.4 Client
VisualSVN Server 1.6	• Aigenta Unified SCC 1.0.6 • PushOK SVN SCC 1.5.1.1 • PushOK SVN SCC x64 versión 1.6.3.1 • TamTam SVN SCC 1.2.24

24.3 Carpeta de trabajo local

Los archivos con los que trabajará se deberían guardar en una jerarquía dentro de la carpeta local de la copia de trabajo:

```
Carpeta local de la copia de trabajo
|
|-- MiProyecto.spp
|-- QuickStart
|   |-- QuickStart.css
|   |-- QuickStart.xml
|   |-- QuickStart.xsd
|-- Grouping
|   | -- Persons
|   |    | -- Persons.xml
```

El archivo de proyecto de la aplicación (archivo `.spp`) está por lo general dentro de la carpeta local de la copia de trabajo (*ver ejemplo anterior*).

Cuando uno de los archivos de esta carpeta (copia de trabajo) se ponen bajo control de código fuente, la estructura de la carpeta local de la copia de trabajo se reproduce por completo o parcialmente en el repositorio. Por ejemplo, si el archivo `Persons.xml` de la carpeta local del ejemplo anterior se pone bajo control de código fuente, entonces su ruta de acceso en el repositorio sería esta:

```
[CarpetaRepositorio]/MiProyecto/Grouping/Persons/Persons.xml
```

La carpeta `MiProyecto` de la carpeta del repositorio está enlazada a la carpeta local. Por lo general tendrá el nombre del proyecto, pero puede ponerle otro nombre si quiere.

Si se pone bajo control de código fuente todo el proyecto de la aplicación (seleccione el nombre del proyecto en la ventana Proyecto y póngalo bajo control de código fuente), entonces se recrea toda la estructura de la carpeta local en el repositorio.

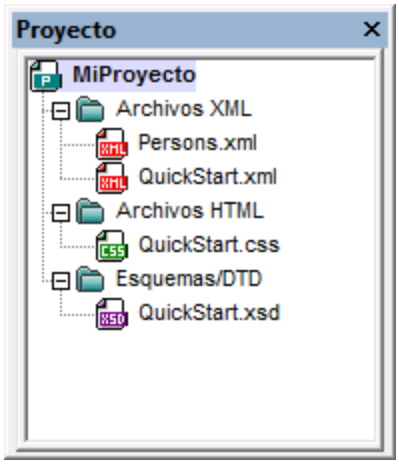
Nota: los archivos que están fuera de la carpeta local de la copia de trabajo se pueden añadir al proyecto de la aplicación. Pero dependiendo del sistema SCS que utilice podrá poner esos archivos bajo control de código fuente o no. Algunos sistemas de control de código fuente tienen problemas a la hora de poner en el repositorio archivos situados fuera de la carpeta local. Por tanto, recomendamos que guarde en la carpeta local todos los archivos que quiera poner bajo control de código fuente.

© 2016-2022 Altova GmbH

24.4 Proyecto de la aplicación

Cree o cargue el proyecto de su aplicación de Altova que quiere poner bajo control de código fuente. Si desea poner un solo archivo bajo control de código fuente, este archivo debe estar incluido en el proyecto (porque al control de código fuente solamente se puede acceder desde un proyecto).

Por ejemplo, imagine que trabaja con un proyecto de la aplicación XMLSpy. Las propiedades del proyecto están guardadas en un archivo `.spp`. En la aplicación, el proyecto se muestra en la ventana Proyecto (*imagen siguiente*). El proyecto de la imagen se llama `MiProyecto` y sus propiedades están guardadas en el archivo `MiProyecto.spp`.



Puede poner bajo control de código fuente el proyecto entero (todos sus archivos) o solamente algunos archivos del proyecto. **Solamente se pueden poner bajo control de código fuente los archivos que están incluidos en el proyecto**. Por tanto, para poner archivos bajo control de código fuente, antes debe agregarlos al proyecto. El archivo de proyecto (archivo `.spp`) se pone bajo control de código fuente automáticamente cuando uno de sus archivos se ponen bajo control de código fuente.

El proyecto y sus archivos se ponen bajo control de código fuente con el comando **Proyecto | Control de código fuente | Agregar al control de código fuente** (*ver el apartado siguiente*).

No obstante, recuerde que la estructura de la carpeta del repositorio no se corresponde a la estructura de la carpeta del proyecto (*imagen anterior*) sino a la estructura de la carpeta local de la copia de trabajo[1102] (*ver más abajo*). En el ejemplo siguiente observe que la estructura de la carpeta `MiProyecto` del repositorio se corresponde con la de la carpeta local de la copia de trabajo. Observe también que la carpeta enlazada está dentro de la carpeta del repositorio.

```
Carpeta local de la copia de trabajo            Repositorio
|                                                |
|-- MiProyecto.spp                               |-- MiProyecto (enlazada a la copia
local)
|-- QuickStart                                   |   |-- MiProyecto.spp
|   |-- QuickStart.css                           |   |-- QuickStart
|   |-- QuickStart.xml                           |   |   |-- QuickStart.css
|   |-- QuickStart.xsd                           |   |   |-- QuickStart.xml
|-- Grouping                                     |   |   |-- QuickStart.xsd
```

```
|   | -- Persons                                    |   |-- Grouping
|   |    | -- Persons.xml                           |   |   | -- Persons
                                                    |   |   |    | -- Persons.xml
```

Nota: un proyecto de la aplicación puede contener carpetas de proyecto (verdes) o carpetas externas (amarillas). Solamente puede poner bajo control de código fuente archivos que estén dentro de carpetas de proyecto (verdes). Los archivos de las carpetas externas no se pueden poner bajo control de código fuente.

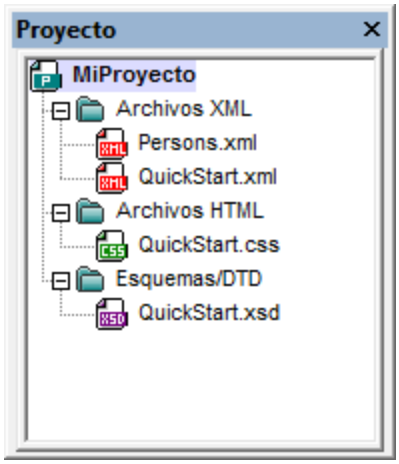
Nota: los archivos que están fuera de la carpeta local de la copia de trabajo se pueden añadir al proyecto de la aplicación. Pero dependiendo del sistema SCS que utilice podrá poner esos archivos bajo control de código fuente o no. Algunos sistemas de control de código fuente tienen problemas a la hora de poner en el repositorio archivos situados fuera de la carpeta local. Por tanto, recomendamos que guarde en la carpeta local todos los archivos que quiera poner bajo control de código fuente.

© 2016-2022 Altova GmbH

24.5 Agregar al control de código fuente

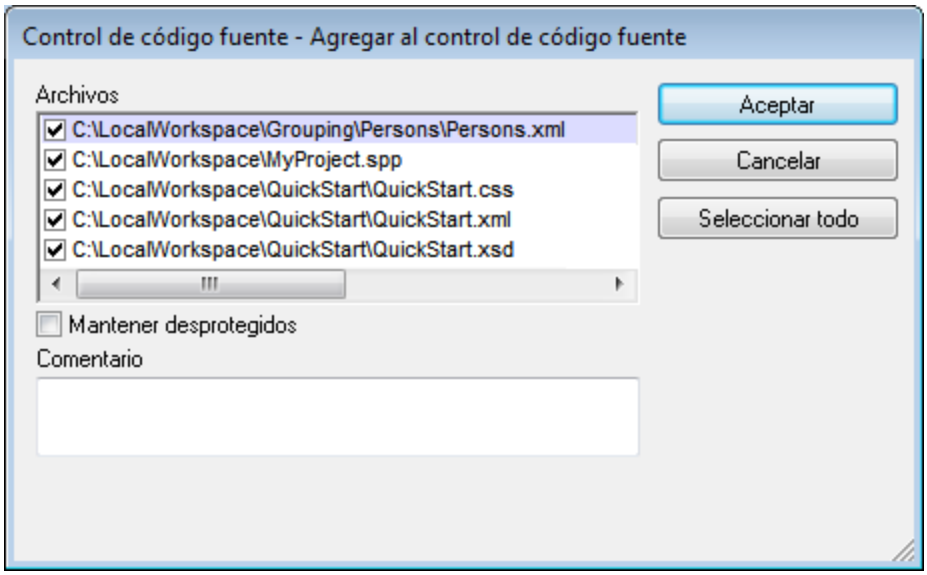
Cuando se agrega el proyecto al control de código fuente se crean automáticamente los enlaces correspondientes y la estructura del repositorio antes de poner el archivo de proyecto (archivo `.spp`) o los archivos bajo control de código fuente. A continuación explicamos cómo agregar el proyecto al control de código fuente.

Seleccione el proyecto en la ventana Proyecto (*imagen siguiente*) y asegúrese de que está resaltado el título del proyecto (como en la imagen). También puede seleccionar uno o varios archivos. Si añade un archivo al control de código fuente, se añade también el archivo de proyecto (archivo `.spp`).



Después seleccione el comando **Proyecto | Control de código fuente | Agregar al control de código fuente**. Aparecen los cuadros de diálogo de conexión y configuración de sus sistema de control de código fuente. (Para cambiar de sistema haga clic en **Proyecto | Control de código fuente | Cambiar control de código fuente**.)

Siga las instrucciones del sistema de control de código fuente para realizar la conexión y configurar las opciones. Cuando termine, los archivos que seleccionó y el archivo de proyecto (archivo .spp) aparecen en el cuadro de diálogo "Agregar al control de código fuente" (*imagen siguiente*). Seleccione los archivos que desea agregar y pulse el botón **Aceptar**.



Los archivos se añadirán al repositorio y estarán protegidos o desprotegidos dependiendo de si marcó o no la casilla *Mantener desprotegidos*.

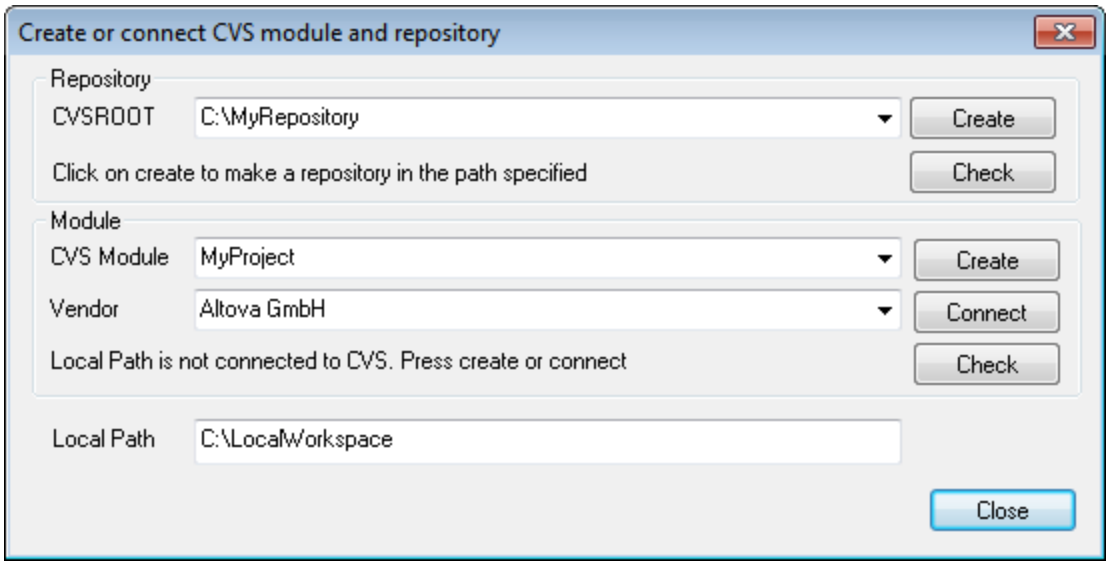
Notas de configuración

Puede que se solicite crear una carpeta para el proyecto en el repositorio (si todavía no se creó). La carpeta local de la copia de trabajo[1102] estará enlazada a esta carpeta del repositorio (*ver ejemplo siguiente)*.

```
Carpeta local de la copia de trabajo                Repositorio
|                                                   |
|-- MiProyecto.spp                                  |-- MiProyecto (enlazada a la copia
local)
|-- QuickStart                                      |   |-- MiProyecto.spp
|   |-- QuickStart.css                              |   |-- QuickStart
|   |-- QuickStart.xml                              |   |   |-- QuickStart.css
|   |-- QuickStart.xsd                              |   |   |-- QuickStart.xml
|-- Grouping                                        |   |   |-- QuickStart.xsd
|   | -- Persons                                    |   |-- Grouping
|   |    | -- Persons.xml                           |   |   | -- Persons
                                                    |   |   |    | -- Persons.xml
```

Por ejemplo, aquí puede ver el cuadro de diálogo de configuración de Jalindi Igloo. El campo *CVSROOT* es la ruta de acceso de la carpeta del repositorio.

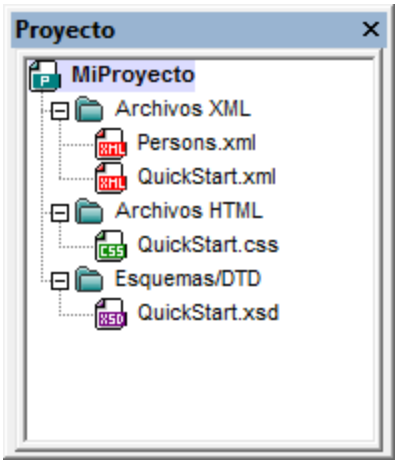
© 2016-2022 Altova GmbH



En la imagen anterior, la ruta de acceso local se corresponde con la ruta de acceso del módulo CVS `MiProyecto`.

24.6 Trabajar con control de código fuente

Para trabajar con control de código fuente seleccione el proyecto, una carpeta de proyecto o un archivo del proyecto en la ventana Proyecto (imagen siguiente) y seleccione un comando del menú **Proyecto | Control de código fuente**. Los comandos **Proteger** y **Desproteger** también están disponibles en el menú contextual que aparece al hacer clic con el botón derecho en las entradas de la ventana Proyecto.



En esta sección describimos las características principales del control de código fuente:

- Agregar o quitar del control de código fuente 1108
- Proteger, Desproteger 1109
- Obtener archivos como archivos de solo lectura 1112
- Copiar y compartir desde el control de código fuente 1113
- Cambiar control de código fuente 1116

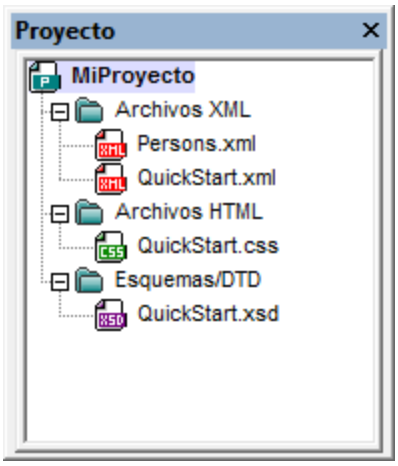
Los demás comandos del menú **Proyecto | Control de código fuente** se describen en la Referencia del usuario 1302. Para más información consulte la documentación de sus sistema de control de código fuente.

24.6.1 Agregar o quitar del control de código fuente

Agregar

Después de añadir un proyecto al control de código fuente, puede añadir archivos o grupos de archivos al sistema de control. Seleccione el archivo en la ventana Proyecto y haga clic en el comando **Proyecto | Control de código fuente | Agregar al control de código fuente**. Para seleccionar varios archivos mantenga pulsada la tecla **Ctrl** mientras hace clic en los archivos que desea añadir. Si ejecuta el comando en una carpeta de proyecto (de color verde), entonces se agregan al control de código fuente todos los archivos y subcarpetas de la carpeta.

© 2016-2022 Altova GmbH



Cuando se agregan archivos al control de código fuente, la jerarquía de la carpeta local se copia al repositorio[1103] (no la jerarquía de la carpeta de proyecto). Es decir, si un archivo está en una subcarpeta de la carpeta local, entonces se crean en el repositorio la carpeta primaria y las demás carpetas antecesoras del archivo.

Cuando se añade el primer archivo de un proyecto al control de código fuente, se crean los enlaces correspondientes en el repositorio y se añade automáticamente el archivo de proyecto (archivo `.spp`). Para más información consulte el apartado Agregar al control de código fuente[1105].

Símbolos de control de código fuente

Las carpetas y los archivos aparecen con un icono diferente dependiendo de su estado:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

Quitar

Para quitar un archivo del control de código fuente, selecciónelo y haga clic en el comando **Proyecto | Control de código fuente | Quitar del control de código fuente**. También puede quitar: (i) archivos de una carpeta de proyecto ejecutando el comando sobre la carpeta y (ii) todo el proyecto, ejecutando el comando sobre el proyecto.

24.6.2 Proteger, desproteger

Tras poner un archivo de proyecto bajo control de código fuente, puede desprotegerlo o protegerlo. Para ello seleccione el archivo (en la ventana Proyecto) y seleccione el comando pertinente en el menú **Proyecto | Control de código fuente**, es decir **Desproteger** o **Proteger**.

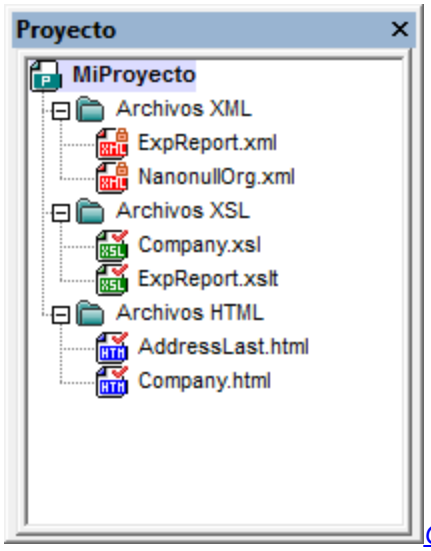
Cuando se desprotege un archivo, se pone una copia del archivo del repositorio en la carpeta local. Los archivos desprotegidos se pueden editar. Si un archivo que está bajo control de código fuente no se

desprotege, no se puede editar. Después de editar el archivo, los cambios se pueden guardar en el repositorio protegiendo el archivo. Incluso si no guarda el archivo, al protegerlo se guardan los cambios en el repositorio. El icono del archivo indica si está desprotegido o protegido.

Las carpetas y los archivos aparecen con un icono diferente dependiendo de su estado:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

Si selecciona el proyecto o una carpeta del proyecto, se seleccionan todos los archivos del objeto seleccionado. Para seleccionar varios objetos (archivos y carpetas) pulse la tecla **Ctrl** mientras hace clic en los objetos. La imagen siguiente muestra un proyecto que se desprotegió. Después se protegió el archivo `ExpReport.xml`.



Obtener archivos como archivos de solo lectura 1112

Guardar cambios y rechazar cambios

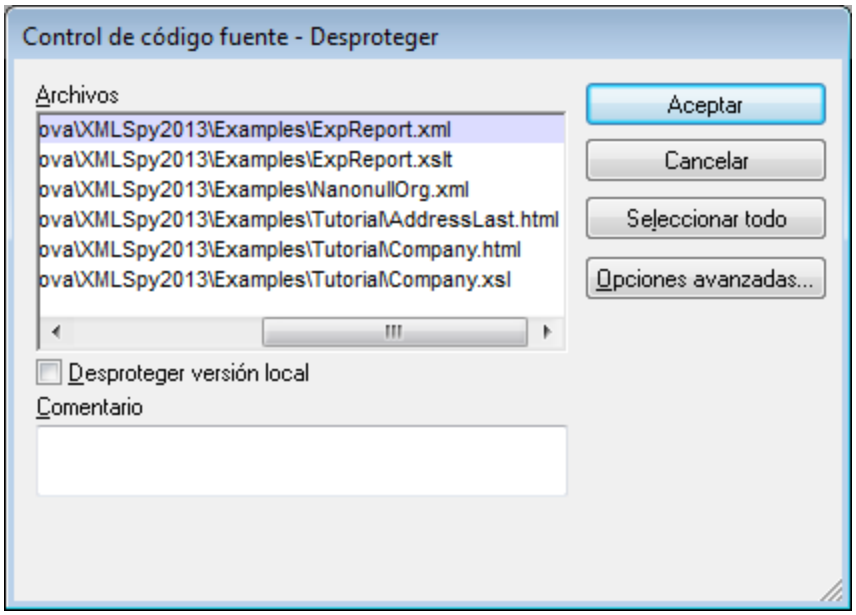
Recuerde que al proteger un archivo puede elegir si se protegen los cambios o no. Es decir, puede guardar los cambios en el repositorio y seguir con el archivo desprotegido. Esto es muy útil si desea guardar cambios cada cierto tiempo y después seguir editando el archivo desprotegido.

Si desprotegió un archivo y realizó cambios pero quiere rechazarlos por cualquier motivo, puede volver a la versión del documento guardada en el repositorio con solo hacer clic en el comando **Proyecto | Control de código fuente | Deshacer desprotección**.

Desproteger

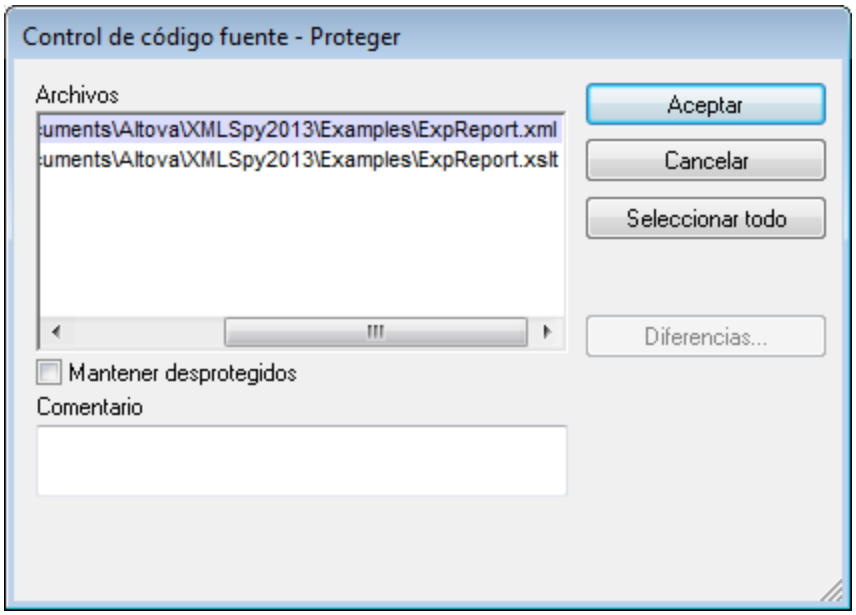
El cuadro de diálogo "Desproteger" (*imagen siguiente*) sirve para: (i) seleccionar los archivos que desea desproteger y (ii) seleccionar si se desprotege la versión del repositorio o la versión local.

© 2016-2022 Altova GmbH



Proteger

El cuadro de diálogo "Proteger" (*imagen siguiente*) sirve para: (i) seleccionar los archivos que desea proteger y (ii) mantener el archivo desprotegido.

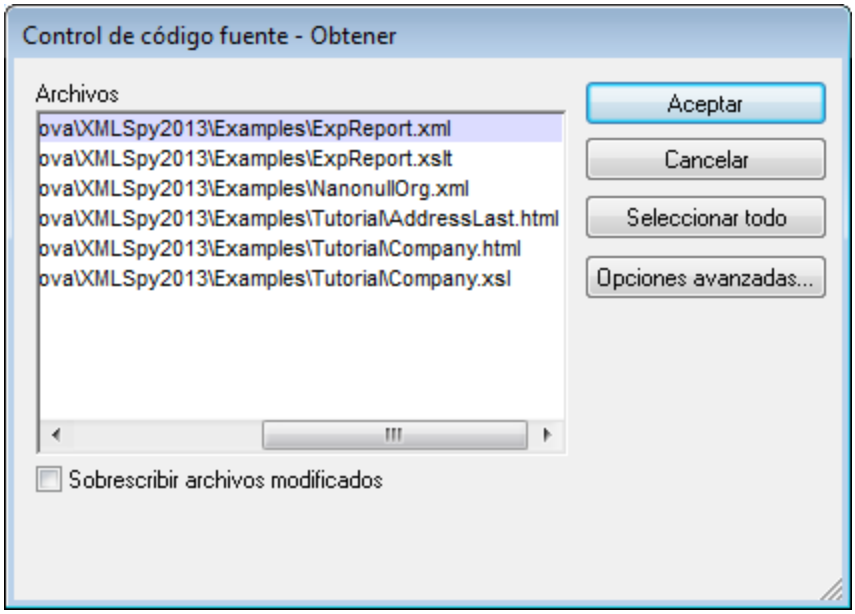


Nota: en estos dos cuadros de diálogo aparecen varios archivos si el objeto seleccionado (el proyecto o las carpetas de proyecto) contienen varios archivos.

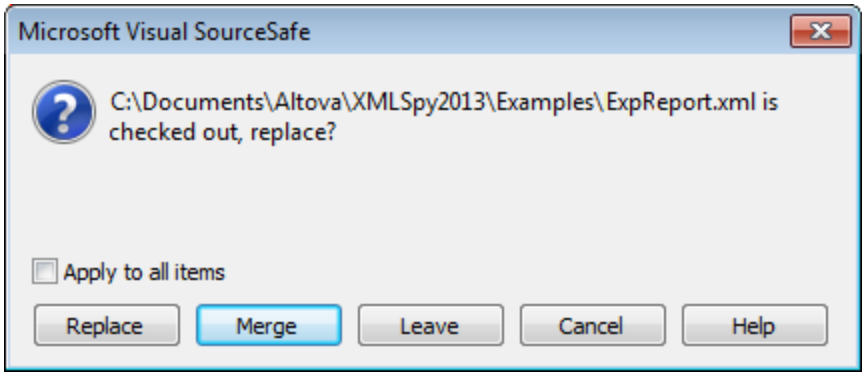
24.6.3 Obtener archivos como archivos de solo lectura

El comando **Obtener** (del menú **Proyecto | Control de código fuente**) recupera los archivos del repositorio como archivos de solo lectura. (Para poder editar un archivo, antes es necesario desprotegerlo.) El cuadro de diálogo "Obtener" enumera los archivos del objeto (proyecto o carpeta) sobre el que se ejecutó el comando **Obtener** (*imagen siguiente*). En el cuadro de diálogo puede marcar las casillas de los archivos que desea recuperar.

Nota: el comando **Obtener carpetas** sirve para seleccionar subcarpetas del repositorio (si su sistema de control de código fuente permite esta opción).



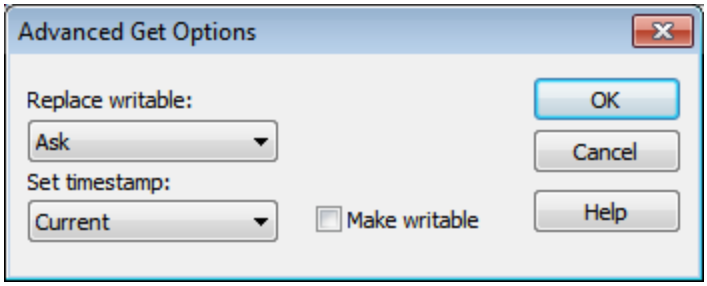
Si quiere, puede sobrescribir los archivos desprotegidos modificados marcando la opción *Sobrescribir archivos modificados* situada en la parte inferior del cuadro de diálogo. Al hacer clic en **Aceptar** los archivos se sobrescriben. Si alguno de los archivos sobrescritos está abierto, aparece un aviso preguntando si desea volver a cargar los archivos (botón **Volver a cargar**), cerrar los archivos (botón **Cerrar**) o conservar la vista actual del archivo (botón **Cancelar**).



© 2016-2022 Altova GmbH

Opciones avanzadas

El cuadro de diálogo "Opciones avanzadas" (*imagen siguiente*) se abre con el botón **Opciones avanzadas** del cuadro de diálogo "Obtener" (*primera imagen de este apartado*).



Aquí puede seleccionar (i) si reemplaza los archivos que se pueden escribir y que están desprotegidos, (ii) la marca de tiempo y (iii) si la propiedad de solo lectura del archivo recuperado se cambia para que el archivo se pueda escribir.

Obtener la versión más reciente

El comando **Obtener la versión más reciente** (del menú **Proyecto | Control de código fuente**) recupera y coloca en el directorio de trabajo la versión más reciente de los archivos seleccionados del control de código fuente. Los archivos se recuperan como archivos de solo lectura y no están desprotegidos. Este comando es igual que el comando **Obtener**, la única diferencia es que no aparece el cuadro de diálogo "Obtener".

Si los archivos seleccionados están desprotegidos, la acción que se toma dependerá de lo que haga su sistema de control en situaciones así. Por lo general, el sistema de control de código fuente le preguntará si desea reemplazar el archivo desprotegido, combinarlo o dejarlo tal y como está.

Nota: cuando seleccione este comando para una carpeta, el comando se aplica de forma recursiva. Esto significa que se aplica a todos los archivos situados dentro de la carpeta actual.

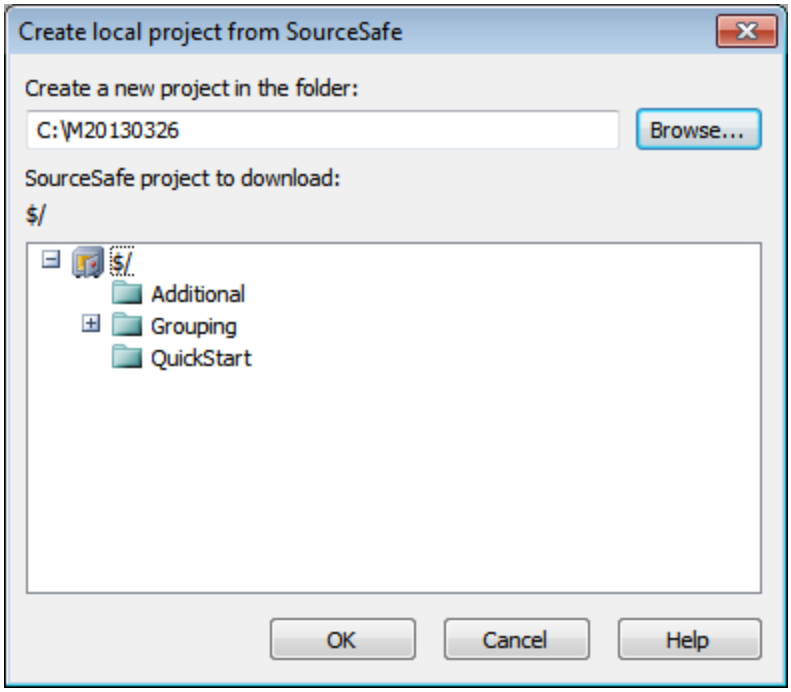
24.6.4 Copiar y compartir desde el control de código fuente

El comando **Abrir desde el control de código fuente** crea un proyecto de aplicación nuevo a partir de un proyecto bajo control de código fuente.

Estos son los pasos para crear el proyecto:

1. Dependiendo del sistema de control de código fuente que utilice, quizás sea necesario comprobar que ningún archivo del proyecto está desprotegido antes de crear un proyecto nuevo.
2. No es necesario tener abierto un proyecto en la aplicación.
3. Seleccione el comando **Proyecto | Control de código fuente | Abrir desde el control de código fuente**.
4. El sistema de control de código fuente que tiene instalado muestra sus cuadros de diálogo de verificación y conexión. Conéctese al repositorio pertinente (es decir, a la carpeta enlazada del repositorio que equivale a la carpeta local 1103).

5. En el cuadro de diálogo que aparece (*imagen siguiente*) busque la carpeta local en la que desea copiar el contenido de la carpeta del repositorio (a la que se acaba de conectar). En la imagen siguiente, por ejemplo, la carpeta asociada del repositorio se llama `MiProyecto` y se representa con el signo `$`. La carpeta local es `C:\M20130326`.



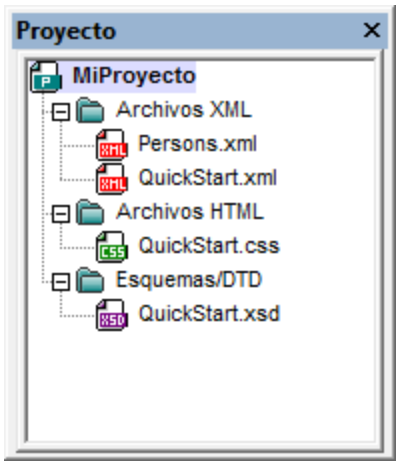
6. Haga clic en **OK**. El contenido de la carpeta asociada (`MiProyecto`) se copiará a la carpeta local `C:\M20130326`. y aparece un cuadro de diálogo pidiendo que seleccione el archivo de proyecto (archivo `.spp`) que desea crear como proyecto nuevo.
7. Seleccione el archivo `.spp` que se copió a la carpeta local. En nuestro ejemplo, se trata del archivo `MiProyecto.spp` situado en la carpeta `C:\M20130326`. En la aplicación se crea un proyecto nuevo llamado `MiProyecto` y este proyecto aparece en la ventana Proyecto. Los archivos del proyecto están en la carpeta `C:\M20130326`.

Compartir desde el control de código fuente

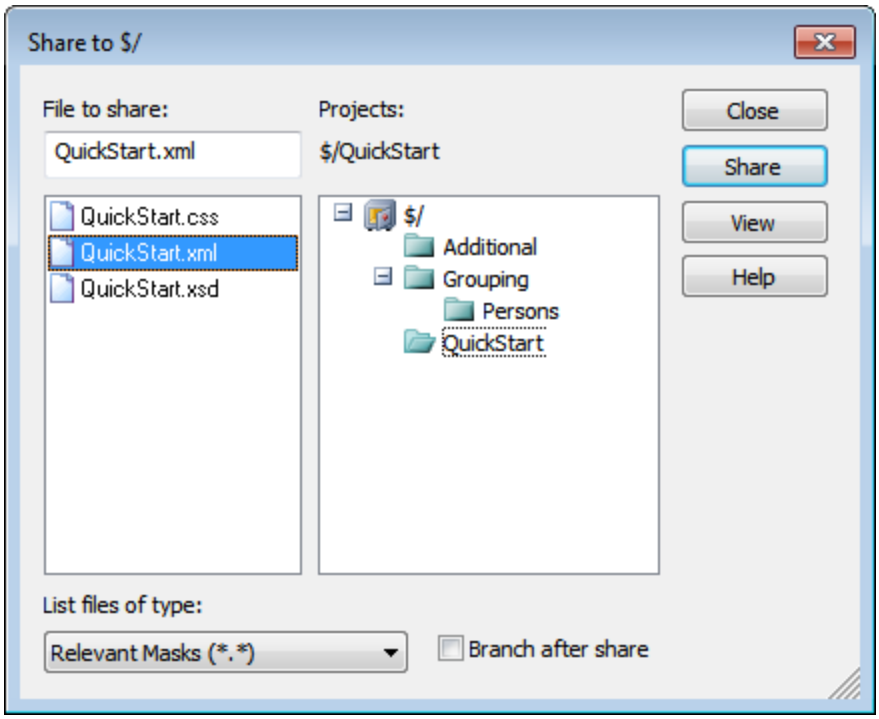
El comando **Compartir desde el control de código fuente** está disponible si su sistema de control de código fuente permite compartir contenido. Por ejemplo, puede compartir un archivo para que esté disponible en varias ubicaciones locales. Si se realizan cambios en uno de estos archivos locales, los cambios se reflejan en todas las versiones "compartidas" del archivo.

En la ventana Proyecto seleccione el proyecto (*imagen siguiente*) y después haga clic en **Compartir desde el control de código fuente**.

© 2016-2022 Altova GmbH

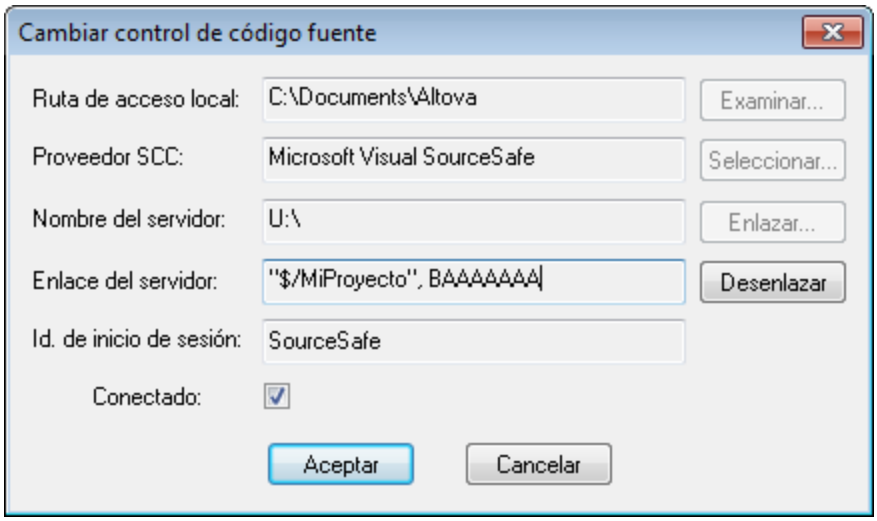


Aparece el cuadro de diálogo "Share To [carpeta]" (*imagen siguiente*).



Ahora puede seleccionar los archivos que desea compartir. Primero seleccione en el panel derecho la carpeta donde están los archivos. Los archivos de la carpeta elegida aparecen en el panel izquierdo. Seleccione el archivo que desea compartir (para seleccionar varios pulse la tecla **Ctrl** mientras hace clic en los archivos). Los archivos seleccionados aparecen en el cuadro situado en la parte superior izquierda del diálogo. Los archivos desaparecen del panel izquierdo. Haga clic en **Share** (*compartir*) y después en **Close** (cerrar) para copiar los archivos seleccionados a la carpeta local compartida.

El nombre de la carpeta compartida aparece en el nombre del cuadro de diálogo. En la imagen anterior, por ejemplo, se trata de la carpeta local (porque el signo `$` representa la carpeta del repositorio a la que está enlazada la carpeta local). En el cuadro de diálogo "Cambiar control de código fuente" puede ver y configurar la carpeta compartida (*imagen siguiente*) cambiando la ruta de acceso local y el enlace de servidor.



Para más información consulte la documentación de su sistema de control de código fuente.

24.6.5 Cambiar control de código fuente

El control de código fuente se puede cambiar con dos comandos diferentes del menú **Proyecto | Control de código fuente**:

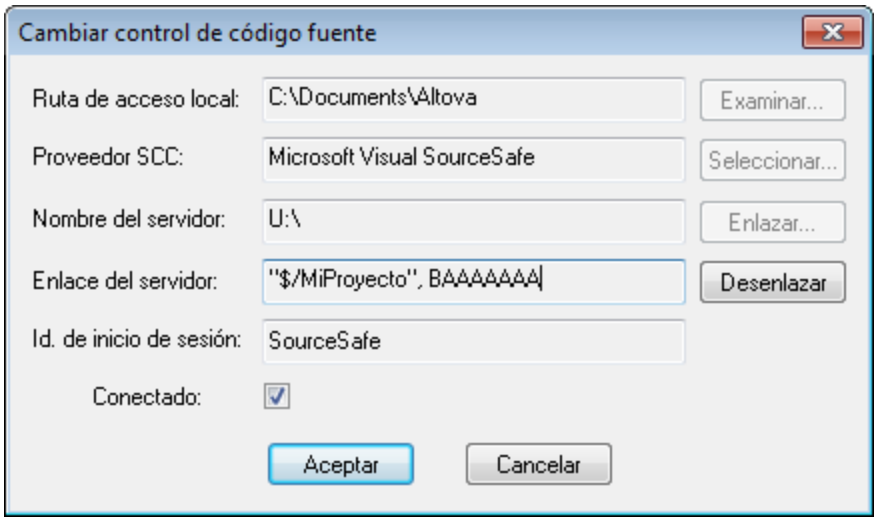
- Administrador del control de código fuente, que abre el programa de control de código fuente y permite configurar bases de datos y enlaces.
- Cambiar control de código fuente, que abre el cuadro de diálogo del mismo nombre, donde puede cambiar de sistema (ver la descripción que aparece más abajo).

El enlace actual es lo que usa el proyecto de aplicación activo para conectarse a la BD del control de código fuente. Por tanto, en enlace actual debe ser correcto. Esto significa que el archivo de proyecto de la aplicación (archivo `.spp`) debe estar en la carpeta local y la carpeta enlazada del repositorio debe ser la base de datos donde se almacenan los archivos de este proyecto. Por lo general, la carpeta enlazada y sus subcarpetas se corresponden con la carpeta de la copia de trabajo local y sus subcarpetas.

En el cuadro de diálogo "Cambiar control de código fuente" (*imagen siguiente*) puede cambiar el sistema de control (*Proveedor SCC*), la carpeta local (*Ruta de acceso local*) y el enlace al repositorio (*Nombre del servidor* y *Enlace del servidor*).

Las opciones solamente se pueden cambiar después de desenlazar el enlace actual. Para ello, pulse el botón **Desenlazar**. Ahora puede editar las demás opciones.

© 2016-2022 Altova GmbH



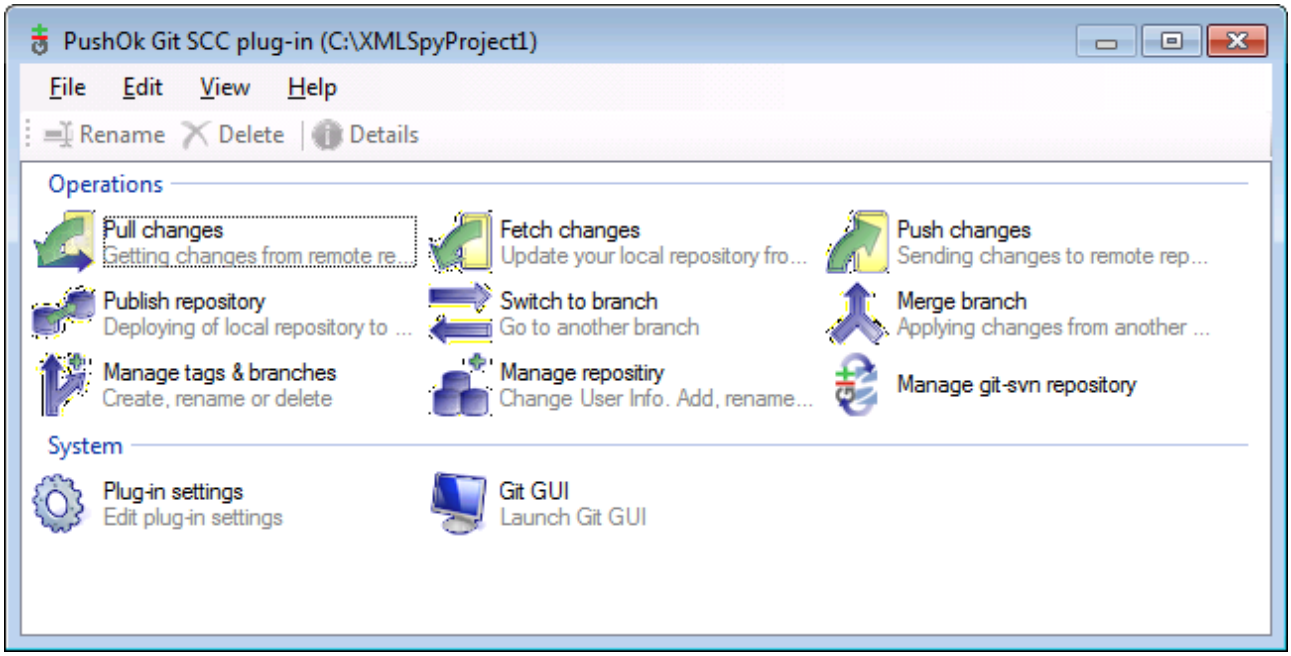
1. Con el botón **Examinar** puede buscar la carpeta local y con el botón Seleccionar puede seleccionar uno de los sistemas de control de código fuente instalados.
2. Después puede enlazar la carpeta local a una BD del repositorio. Para ello pulse el botón **Enlazar**. Aparece el cuadro de diálogo de conexión de sus sistema de control.
3. Si escribió un *Id. de inicio de sesión*, este identificador se pasa al sistema de control. De lo contrario, deberá escribir sus datos de inicio de sesión en el diálogo de conexión.
4. Seleccione la BD del repositorio a la que desea enlazar esta carpeta local.
5. Cuando termine haga clic en **Aceptar**.

24.7 Control de código fuente con Git

XMLSpy es compatible con el sistema de control de versiones Git por medio de un complemento externo llamado **GIT SCC plug-in** (http://www.pushok.com/software/git.html).

> Cuando se redactó esta documentación, la versión del complemento **GIT SCC plug-in** era una versión experimental. Para usar el complemento es necesario registrarse con el autor del complemento.

El complemento GIT SCC permite trabajar con repositorios Git utilizando los comandos del menú **Proyecto | Control de código fuente** de XMLSpy. Recuerde que los comandos de este menú vienen de la API del complemento Microsoft Source Control, cuyo diseño es diferente al de Git. Como consecuencia, el complemento hace de intermediario entre las funciones tipo Visual Source Safe y las funciones de Git. Esto significa, por un lado, que algunos comandos como **Obtener la versión más reciente** no estarán habilitados cuando trabaje con Git. Por otro lado, hay acciones nuevas propias de Git que están disponibles en el cuadro de diálogo de administración del código fuente (**Proyecto | Control de código fuente | Administrador del control de código fuente** en XMLSpy).



En el menú **Proyecto | Control de código fuente** también encontrará los comandos más frecuentes de Git.

Los diferentes apartados de esta sección describen la configuración inicial del complemento y el flujo de trabajo básico:

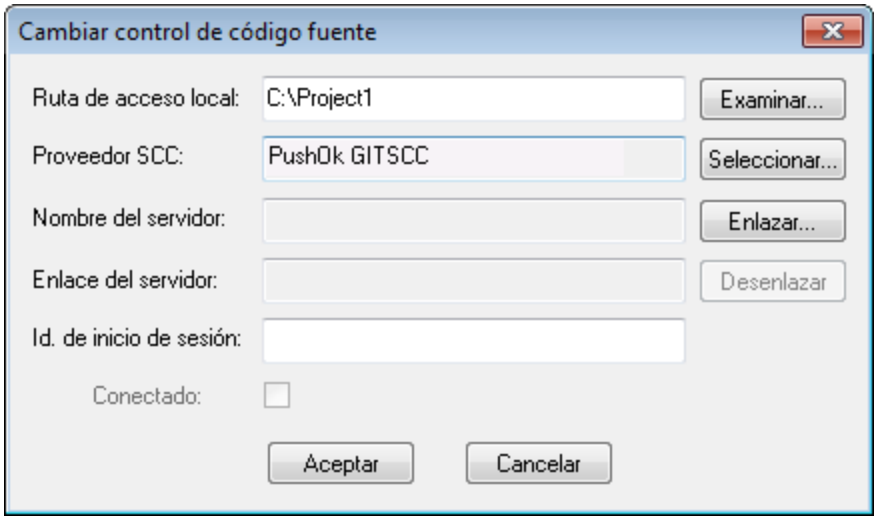
- Habilitar Git con el complemento GIT SCC 1119
- Agregar un proyecto al control de código fuente de Git 1119
- Clonar un proyecto desde el control de código fuente de Git 1121

© 2016-2022 Altova GmbH

24.7.1 Habilitar Git con el complemento de control de código fuente

Para habilitar el control de código fuente de Git en XMLSpy es necesario tener instalado el complemento externo **PushOK GIT SCC plug-in**, registrarse y seleccionarlo en la lista de proveedores de control de código fuente:

1. Descargue el archivo de instalación del complemento desde el sitio web del autor (http://www.pushok.com), ejecútelo y siga las instrucciones que aparecen en pantalla.
2. En el menú **Proyecto** de XMLSpy, haga clic en **Proyecto | Control de código fuente | Cambiar de control de código fuente** y seleccione **PushOk GITSCC**. Si **Push Ok GITSCC** no aparece en la lista de proveedores, es probable que la instalación del complemento no finalizara correctamente. Consulte la documentación del autor para resolver este problema.



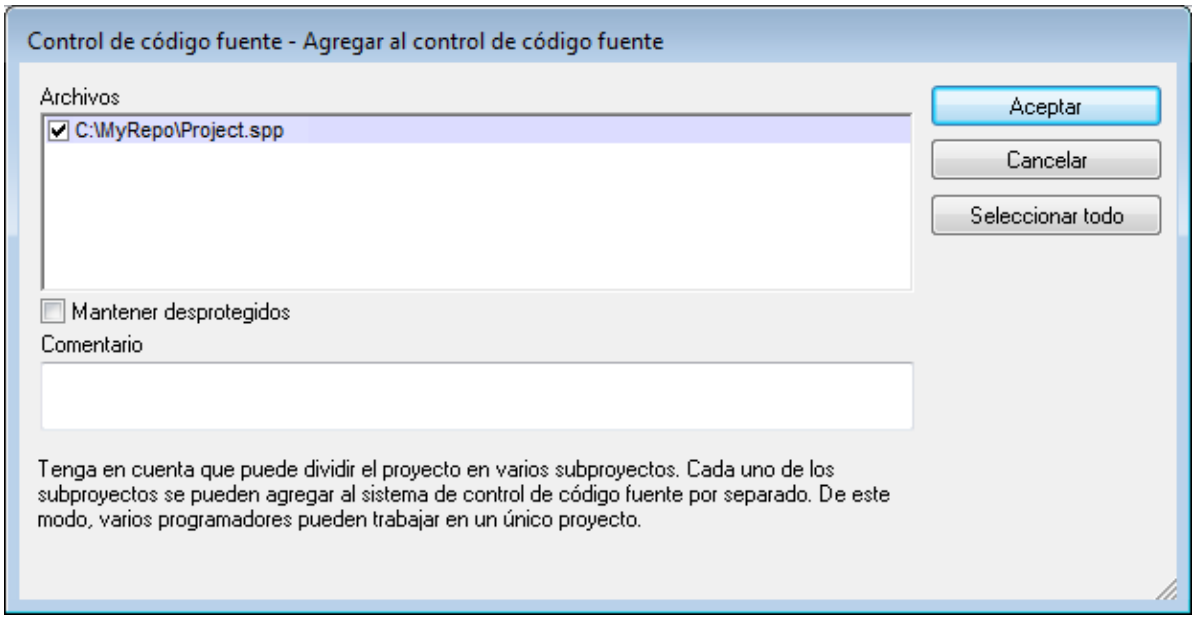
3. Para terminar debe registrar el complemento haciendo clic en **Registration**. Siga los pasos del asistente para terminar de registrar el complemento.

24.7.2 Agregar un proyecto al control de código fuente de Git

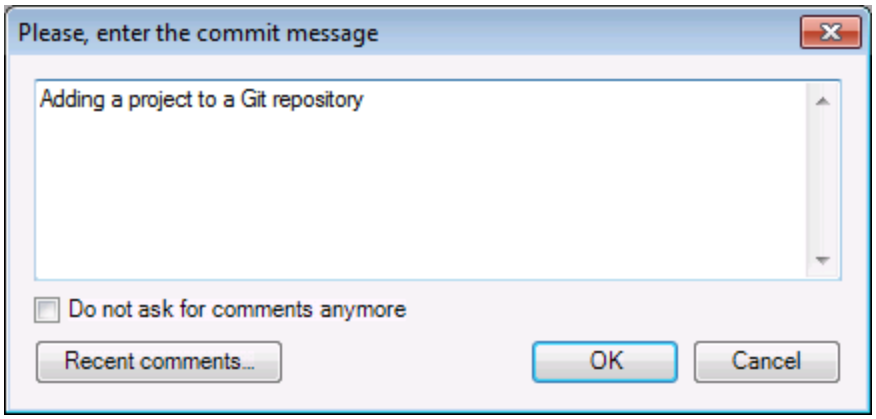
Puede guardar proyectos de XMLSpy como repositorios de Git. La estructura de los archivos o carpetas que añada al proyecto se corresponderán con la estructura del repositorio Git.

Para agregar un proyecto al control de código fuente de Git:

1. Compruebe que el proveedor de control de código fuente seleccionado es **PushOK GIT SCC Plug-in** (*ver el apartado anterior*[1119]).
2. Cree un proyecto nuevo con el comando de menú **Proyecto | Crear proyecto**.
3. Guarde el proyecto en una carpeta local (p. ej. `C:\MyRepo\Project.spp`).
4. Ahora haga clic en **Proyecto | Control de código fuente | Agregar al control de código fuente**.



5. Haga clic en **Aceptar**.



6. Escriba el texto del mensaje de confirmación y haga clic en **OK** para agregar el proyecto al control de código fuente.

Ahora ya puede añadir archivos y carpetas al proyecto. Recuerde que todos los archivos y carpetas del proyecto deben estar bajo la carpeta raíz del proyecto. Por ejemplo, si creó el proyecto en la carpetea `C:\MyRepo`, entonces solamente podrá añadir al proyecto los archivos que estén bajo `C:\MyRepo`. Si intenta añadir archivos de proyecto que estén fuera de la carpeta raíz del proyecto, aparecerá este mensaje de advertencia:

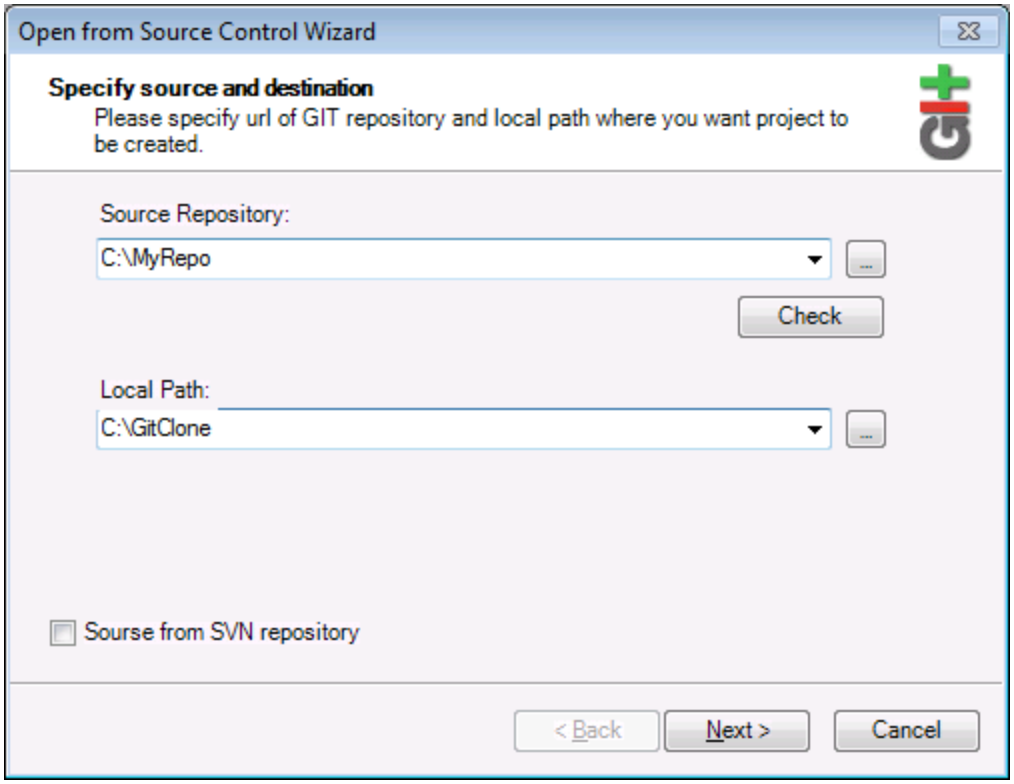
Sólo se pueden agregar archivos a una ubicación bajo la raíz de enlace del proyecto (C:\MyRepo).

© 2016-2022 Altova GmbH

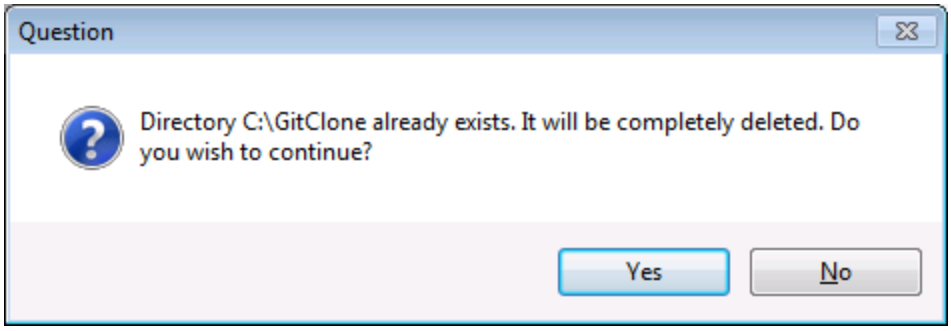
24.7.3 Clonar un proyecto desde el control de código fuente de Git

Los proyectos que ya estén en el control de código fuente de Git (*ver el apartado anterior*[1119]) se pueden abrir desde el repositorio Git:

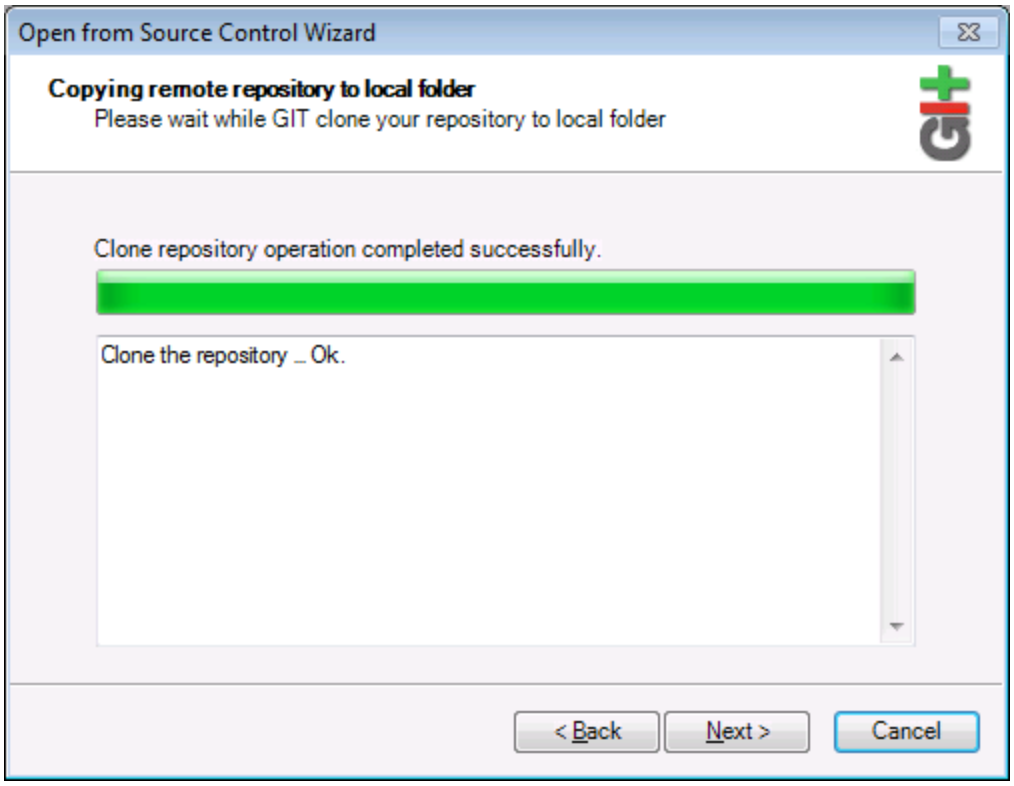
1. Compruebe que el proveedor de control de código fuente seleccionado es **PushOK GIT SCC Plug-in** (*ver el apartado habilitar Git con complemento de control de código fuente GIT SCC*[1119]).
2. Haga clic en **Proyecto | Control de código fuente | Abrir desde el control de código fuente**.
3. Escriba la ruta de acceso o la URL del repositorio fuente. Haga clic en el botón **Check** para verificar la ruta de acceso o la dirección URL.



4. En el campo *Local Path* escriba la ruta de acceso de la carpeta local donde desea crear el proyecto y haga clic en **Next** para continuar. Si la carpeta local ya existe (aunque esté vacía), aparece este cuadro de diálogo preguntando si desea borrar totalmente la carpeta:



5. Haga clic en **Yes** para confirmar y después en **Next** para continuar.



6. Siga los pasos del asistente hasta el final.
7. Al final aparece un cuadro de diálogo "Explorar" donde puede abrir el proyecto de XMLSpy (archivo *.spp). Seleccione el archivo de proyecto para cargar el contenido del proyecto en XMLSpy.

© 2016-2022 Altova GmbH

25 XMLSpy en Visual Studio

XMLSpy se puede integrar en las versiones 2012/2013/2015/2017/2019/2022 del entorno IDE Microsoft Visual Studio. Esto hace posible incorporar las avanzadas funciones de edición XML al avanzado entorno de desarrollo de Visual Studio.

En esta sección describimos:

- El proceso general de instalación[1124] y la integración del complemento de XMLSpy para Visual Studio.
- Las diferencias[1125] entre XMLSpy y el complemento de XMLSpy para Visual Studio.
- Los depuradores de XMLSpy[1128] en Visual Studio.

Nota: las imágenes y capturas de pantalla de esta sección son de Visual Studio 2010. Si utiliza otra versión de Visual Studio, puede que encuentre diferencias entre estas imágenes y su versión.

25.1 Instalación del complemento de XMLSpy

Siga estas instrucciones para instalar el complemento de XMLSpy para Visual Studio:

1. Instale Microsoft Visual Studio 2012/2013/2015/2017/2019/2022. Tenga en cuenta que a partir de su versión 2022, Visual Studio solo está disponible como aplicación de 64 bits.
2. Instale XMLSpy (ediciones Enterprise o Professional Edition). Si tiene instalada la versión 2022 o una más avanzada, debe instalar la versión de 64 bits de XMLSpy.
3. Descargue y ejecute el paquete de integración de XMLSpy. Este paquete está disponible en la página de descargas de XMLSpy (ediciones Enterprise y Professional) de www.altova.com/es.

Tras instalar el paquete de integración podrá empezar a usar XMLSpy en el entorno de Visual Studio.

Nota importante

Debe usar el paquete de integración correspondiente a su versión de XMLSpy (la versión actual es 2023). El paquete de integración no es específico de ninguna versión, por lo que se puede usar tanto para la edición Enterprise como Professional.

© 2016-2022 Altova GmbH

25.2 Diferencias entre XMLSpy y XMLSpy para Visual Studio

Este apartado enumera las diferencias entre el complemento de XMLSpy para Visual Studio y la versión independiente de XMLSpy. Primero describimos las funciones que no están disponibles en la versión de XMLSpy para Visual Studio y después explicamos las diferencias entre las dos versiones.

- Funciones no disponibles en el complemento de XMLSpy para Visual Studio 1125
- Menús adicionales de XMLSpy en Visual Studio 1125
- Ayudantes de entrada en Visual Studio 1125
- Funciones idénticas, comandos distintos 1126
- Comandos de XMLSpy como comandos de Visual Studio 1126

Funciones no disponibles en el complemento de XMLSpy para Visual Studio

Estas funciones de XMLSpy no están disponibles en la versión para Visual Studio:

- El entorno de scripting no está disponible por ahora (**Herramientas | Opciones de XMLSpy | Scripting**).
- No puede abrir archivos en una ventana separada del explorador (opción de **Herramientas | Opciones | Vista**). Esto significa que la vista Texto y la vista Explorador están siempre en la misma ventana.
- Los iconos de estado de texto de la vista Authentic 597.
- Ninguna de las funciones de control de código fuente.
- Ninguna función de comparación (disponibles en el menú **Herramientas** de la versión independiente de XMLSpy).

Menús adicionales de XMLSpy en Visual Studio

Estos comandos aparecen sólo en la versión de XMLSpy para Visual Studio:

- **Ver | Ventanas de herramientas de XMLSpy**
- **Ver | Vista XMLSpy**
- **XMLSpy** (incluye las opciones de menú de Recursos globales y poder usar los temas de XMLSpy)
- **Herramientas | Opciones de XMLSpy**

Nota: en Visual Studio 2019 y versiones posteriores se puede acceder a la funcionalidad de XMLSpy desde el menú **Extensiones** de Visual Studio. En las versiones anteriores encontrará estas funcionalidades en los menús de nivel superior de Visual Studio.

Ayudantes de entrada (Ventanas de herramientas en Visual Studio)

Los ayudantes de entrada de XMLSpy se llaman *Ventanas de herramientas* en Visual Studio. Es necesario tener en cuenta estos aspectos:

- Puede arrastrar las ventanas de los ayudantes de entrada y colocarlas en cualquier posición del entorno de desarrollo.
- Puede seguir personalizando la interfaz si hace clic con el botón derecho en una pestaña de un ayudante de entrada. El menú contextual que aparece incluye opciones para acoplar la ventana, ocultarla, ocultarla automáticamente o hacerla flotante.

Funciones idénticas, comandos distintos

Algunas funciones de XMLSpy están disponibles en Visual Studio pero el comando utilizado para acceder a ellas se llama de otra manera. Se trata de estas funciones:

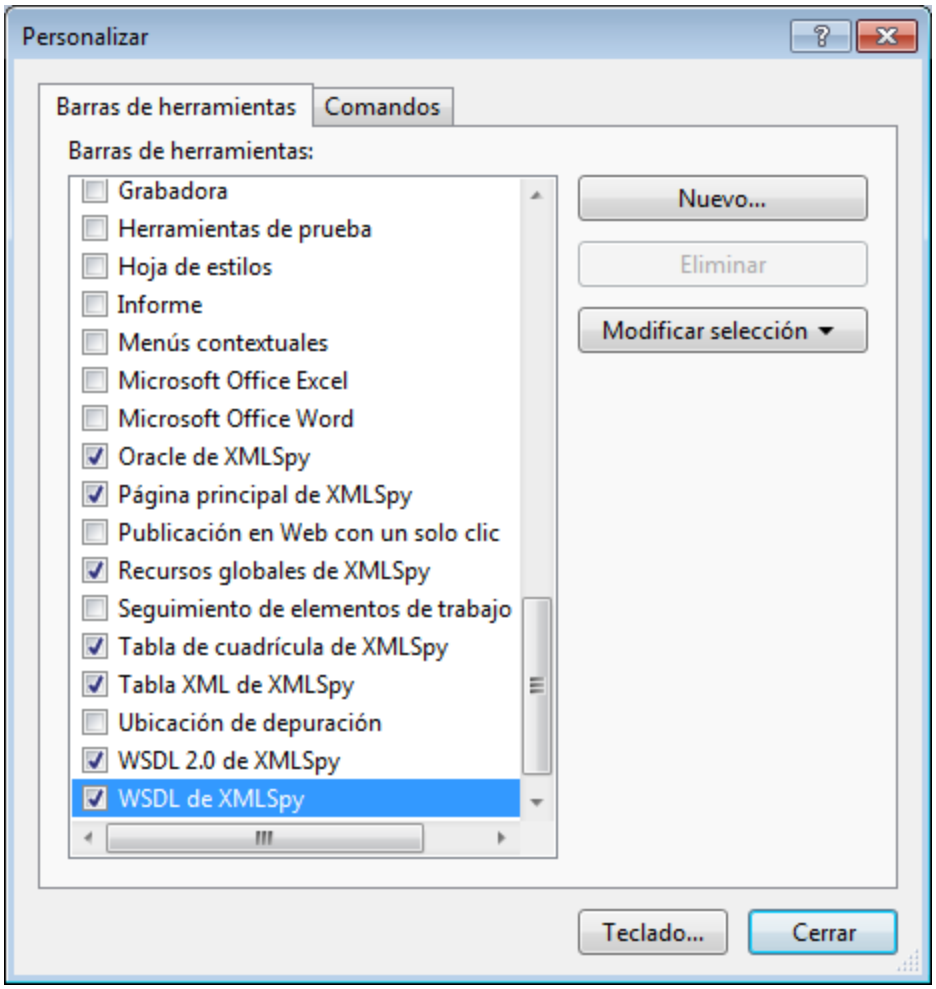
XMLSpy	***Visual Studio***	***Función***
Archivo | Abrir | Cambiar a URL	Archivo | Abrir | Sitio web	Abre un archivo desde una URL
Cambiar a URL | Guardar	Archivo | Guardar el archivo de XMLSpy en una URL	Guarda el archivo en una URL

Comandos de XMLSpy como comandos de Visual Studio

Algunos comandos de XMLSpy son comandos de Visual Studio en la interfaz de Visual Studio. Se trata de estos comandos:

- ***Deshacer/Rehacer:*** estos comandos de Visual Studio afectan a todas las acciones realizadas en el entorno de Visual Studio.
- ***Proyectos:*** los proyectos de XMLSpy se consideran proyectos de Visual Studio.
- ***Personalizar las barras de herramientas/Personalizar los comandos:*** las pestañas *Barras de herramientas* y *Comandos* (*ver imagen siguiente*) del cuadro de diálogo Personalizar (**Herramientas | Personalizar**) incluye tanto los comandos de Visual Studio como los de XMLSpy.

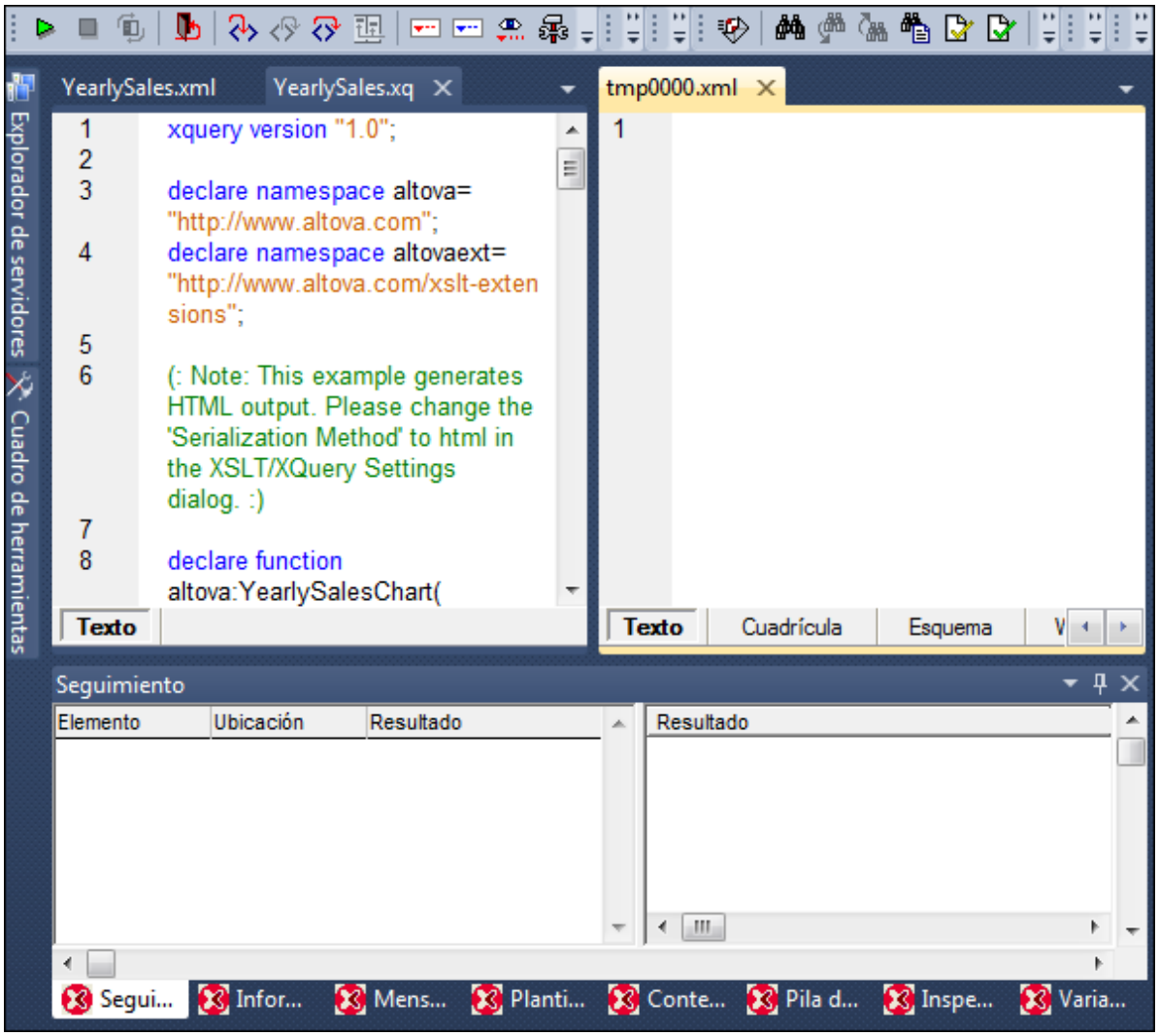
© 2016-2022 Altova GmbH



- ***Vistas:*** en el menú **Ver** los comandos **Ventanas de herramientas de XMLSpy** y **Vista XMLSpy** incluyen opciones para activar/desactivar las ventanas de los ayudantes de entrada y otras barras laterales, para cambiar de vista de edición y para activar/desactivar determinadas guías de edición.
- ***Ayuda de XMLSpy:*** este menú de XMLSpy es un submenú del menú **Ayuda** de Visual Studio.

25.3 Depuradores de XMLSpy en Visual Studio

XMLSpy incluye un depurador XSLT/XQuery (Enterprise y Professional Edition) y un depurador SOAP (Enterprise Edition). El proceso del depurador exige presentar varios archivos en la vista del depurador (por ejemplo, un archivo XML, un archivo XSLT y el archivo de resultados). En Visual Studio todos estos archivos se presentan en tres pestañas diferentes. Si desea ordenar las pestañas de otra manera puede separar las pestañas. Para ello haga clic en la barra de título de una pestaña y arrástrela hasta la posición deseada (*ver imagen siguiente*).



Nota: las imágenes y capturas de pantalla de esta sección son de Visual Studio 2010. Si utiliza otra versión de Visual Studio, puede que encuentre diferencias entre estas imágenes y su versión.

© 2016-2022 Altova GmbH

26 XMLSpy en Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto que incorpora varios tipos de aplicaciones en forma de complementos. Gracias al complemento de XMLSpy para Eclipse puede usar todas las funciones de XMLSpy desde la plataforma Eclipse 2022-09, 2022-06, 2022-03, 2021-12. El complemento de XMLSpy para Eclipse está disponible para todas las plataformas Windows. Las versiones de Eclipse compatibles son: 2022-09, 2022-06, 2022-03, 2021-12.

En esta sección explicamos

- Cómo instalar el paquete de integración de XMLSpy para Eclipse e integrar XMLSpy en Eclipse 1130
- La perspectiva de <%APPNAME% en Eclipse 1132
- Otros puntos de entrada de XMLSpy en Eclipse 1135
- Perspectivas del depurador de XMLSpy 1137

Nota: las funciones de control de código fuente de XMLSpy no están disponibles en la versión de XMLSpy para Eclipse.

26.1 Instalación del complemento de XMLSpy para Eclipse

Requisitos

- Eclipse 2022-09, 2022-06, 2022-03, 2021-12 (http://www.eclipse.org).
- Un Java JRE/JDK para la plataforma de 64 bits
- XMLSpy Enterprise o Professional Edition de 64 bits

Nota: todos los requisitos anteriores deben corresponder a la plataforma de 64 bits. El complemento para Eclipse de versiones anteriores de 32 bits ya no es compatible, aunque es posible que aún funcione.

Una vez haya comprobado que cumple con los requisitos de la lista anterior, puede instalar el paquete de integración de XMLSpy de 64 bits para integrar la aplicación en Eclipse. La integración se puede llevar a cabo al instalar el paquete de integración o de forma manual desde Eclipe una vez haya instalado el paquete de integración. El paquete de integración de XMLSpy se puede descargar en https://www.altova.com/es/components/download.

Nota: debe cerrar Eclipse para poder instalar o desinstalar el paquete de integración de XMLSpy.

Instalar el complemento de XMLSpy para Eclipse

Puede integrar el complemento de XMLSpy en Eclipse ante la instalación del paquete de integración de XMLSpy. Para ello:

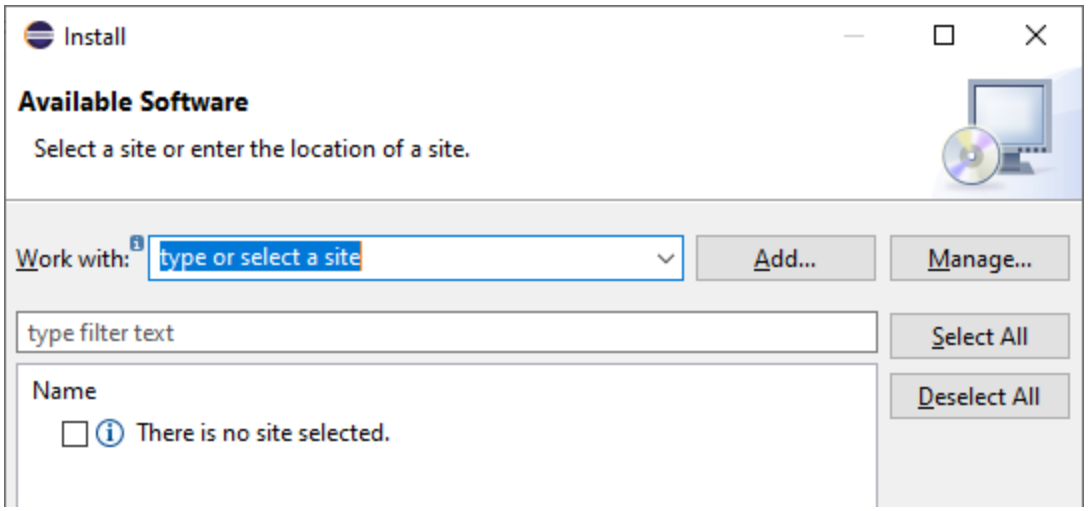
1. Ejecute el paquete de integración de XMLSpy para iniciar el asistente de instalación.
2. Cuando el instalador se lo pida seleccione *Instalar el complemento para Eclipse* y haga clic en **Aceptar**.
3. Cuando le pida que escoja cómo quiere integrar el complemento de XMLSpy en Eclipse seleccione *Permitir integración de XMLSpy en Eclipse* y navegue hasta el directorio en el que está el ejecutable de Eclipse (`eclipse.exe`).
4. Haga clic en **Siguiente** y complete la instalación.

Si elige la integración automática los menús y la perspectiva de XMLSpy se habilitarán en Eclipse la siguiente vez que lo inicie.

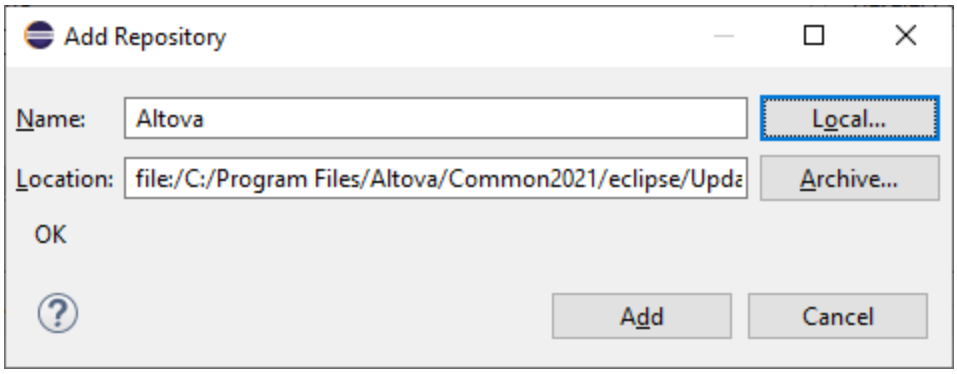
Integrar el complemento de XMLSpy para Eclipse manualmente

1. En Eclipse seleccione el comando de menú **Ayuda | Instalar software nuevo**.
2. En el cuadro de diálogo Instalar haga clic en **Agregar**.

© 2016-2022 Altova GmbH



3. En el cuadro de diálogo "Agregar repositorio" haga clic en **Local**. Navegue hasta la carpeta `C:\Program Files\Altova\Common2023\eclipse\UpdateSite` y selecciónsela. Elija un nombre para el sitio (por ejemplo, "Altova").



4. Repita los pasos 2-3 del punto anterior pero esta vez elija la carpeta `C:\Program Files\Altova\XMLSpy2023\eclipse\UpdateSite` y un nombre como "Altova XMLSpy".
5. En el cuadro de diálogo seleccione *Solo sitios locales*. A continuación seleccione la carpeta "Altova category" y haga clic en **Siguiente**.
6. Revise los elementos que va a instalar y haga clic en **Siguiente** para continuar.
7. Marque la casilla correspondiente para aceptar el acuerdo de licencia.
8. Haga clic en Finalizar para terminar la instalación.

Nota: si hay algún problema con el complemento (iconos que faltan, por ejemplo), pruebe a iniciar Eclipse desde la línea de comandos con el elemento flag `-clean`.

26.2 Perspectiva de XMLSpy en Eclipse

En Eclipse una perspectiva es una vista de la interfaz del usuario configurada con función de operador. Cuando el complemento de XMLSpy está integrado en Eclipse, se crea automáticamente una perspectiva XMLSpy predeterminada. Esta perspectiva es una interfaz gráfica del usuario que incluye los elementos de la interfaz de XMLSpy: sus vistas de edición, menús, ayudantes de entrada y otras barras laterales.

Al abrir un tipo de archivo asociado a XMLSpy , el archivo se puede editar en la perspectiva XMLSpy. También puede abrir archivos de otro tipo en otra perspectiva de Eclipse. Además, puede cambiar de perspectiva en todo momento, lo cual permite editar o procesar el archivo en otro entorno. Por tanto, las perspectivas tienen dos ventajas:

1. Permiten cambiar el entorno de trabajo para el archivo activo
2. Y permiten cambiar de un archivo a otro sin necesidad de abrir un entorno de desarrollo nuevo (el entorno asociado está disponible en la perspectiva)

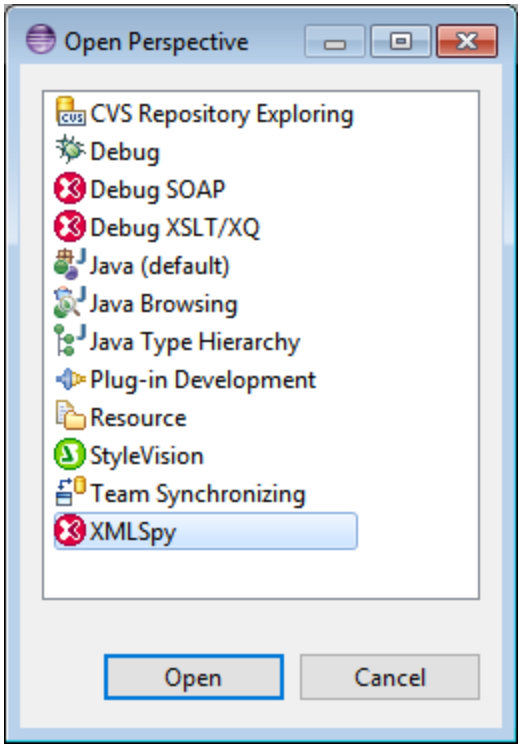
Para trabajar con la perspectiva XMLSpy es necesario:

- Cambiar a la perspectiva XMLSpy.
- Configurar las preferencias de la perspectiva XMLSpy.
- Personalizar la perspectiva XMLSpy.

Cambiar a la perspectiva XMLSpy

En Eclipse seleccione el comando **Window | Perspective | Open Perspective | Other**. En el diálogo que aparece (*imagen siguiente*) seleccione **XMLSpy** y haga clic en **OK**.

© 2016-2022 Altova GmbH



La ventana vacía o el documento activo tendrán ahora la perspectiva XMLSpy. Así es como se cambia de perspectiva desde el menú. Para cambiar más rápido de una perspectiva a otra se puede incluir la perspectiva deseada en el submenú **Open Perspective**, sobre la opción **Other**. Esto se hace en el cuadro de diálogo de personalización.

También puede cambiar de perspectiva al abrir un archivo o activar un archivo. La perspectiva asociada al tipo de archivo se abre automáticamente al abrir el archivo por primera vez. Antes de que cambie la perspectiva, aparece un cuadro de diálogo donde debe indicar si desea abrir automáticamente la perspectiva asociada a este tipo de archivo..

Marque la casilla *Do Not Ask Again* (No volver a preguntar) si desea asociar la perspectiva a ese tipo de archivo sin tener que confirmarlo cada vez que abra un archivo de este tipo. Después haga clic en **OK**.

Configurar las preferencias de la perspectiva XMLSpy

Para abrir el cuadro de diálogo de preferencias, seleccione el comando **Window | Preferences**. En la lista de perspectivas del panel de la izquierda, seleccione XMLSpy y después las opciones que desea usar. Para terminar haga clic en **OK**.

Las preferencias de una perspectiva incluyen:

- una opción para cambiar automáticamente de perspectiva al abrir un archivo de un tipo determinado (*ver párrafo anterior*)
- opciones para incluir o excluir algunas barras de herramientas de XMLSpy
- Acceso a las opciones de XMLSpy.

Personalizar la perspectiva XMLSpy

En Eclipse puede configurar qué accesos directos y comandos aparecen en cada perspectiva. Para abrir el cuadro de diálogo para personalizar la perspectiva (*imagen siguiente*), active la perspectiva que desea personalizar (en este caso la perspectiva XMLSpy) y seleccione el comando **Window | Perspective | Customize Perspective**.

- En las pestañas Tool Bar Visibility y Menu Visibility puede indicar qué barras de herramientas y menús quiere que se vean.
- En la pestaña Action Set Availability puede añadir conjuntos de acciones a sus menús principales y a la barra de herramientas. Si quiere habilitar un grupo de acciones marque la casilla correspondiente.
- En la pestaña Shortcuts del cuadro de diálogo "Customize Perspective" puede definir accesos directos para los submenús. Seleccione el submenú en el cuadro combinado Submenus, después seleccione una categoría de accesos directos y por último seleccione los accesos directos que desea incluir en la perspectiva.

Haga clic en **Apply & Close** para terminar de personalizar la perspectiva y que los cambios surtan efecto.

© 2016-2022 Altova GmbH

26.3 Otros puntos de entrada de XMLSpy en Eclipse

Además de la perspectiva de XMLSpy existen otros dos puntos de entrada desde los que puede acceder a las funciones de XMLSpy:

- El menú de XMLSpy
- La barra de herramientas de XMLSpy

El menú de XMLSpy en Eclipse

El menú para Eclipse de **XMLSpy** contiene comandos de XMLSpy para acceder a las funciones correspondientes de XMLSpy. Estos comandos aparecen en varios menús de la versión independiente de XMLSpy.

En la parte inferior de este menú hay comandos que sirven para definir el tema de la perspectiva de XMLSpy en Eclipse.

Barra de herramientas de XMLSpy en Eclipse

La barra de herramientas de XMLSpy contiene estos botones:



Estos comandos sirven para:

- abrir la Ayuda de XMLSpy
- acceder a los comandos de XMLSpy (aunque también puede acceder a ellos desde el menú **XMLSpy**)

Nota: los comandos de la barra de herramientas no son compatibles. Si ha configurado en XMLSpy algún comando de la barra de herramientas que ejecute un comando o script, ese comando no estará disponible en el complemento.

Formatos de archivo de XMLSpy y comportamiento de las vistas Eclipse

Cuando en Eclipse están activos determinados tipos de archivos conocidos por XMLSpy, aparecen las vistas Elementos, Atributos y Entidades adaptadas al tipo de archivo activo. Por ejemplo, si el archivo activo es un archivo .css, la vista Elementos pasa a llamarse Esquema CSS. En esta tabla puede ver las correspondencias entre el tipo de archivo y el nombre que toman las vistas:

Si el formato del archivo activo es...	La vista Elementos pasa a llamarse...	La vista Atributos pasa a llamarse...	La vista Entidades pasa a llamarse...
.css	Esquema CSS	Propiedades CSS	Elementos HTML
.xquery, .xq	Palabras clave XQuery	Variables XQuery	Funciones XQuery
.xsd	Componentes	Detalles	Facetas

Si cierra una de estas vistas, puede volver a restaurarla con el comando de menú **Window | Show View** de Eclipse. Sin embargo, recuerde que las vistas aparecen en este menú con su nombre genérico (es decir,

Elementos, Atributos y Entidades). Por ejemplo, si cierra la vista Esquema CSS, cuando intente restaurarla desde el menú **Window | Show** deberá seleccionar la vista Elementos (**Show | View | Elementos**).

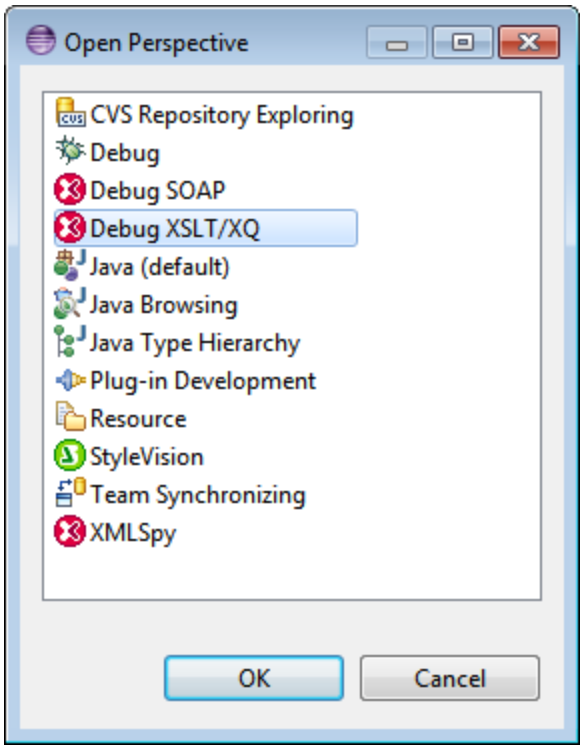
Otra manera de restaurar la vista es restaurar los valores predeterminados de la perspectiva XMLSpy con el comando **Window | Reset Perspective**.

© 2016-2022 Altova GmbH

26.4 Perspectivas del depurador de XMLSpy

En la edición Enterprise Edition de XMLSpy hay dos depuradores: el depurador XSLT/XQuery y el depurador SOAP. La edición Professional Edition de XMLSpy incluye sólo el depurador XSLT/XQuery. Eclipse ofrece perspectivas para los depuradores que incluye su edición de XMLSpy.

Para cambiar a una perspectiva de depurador, seleccione el comando **Window | Open Perspective | Other**. En el cuadro de diálogo que aparece (*imagen siguiente*) seleccione el depurador que desea abrir (p. ej. Depurar XSLT o XQ) y haga clic en **OK**.



Como resultado la ventana vacía o el documento activo toma la perspectiva del depurador seleccionado. Así es como se cambia de perspectiva desde el menú. Para cambiar más rápido de perspectiva, puede incluir la perspectiva que desea utilizar en el submenú **Open Perspective**, sobre la opción **Other**. Esto puede configurarse en el cuadro de diálogo de personalización.

Para más información sobre cómo usar los depuradores, consulte las secciones correspondientes de la documentación:

- XSLT 490
- XQuery 505
- WSDL y SOAP 775

27 Generador de código

XMLSpy incluye un generador integrado que puede generar archivos de clases en Java, C++ o C# a partir de esquemas XML. El código generado consiste en bibliotecas contenedoras de esquemas rigurosas que permiten crear aplicaciones de software que procesen datos XML. Estas bibliotecas permiten trabajar trabajar con datos XML de forma programática usando tipos generados a partir del esquema, sin prestar demasiada atención a la API XML subyacente.

El editor de esquemas de XMLSpy es una herramienta de modelado y diseño de prototipos de software que permite crear rápidamente prototipos de alto nivel de aplicaciones XML en XML Schema y después generarlas automáticamente. Para conciliar los cambios en el modelo de contenido del esquema XML de una aplicación con una implementación de software basta con volver a ejecutar el generador de código.

El código generado es compatible con estas operaciones:

- leer archivos XML en una representación en memoria de un Modelo de Objetos del Documento (DOM)
- escribir archivos XML desde una representación de un DOM en un sistema de archivos
- convertir cadenas en estructuras XML DOM y viceversa.

El código generado se expresa en los lenguajes de programación C++, Java o C#.

Lenguaje de destino	**C++**	**C#**	**Java**
Entornos de desarrollo	Microsoft Visual Studio 2013, 2015, 2017, 2019, 2022	Microsoft Visual Studio 2013, 2015, 2017, 2019, 2022 Marcos de destino: • .NET Framework • .NET Core 3.1 • .NET 5.0 • .NET 6.0	Java SE JDK 8 u 11 (incluido OpenJDK) Eclipse 4.4 o superior Apache Ant (compile el archivo.xml)
Implementaciones XML DOM	MSXML 6.0 Apache Xerces 3	System.Xml	JAXP

C++

Puede configurar si el resultado generado en C++ debe usar MSXML 6.0 o Apache Xerces 3. XMLSpy genera archivos completos de proyecto (.vcproj) y solución (.sln) para todas las versiones de Visual Studio compatibles (véase la tabla de más arriba). El código generado puede ser compatible con MFC.

Requisitos:

1. Para compilar el código C++ generado debe tener Windows SDK instalado en su equipo.
2. Para usar Xerces 3 para C++ debe instalarlo y compilarlo según las instrucciones de http://xerces.apache.org/xerces-c/. Asegúrese de que añade la variable de entorno XERCES3 que apunta al directorio en el que instaló Xerces, por ejemplo `C:\xerces-c-3.2.2`. Además, la variable de

© 2016-2022 Altova GmbH

entorno PATH debe incluir la ruta de la ubicación donde se encuentran los archivos binarios Xerces, por ejemplo: `%XERCES3%\bin`.

3. Al compilar código C++ para Visual Studio mientras usa una biblioteca Xerces precompilada para Visual C++ debe cambiar la configuración del compilador en todos los proyectos de la solución:

   a) Seleccione todos los proyectos en el explorador de soluciones.
   b) En el menú **Proyecto** haga clic en **Propiedades**.
   c) Haga clic en **Propiedades de la configuración | C/C++ | Lenguaje**.
   d) En la lista de configuraciones seleccione *Todas las configuraciones*.
   e) Cambie **Treat wchar_t as Built-in Type** por **No (/Zc:wchar_t-)**.

C#

El código C# generado lo puede usar cualquier lenguaje de programación apto para .NET, como VB.NET, Managed C++ o J#. Los archivos de proyecto se pueden generar para todas las versiones compatibles de Visual Studio (véase la tabla de más arriba).

Java

El código Java generado se escribe conforme a la API Java para procesamiento XML (JAXP) e incluye un archivo de compilación Ant y archivos de proyecto para las versiones compatibles de Java y Eclipse (véase la tabla de más arriba).

Resultados generados

La carpeta de destino designada para el código generado incluye todas las bibliotecas y los archivos necesarios para manipular archivos XML de forma programática, es decir:

- Bibliotecas estándar de Altova
- Bibliotecas contenedoras de esquemas
- Una aplicación de prueba vacía con código fuente de muestra. Se trata de una aplicación compilable que llama a un método Example() vacío. Puede añadir su código de muestra a este método para hacer pruebas de forma fácil y rápida con la biblioteca que acaba de generar.

Plantillas de generación de código

El código generado se puede personalizar con un archivo de plantilla[1241] llamado SPL (Spy Programming Language). SPL permite controlar íntegramente asignaciones de tipos de datos integrados en el esquema XML con tipos de datos primitivos de un lenguaje de programación en particular. Esto le permite compilar sus propias plantillas para automatizar la generación de prácticamente cualquier otro formato, por ejemplo: EJB, archivos WSDL, scripts SQL o código ASP o WML.

27.1 Novedades

Versión 2022 R2

- Compatibilidad con Visual Studio 2022.
- Compatibilidad con .NET 6.0.

Versión 2021 R2

- Al generar código en C# puede seleccionar .NET Core 3.1, .NET 5.0 o .NET 6.0 como marco de destino desde las opciones del generador de código[1239] (además de la compatibilidad con proyectos .NET Framework que ya existe).

Versión 2020 R2

- El código generado para las bibliotecas contenedoras de esquemas XML ahora ofrecen un mayor control sobre los espacios de nombres y los prefijos de los elementos. Por ejemplo, hemos introducido métodos nuevos para declarar o sobrescribir los espacios de nombres de un elemento o para anexar un elemento con un espacio de nombres prefijado. Consulte Ejemplo: orden de compra[1179] .

Versión 2020

- Compatibilidad con Visual Studio 2019.
- Fin de compatibilidad con Visual Studio 2008.

Versión 2018

- Compatibilidad con Microsoft Visual Studio 2013, 2015, 2017.
- Fin de compatibilidad con Visual Studio 2005 y Xerces 2.x.

Versión 2014
- Eliminación de la opción de modo de compatibilidad para la generación de código.

Versión 2011
Corrección de fallos y mejoras.

Versión 2010 R3
- Compatibilidad con Microsoft Visual Studio 2010.
- Compatibilidad con MSXML 6.0 en el código C++ generado.
- Compatibilidad con componentes de destino de 64 bits en proyectos C++ y C#.

Versión 2010 R2
- Generación de código C++ para la plataforma Linux.

Versión 2010
- Las facetas "enumeration" de XML Schema ya están disponibles como constantes simbólicas en las clases generadas (usando plantillas de 2007r3).

Versión 2009 sp1

© 2016-2022 Altova GmbH

- Compatibilidad con Apache Xerces 3.x (las versiones Xerces 2.6.x y posteriores siguen siendo compatibles).

Versión 2008 R2

- Generación de archivos de proyectos de Visual Studio 2008 para C# y C++.

Versión 2008

- Las plantillas SPL 2007 R3 se han mejorado:
    - Ahora puede eliminar elementos uno por uno
    - Ahora puede acceder a metadatos del esquema (p. ej. nombres de elementos, facetas, enumeraciones, apariciones, etc.)
    - Los tipos complejos derivados por extensión ahora se generan como clases derivadas

Versión 2007 R3

El generador de código se ha rediseñado para la versión 2007 Release 3 con el objetivo de simplificar el uso del código generado, reducir el volumen del código y mejorar el rendimiento.

- Gestión de documentos y nodos XML con propiedad explícita para evitar pérdida de memoria y habilitar subprocesos múltiples
- Sintaxis nueva para evitar colisiones de nombres
- Tipos de datos nuevos para un uso más sencillo y un mayor rendimiento (uso de tipos nativos cuando sea posible, control de valores null...)
- Los atributos ya no se generan como colecciones
- El contenido de elementos simples ahora también se considera un atributo especial
- Un modelo de objetos interno nuevo (importante para las plantillas SPL personalizadas)
- Modo de compatibilidad para generar código en el estilo de versiones anteriores

27.2 Generar código fuente a partir de esquemas XML o DTD

Con el generador de código de XMLSpy puede generar código de programa C#, C++ o Java a partir de esquemas XML o documentos DTD. Las bibliotecas contenedoras de esquemas generadas se pueden integrar en otras aplicaciones para leer, modificar o escribir documentos XML mediante programación.

Generar código de programa

1. Abra el esquema para el que desea generar código fuente.
2. Seleccione el comando de menú **DTD/Schema | Generar código de programa**.
3. En la pestaña *Elegir plantilla* debe configurar las opciones de generación de código (*ver* *Opciones de generación de código* 1239).
4. Haga clic en **Aceptar**. Aparece el cuadro de diálogo "Buscar carpeta".
5. Seleccione la carpeta de destino y haga clic en **Aceptar**.
6. Aparece un mensaje preguntando si desea abrir el proyecto recién creado en Microsoft Visual Studio. Haga clic en **Sí**.
7. Si generó código Java, entonces se le pregunta si desea abrir el directorio de salida pertinente.

Cuando genera código a partir de un esquema XML o de un documento DTD, XMLSpy crea estas bibliotecas:

C++ o C#	Java	Objetivo
Altova	`com.altova`	Biblioteca base que contiene compatibilidad con Common Language Runtime, idéntica para todos los esquemas.
AltovaXML	`com.altova.xml`	Biblioteca base que contiene compatibilidad con Common Language Runtime para XML, idéntica para todos los esquemas.
MiEsquema	*`com.MiEsquema`*	Biblioteca que contiene declaraciones generadas a partir del esquema de entrada, llamado archivo de esquema o DTD. Esta biblioteca es un contenedor DOM (W3C Document Object Model) que permite leer, modificar y crear documentos XML de forma sencilla y segura. Todos los datos están dentro del DOM y hay métodos para extraer y actualizar datos y crear datos nuevos en el DOM. El código C++ generado es compatible con Microsoft MSXML o Apache Xerces 3. Por lo general, la sintaxis para usar el código generado es parecida en ambas implementaciones salvo por algunas diferencias (por ejemplo, Xerces admite más funciones de sobrecarga). El código C# generado utiliza la biblioteca System.XML estándar de .NET como implementación DOM subyacente.

 © 2016-2022 Altova GmbH

		El código Java generado utiliza JAXP como interfaz DOM subyacente.
MiEsquemaTest	`com.MiEsquemaTest`	El código generado también incluye una aplicación de prueba básica, que se llama igual que el esquema utilizado (p. ej. `MiEsquemaTest`). Se trata de una aplicación compilable que llama a un método Example() vacío. Puede añadir su código de prueba a este método para probar la biblioteca recién generada de forma fácil y rápida.

Cuando cree prototipos de aplicaciones a partir de esquemas XML que cambien con frecuencia, a veces será necesario generar código una y otra vez en el mismo directorio para que los cambios en el esquema se reflejen inmediatamente en el código. Recuerde que la aplicación de prueba que se genera y las bibliotecas de Altova se sobrescribirán cada vez que genere código en el mismo directorio de destino. Por tanto, recuerde que no debe añadir código a la aplicación de prueba que se genera, sino que debe integrar las bibliotecas de Altova en el proyecto (ver *Integrar bibliotecas contenedoras de esquemas* 1152).

Espacios de nombres y generación de nombres

XMLSpy genera clases que se corresponden con todos los elementos o tipos complejos declarados que redefinen todos los tipos complejos del esquema XML, conservando la derivación de clases tal y como definan las extensiones de tipos complejos del esquema. Cuando se trate de esquemas complejos que importen componentes de esquemas desde varios espacios de nombres distintos, XMLSpy conserva esta información generando los espacios de nombres C#/C++ o los paquetes Java correspondientes.

Por lo general, el generador de código intenta conservar los nombres del esquema original para los espacios de nombres, clases y miembros generados. En el lenguaje de destino los caracteres no válidos que se encuentren en los identificadores se reemplazan con "_". A los nombres que estén en conflicto con otros nombres o palabras reservadas se les anexa un número para que sean nombres únicos. Si cambia la configuración predeterminada de la plantilla SPL 1241, puede que la generación de nombres se vea afectada.

Los espacios de nombres del esquema XML se convierten en paquetes en Java o en espacios de nombres en C#/C++, usando siempre el prefijo de espacios de nombres del esquema como espacio de nombres del código. La biblioteca completa se adjunta en un paquete o espacio de nombres derivado del nombre del archivo de esquema. Así podrá usar varias bibliotecas generadas en un programa sin que haya conflictos entre nombres.

Tipos de datos

XML Schema tiene un modelo de tipos de datos más complejo que Java, C# o C++. El generador de código convierte los tipos XML Schema integrados en tipos primitivos propios de cada lenguaje de programación o en clases derivadas con la biblioteca de Altova. Los tipos complejos y derivados definidos en el esquema se convierten en clases en la biblioteca que se genera. Las facetas de enumeración de tipos simples se convierten en constantes simbólicas.

La asignación entre tipos simples se puede configurar en la Referencia de SPL 1241.

Si los archivos de instancia XML utilizan tipos de esquema relacionados con horas y duración, estos se convierten en clases nativas de Altova en el código generado. Para más información consulte estos apartados:

- Referencia de las clases generadas (C++) 1187
- Referencia de las clases generadas (C#) 1204
- Referencia de las clases generadas (Java) 1220

Para más información sobre la conversión de tipos y otros detalles propios de cada lenguaje consulte estos apartados:

- Bibliotecas contenedoras de esquemas (C++) 1145
- Bibliotecas contenedoras de esquemas (C#) 1148
- Bibliotecas contenedoras de esquemas (Java) 1150

Gestión de memoria

Una estructura DOM está compuesta por nodos que siempre son propiedad de un documento DOM concreto, aunque el nodo no forme parte del contenido del documento en ese momento. Todas las clases generadas son referencias a los nodos DOM que representan, no valores. Esto significa que si se asigna una instancia de una clase generada, el valor no se copia, sino que se crea una referencia adicional a los mismos datos.

Compatibilidad con XML Schema

Estas construcciones de XML Schema se traducen en el código:

a) Espacios de nombres XML

b) Tipos simples:

- Tipos de XML Schema integrados
- Tipos simples derivados por extensión
- Tipos simples derivados por restricción
- Facetas
- Enumeraciones
- Patrones

c) Tipos complejos

- Nodo anyType integrado
- Tipos complejos definidos por el usuario
- Derivados por extensión: se asignan a clases derivadas
- Derivados por restricción
- Contenido complejo
- Contenido simple
- Contenido mixto

Estas características de XML Schema avanzadas no son compatibles con las clases contenedoras generadas:

- Comodines: `xs:any` y `xs:anyAttribute`
- Modelos de contenido (sequence, choice, all). El compositor de nivel superior está disponible en el SPL 1241, pero las clases generadas no lo exigen.
- Valores predeterminados y fijos para atributos. Están disponibles en el SPL 1241, pero las clases generadas no los exigen ni establecen.
- Los atributos `xsi:type`, tipos abstractos. Cuando necesite escribir el atributo `xsi:type`, utilice el método `SetXsiType()` de las clases generadas.

© 2016-2022 Altova GmbH

- Tipos de unión: no todas las combinaciones son compatibles.
- Los grupos de sustitución son compatibles en parte solamente (se resuelven como "choice").
- Atributos `nillable="true"` y `xsi:nil`
- Restricciones de unicidad
- Restricciones de identidad (`key` y `keyref`)

27.2.1 Bibliotecas contenedoras de esquemas (C++)

Tipos de caracteres

El código C++ generado se puede compilar con o sin compatibilidad con Unicode. Dependiendo de la configuración elegida, los tipos `string_type` y `tstring` se definirán como `std::string` o `std::wstring` y estarán formados por caracteres estrechos o anchos. Para usar caracteres Unicode en su archivo XML que no se puedan representar con el conjunto de caracteres de 8 bits actual deberá habilitar la compatibilidad con Unicode. Deberá prestar especial atención a las macros `_T()`. Esta macro garantizan el correcto almacenamiento de las constantes de cadena (tanto si se compila para programas Unicode como para programas no Unicode).

Tipos de datos

Esta es la correspondencia predeterminada de tipos XML Schema con tipos de datos C++:

XML Schema	C++	Observaciones
`xs:string`	`string_type`	`string_type` se define como `std::string` o `std:wstring`
`xs:boolean`	`bool`	
`xs:decimal`	`double`	C++ no tiene un tipo decimal por lo que se debe usar `double`.
`xs:float, xs:double`	`double`	
`xs:integer`	`__int64`	`xs:integer` tiene un rango ilimitado que se corresponde con `__int64` para mayor eficacia.
`xs:nonNegativeInteger`	`unsigned __int64`	ver anterior
`xs:int`	`int`	
`xs:unsignedInt`	`unsigned int`	
`xs:dateTime, date, time, gYearMonth, gYear, gMonthDay, gDay, gMonth`	`altova::DateTime` 1187	
`xs:duration`	`altova::Duration` 1191	

XML Schema	C++	Observaciones
`xs:hexBinary and xs:base64Binary`	`std::vector<unsigned char>`	La codificación y decodificación de datos binarios se realiza automáticamente.
`xs:anySimpleType`	`string_type`	

Los tipos XML Schema que faltan en la lista son tipos derivados y corresponde al mismo tipo C++ que su correspondiente tipo base.

Clases generadas

Por cada tipo del esquema se genera una clase que contiene un miembro por cada atributo y elemento del tipo. Los miembros se llaman igual que los atributos o elementos del esquema original (en caso de conflicto se anexa un número). Para los tipos simples se generan operadores de asignación y conversión. Para los tipos simples con facetas enumeration, se pueden usar los métodos `GetEnumerationValue()` y `SetEnumerationValue(int)` junto con las constantes generadas para cada valor de enumeración. Además, el método `StaticInfo()` permite acceder a la información del esquema como uno de estos tipos:

`altova::meta::SimpleType` 1197
`altova::meta::ComplexType` 1195

Las clases generadas a partir de tipos complejos incluyen el método `SetXsiType()`, que permite configurar el atributo `xsi:type` del tipo. Este método puede ser muy útil a la hora de crear elementos de instancia XML de un tipo derivado.

Además de las clases para los tipos declarados en el esquema XML, se genera también una clase de documento ("CDoc"). Esta clase contiene todos los elementos raíz posibles como miembros, así como varios métodos más. Para más información sobre esta clase consulte `[SuEsquema]::[CDoc]` 1198.

Nota: el nombre real de la clase depende del nombre del esquema .xsd.

Por cada atributo o elemento miembro de un tipo de esquema se genera una clase nueva. Para más información sobre dichas clases consulte:

`[SuEsquema]::AtributoMiembro` 1202
`[SuEsquema]::ElementoMiembro` 1202

Nota: el nombre real de las clases depende del nombre del atributo o elemento del esquema.

Consulte también el apartado Ejemplo: usar las bibliotecas contenedoras del esquema 1155.

Control de errores

Los errores se notifican por medio de excepciones. Estas son las clases de excepción que están definidas en el espacio de nombres `altova`:

Clase	Clase base	Descripción

© 2016-2022 Altova GmbH

`Error`	`std::logic_error`	Error de lógica de programa interna (independiente de los datos de entrada).
`Exception`	`std::runtime_error`	Clase base para errores de tiempo de ejecución.
`InvalidArgumentsException`	`Exception`	Se llamó a un método con valores de argumento no válidos.
`ConversionException`	`Exception`	Excepción generada cuando falla una conversión de tipo.
`StringParseException`	`ConversionException`	Un valor del espacio léxico no se puede convertir en un espacio de valor.
`ValueNotRepresentableException`	`ConversionException`	Un valor del espacio de valor no se puede convertir en espacio léxico.
`OutOfRangeException`	`ConversionException`	Un valor de origen no se puede representar en el dominio de destino.
`InvalidOperationException`	`Exception`	Se intentó una operación que no es válida en el contexto dado.
`DataSourceUnavailableException`	`Exception`	Se encontró un problema al intentar cargar una instancia XML.
`DataTargetUnavailableException`	`Exception`	Se encontró un problema al intentar guardar una instancia XML.

Todas las clases de excepción contienen un texto de mensaje y un puntero que apunta a una posible excepción interna.

Método	Finalidad
`string_type message()`	Devuelve una descripción textual de la excepción.
`std::exception inner()`	Devuelve la excepción que causó esta excepción (si procede) o devuelve NULL.

Acceso a la información del esquema

La biblioteca generada permite acceder a información estática del esquema por medio de estas clases. Todos los métodos se declaran como `const`.

`altova::meta::Attribute` 1194
`altova::meta::ComplexType` 1195
`altova::meta::Element` 1196
`altova::meta::SimpleType` 1197

Los métodos que devuelven una de las clases de metadatos devuelven un objeto NULL si no existe la propiedad correspondiente.

27.2.2 Bibliotecas contenedoras de esquemas (C#)

Esta es la correspondencia predeterminada de tipos XML Schema con tipos de datos C#:

XML Schema	C#	Observaciones
xs:string	string	
xs:boolean	bool	
xs:decimal	decimal	xs:decimal tiene un rango y una precisión ilimitados y se corresponde decimal para una mayor eficacia.
xs:float, xs:double	double	
xs:long	long	
xs:unsignedLong	ulong	
xs:int	int	
xs:unsignedInt	uint	
xs:dateTime, date, time, gYearMonth, gYear, gMonthDay, gDay, gMonth	Altova.Types.DateTime 1204	
xs:duration	Altova.Types.Duration 1209	
xs:hexBinary and xs:base64Binary	byte[]	La codificación y decodificación de datos binarios se realiza automáticamente.
xs:anySimpleType	string	

Los tipos XML Schema que faltan en la lista son tipos derivados y corresponde al mismo tipo C# que su correspondiente tipo base.

Clases generadas

Por cada tipo del esquema se genera una clase que contiene un miembro por cada atributo y elemento del tipo. Los miembros se llaman igual que los atributos o elementos del esquema original (en caso de conflicto se anexa un número). Para los tipos simples se generan operadores de asignación y conversión. Para los tipos simples con facetas enumeration, se pueden usar los métodos `GetEnumerationValue()` y `SetEnumerationValue(int)` junto con las constantes generadas para cada valor de enumeración. Además, el método `StaticInfo()` permite acceder a la información del esquema como uno de estos tipos:

Altova.Xml.Meta.SimpleType 1213

Altova.Xml.Meta.ComplexType 1212

Las clases generadas a partir de tipos complejos incluyen el método `SetXsiType()`, que permite configurar el atributo `xsi:type` del tipo. Este método puede ser muy útil a la hora de crear elementos de instancia XML de un tipo derivado.

Además de las clases para los tipos declarados en el esquema XML, se genera también una clase de documento ("Doc"). Esta clase contiene todos los elementos raíz posibles como miembros, así como varios métodos más. Para más información sobre esta clase consulte [SuEsquema].[Doc] 1214 .

Nota: el nombre real de la clase depende del nombre del esquema .xsd.

Por cada atributo o elemento miembro de un tipo de esquema se genera una clase nueva. Para más información sobre dichas clases consulte:

[SuTipoEsquema].AtributoMiembro 1217
[SuTipoEsquema].ElementoMiembro 1218

Nota: el nombre real de las clases depende del nombre del atributo o elemento del esquema.

Control de errores

Los errores se notifican por medio de excepciones. Estas son las clases de excepción que están definidas en el espacio de nombres `Altova`:

Clase	Clase base	Descripción
`ConversionException`	`Exception`	Excepción generada cuando falla una conversión de tipo.
`StringParseException`	`ConversionException`	Un valor del espacio léxico no se puede convertir en espacio de valor.
`DataSourceUnavailableException`	`System.Exception`	Se encontró un problema al intentar cargar una instancia XML.
`DataTargetUnavailableException`	`System.Exception`	Se encontró un problema al intentar guardar una instancia XML.

Además se suelen usar estas excepciones .NET:

Clase	Descripción
`System.Exception`	Clase base para errores de tiempo de ejecución.
`System.ArgumentException`	Se llamó a un método con valores de argumento no válidos o falló una conversión de tipo.
`System.FormatException`	Un valor del espacio léxico no se puede convertir en espacio de valor.

Clase	Descripción
System.InvalidCastException	No se puede convertir un valor en otro tipo.
System.OverflowException	No se puede representar un valor de origen en el dominio de destino.

Acceso a la información del esquema

La biblioteca generada permite acceder a información estática del esquema por medio de estas clases.

Altova.Xml.Meta.Attribute[1211]
Altova.Xml.Meta.ComplexType[1212]
Altova.Xml.Meta.Element[1213]
Altova.Xml.Meta.SimpleType[1213]

Las propiedades que devuelven una de las clases de metadatos devuelven un objeto NULL si no existe la propiedad correspondiente.

27.2.3 Bibliotecas contenedoras de esquemas (Java)

Esta es la correspondencia predeterminada de tipos XML Schema con tipos de datos Java:

XML Schema	Java	Observaciones
xs:string	String	
xs:boolean	boolean	
xs:decimal	java.math.BigDecimal	
xs:float, xs:double	double	
xs:integer	java.math.BigInteger	
xs:long	long	
xs:unsignedLong	java.math.BigInteger	Java no tiene tipos sin firmar.
xs:int	int	
xs:unsignedInt	long	Java no tiene tipos sin firmar.
xs:dateTime, date, time, gYearMonth, gYear, gMonthDay, gDay, gMonth	com.altova.types.DateTime[1221]	
xs:duration	com.altova.types.Duration[1226]	

© 2016-2022 Altova GmbH

XML Schema	Java	Observaciones
xs:hexBinary and xs:base64Binary	byte[]	La codificación y decodificación de datos binarios se realiza automáticamente.
xs:anySimpleType	string	

Los tipos XML Schema que faltan en la lista son tipos derivados y corresponde al mismo tipo Java que su correspondiente tipo base.

Clases generadas

Por cada tipo del esquema se genera una clase que contiene un miembro por cada atributo y elemento del tipo. Los miembros se llaman igual que los atributos o elementos del esquema original (en caso de conflicto se anexa un número). Para los tipos simples se generan operadores de asignación y conversión. Para los tipos simples con facetas enumeration, se pueden usar los métodos `GetEnumerationValue()` y `SetEnumerationValue(int)` junto con las constantes generadas para cada valor de enumeración. Además, el método `StaticInfo()` permite acceder a la información del esquema como uno de estos tipos:

`com.altova.xml.meta.SimpleType` 1231
`com.altova.xml.meta.ComplexType` 1230

Las clases generadas a partir de tipos complejos incluyen el método `SetXsiType()`, que permite configurar el atributo `xsi:type` del tipo. Este método puede ser muy útil a la hora de crear elementos de instancia XML de un tipo derivado.

Además de las clases para los tipos declarados en el esquema XML, se genera también una clase de documento ("Doc"). Esta clase contiene todos los elementos raíz posibles como miembros, así como varios métodos más. Para más información sobre esta clase consulte `com.[SuEsquema].[Doc]` 1232.

Nota: el nombre real de la clase depende del nombre del esquema .xsd.

Por cada atributo o elemento miembro de un tipo de esquema se genera una clase nueva. Para más información sobre dichas clases consulte:

`com.[SuEsquema].[SuTipoEsquema].AtributoMiembro` 1235
`com.[SuEsquema].[SuTipoEsquema].ElementoMiembro` 1236

Nota: el nombre real de las clases depende del nombre del atributo o elemento del esquema.

Control de errores

Los errores se notifican por medio de excepciones. Estas son las clases de excepción que están definidas en el espacio de nombres `com.altova`:

Clase	Clase base	Descripción
SourceInstanceUnvailableException	Exception	Se encontró un problema al intentar cargar una instancia XML.
TargetInstanceUnavailableException	Exception	Se encontró un problema al intentar guardar una instancia XML.

Además se suelen usar estas excepciones Java:

Clase	Descripción
java.lang.Error	Error de lógica de programa interna (independiente de los datos de entrada).
java.lang.Exception	Clase base para errores de tiempo de ejecución.
java.lang.IllegalArgumentsException	Se llamó a un método con valores de argumento no válidos o falló una conversión de tipo.
java.lang.ArithmeticException	Excepción generada cuando falla una conversión de tipo numérico.

Acceso a la información del esquema

La biblioteca generada permite acceder a información estática del esquema por medio de estas clases.

com.altova.xml.meta.Attribute 1229
com.altova.xml.meta.ComplexType 1230
com.altova.xml.meta.Element 1231
com.altova.xml.meta.SimpleType 1231

Las propiedades que devuelven una de las clases de metadatos devuelven un objeto NULL si no existe la propiedad correspondiente.

27.2.4 Integrar las bibliotecas contenedoras de esquemas

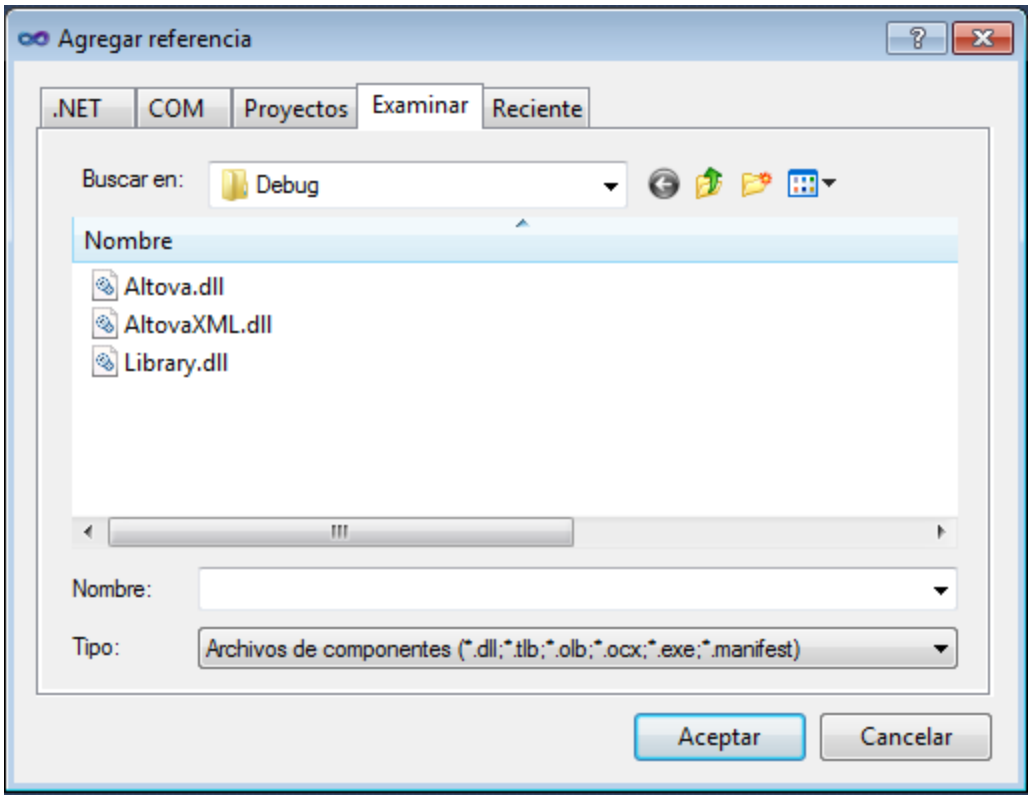
Para usar las bibliotecas de Altova en su proyecto basta con hacerles referencia o incluirlas en el proyecto (ver más abajo).

C#

Para integrar las bibliotecas de Altova en un proyecto C# actual:

© 2016-2022 Altova GmbH

1. Después de generar código a partir de un esquema con XMLSpy (p. ej. a partir de **YourSchema.xsd**), compile la solución **YourSchema.sln** que acaba de generar en Visual Studio. Esta solución está en una carpeta de proyecto que se llama igual que el esquema.
2. Haga clic con el botón derecho en el proyecto en Visual Studio y seleccione **Agregar referencia**.
3. En la pestaña *Examinar* busque las bibliotecas **Altova.dll**, **AltovaXML.dll** y **YourSchema.dll**, ubicadas en el directorio de salida de los proyectos generados (p. ej. **bin\Debug**).



C++

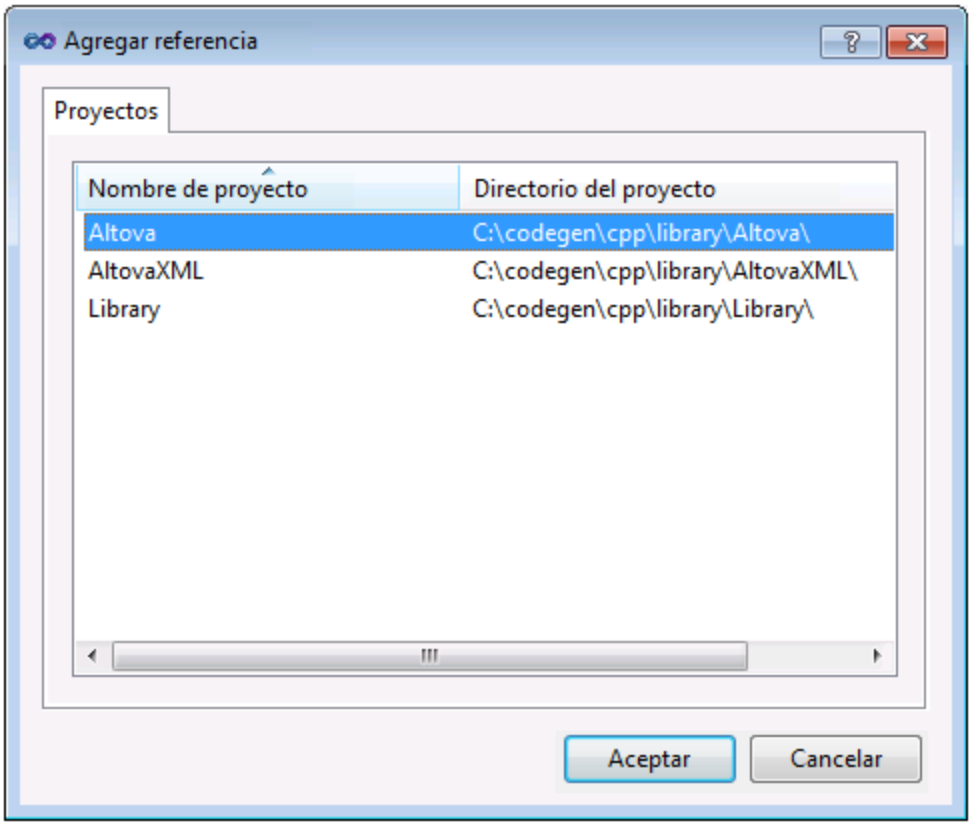
La forma más sencilla de integrar bibliotecas en un proyecto C++ es agregar a la solución los archivos de proyecto generados. Por ejemplo, imagine que generó código a partir de un esquema llamado **Library.xsd** y que seleccionó el directorio de destino **c:\codegen\cpp\library**. Las bibliotecas generadas en este caso están en estas rutas de acceso:

- c:\codegen\cpp\library\Altova.vcxproj
- c:\codegen\cpp\library\AltovaXML\AltovaXML.vcxproj
- c:\codegen\cpp\library\Library.vcxproj

Primero abra la solución **c:\codegen\cpp\library\Library.sln** y compílela en Visual Studio.

Después abra su solución de Visual Studio (en Visual Studio 2010, por ejemplo). Haga clic con el botón derecho en la solución y seleccione **Agregar | Proyecto existente**. Ahora añada uno a uno los archivos de proyecto enumerados más arriba. Tenga paciencia mientras Visual Studio analiza los archivos. Después haga clic con el botón derecho en el proyecto y seleccione **Propiedades**. En el cuadro de diálogo "Páginas de

propiedades" seleccione **Propiedades comunes | Framework y referencias** y después haga clic en **Agregar referencia nueva**. Añada estos proyectos: *Altova*, *AltovaXML* y *Library*.



Si necesita ayuda a la hora de usar las funciones de una biblioteca personal consulte la documentación de MSDN:

- Si tiene pensado generar bibliotecas estáticas, consulte https://msdn.microsoft.com/es-es/library/ms235627(v=vs.100).aspx
- Si tiene pensado generar bibliotecas dinámicas, consulte https://msdn.microsoft.com/es-es/library/ms235636(v=vs.100).aspx

La opción de generar bibliotecas estáticas o dinámicas se encuentra entre las opciones de generación de código (*véase* *Opciones del generador de código* 1239).

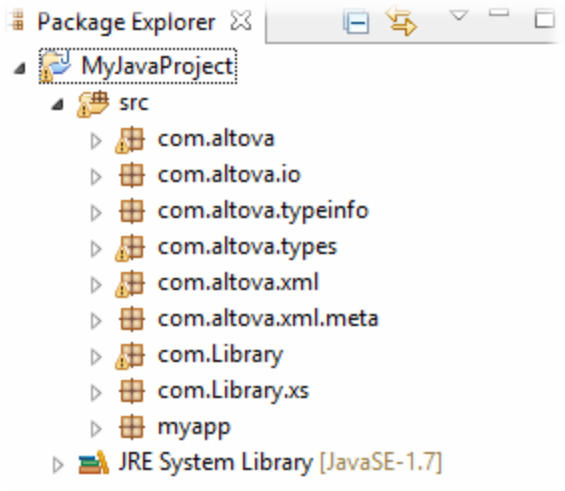
Java

Puede integrar sus paquetes de Altova en su proyecto Java copiando el directorio **com** del código generado en el directorio que almacena los paquetes fuente de su proyecto Java (p. ej. **C:\Workspace\MyJavaProject\src**). Por ejemplo, imagine que generó código en el directorio **c:\codegen\java\library**. Las clases de Altova generadas en este caso están en **c:\codegen\java\library\com**.

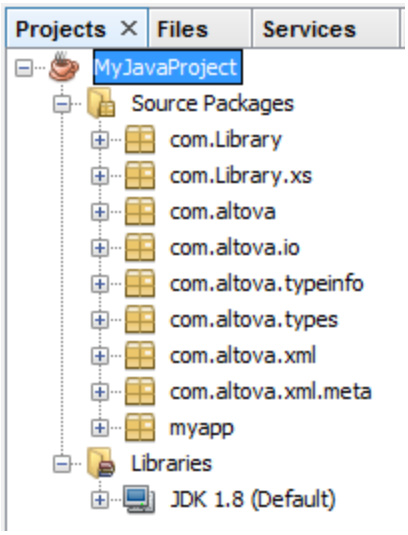
© 2016-2022 Altova GmbH

Tras copiar las bibliotecas actualice el proyecto (en Eclipse seleccione el proyecto en el explorador de paquetes y pulse **F5. En NetBeans IDE 8.0 seleccione el comando de menú Source | Scan for External Changes).**

Una vez terminada la operación de copia, los paquetes de Altova aparecerán en el explorador de paquetes (en Eclipse) o bajo *Source Packages* en el panel *Proyects* (en NetBeans IDE).



Paquetes de Altova en Eclipse 4.4



Paquetes de Altova en NetBeans IDE 8.0.2

27.2.5 Ejemplo: biblioteca

Este ejemplo explica cómo utilizar las bibliotecas contenedoras de esquemas que se generan para escribir o leer mediante programación documentos XML conformes con el esquema. Antes de usar el código de muestra,

recomendamos dedicar un tiempo a comprender la estructura del esquema de ejemplo[1156]. Este esquema es necesario para generar las bibliotecas de código que se utilizan en este ejemplo y si conoce y comprende su estructura podrá trabajar más fácilmente con el código que se genera cuando necesite adaptarlo a sus requisitos personales.

27.2.5.1 Ejemplo de esquema

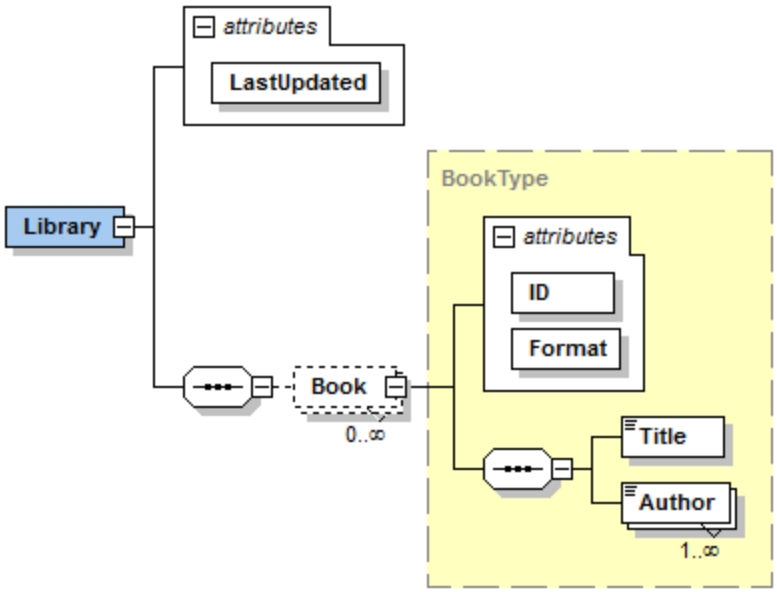
El esquema que se utiliza en este ejemplo describe una biblioteca de libros. Su definición completa aparece más abajo. Si quiere obtener los mismos resultados que se obtienen en esta sección, guarde este fragmento de código como **Doc.xsd**. Este esquema se necesita para generar las bibliotecas de código utilizadas más adelante en esta sección.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.nanonull.com/LibrarySample"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.nanonull.com/LibrarySample" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:element name="Library">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="Book" type="BookType" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attribute name="LastUpdated" type="xs:dateTime"/>
      </xs:complexType>
   </xs:element>
   <xs:complexType name="BookType">
      <xs:sequence>
         <xs:element name="Title" type="xs:string"/>
         <xs:element name="Author" type="xs:string" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:integer" use="required"/>
      <xs:attribute name="Format" type="BookFormatType" use="required"/>
   </xs:complexType>
   <xs:complexType name="DictionaryType">
      <xs:complexContent>
         <xs:extension base="BookType">
            <xs:sequence>
               <xs:element name="FromLang" type="xs:string"/>
               <xs:element name="ToLang" type="xs:string"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:simpleType name="BookFormatType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="Hardcover"/>
         <xs:enumeration value="Paperback"/>
         <xs:enumeration value="Audiobook"/>
         <xs:enumeration value="E-book"/>
      </xs:restriction>
```


© 2016-2022 Altova GmbH

```
    </xs:simpleType>
</xs:schema>
```

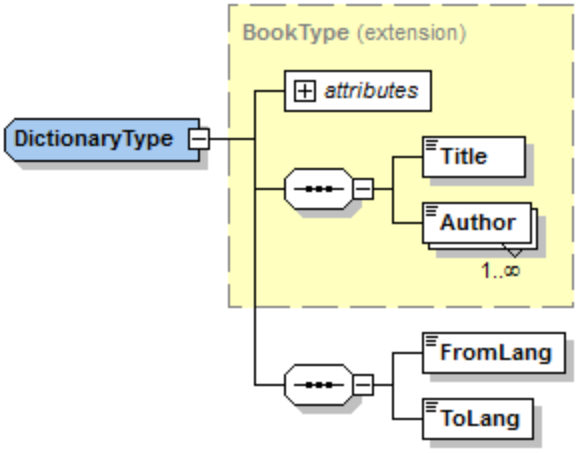
Library es un elemento raíz de un `complexType` cuya representación gráfica aparece a continuación (tomada de la vista Esquema de XMLSpy):



Como puede ver, la biblioteca tiene un atributo **LastUpdated** (definido comos `xs:dateTime`) y almacena una secuencia de libros. Cada libro es un `xs:complexType` y tiene dos atributos: **ID** (definido como `xs:integer`) y **Format**. El formato de los libros puede ser tapa dura (`hardcover`), bolsillo (`paperback`), audiolibro (`audiobook`) o libro electrónico (`e-book`). En el esquema el atributo **Format** está definido como `xs:simpleType` que usa una enumeración de los mencionados valores.

Además cada libro tiene un elemento **Title** (definido como `xs:string`) y como mínimo un elemento **Author** (definido como `xs:string`).

La biblioteca también puede incluir libros que son diccionarios. Estos tienen el tipo `DictionaryType`, que se deriva por extensión del tipo `BookType`. Es decir, un diccionario hereda todos los atributos y elementos de un libro y además toma dos elementos más: **FromLang** y **ToLang** (*imagen siguiente*).



Los elementos **FromLang** y **ToLang** almacenan el idioma de partida y de destino del diccionario.

Veamos a continuación una instancia XML válida de acuerdo con el esquema anterior (siempre y cuando esté en el mismo directorio que el esquema):

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nanonull.com/LibrarySample"
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd" LastUpdated="2016-
02-03T17:10:08.4977404">
   <Book ID="1" Format="E-book">
      <Title>The XMLSpy Handbook</Title>
      <Author>Altova</Author>
   </Book>
   <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="n1:DictionaryType">
      <Title>English-German Dictionary</Title>
      <Author>John Doe</Author>
      <FromLang>English</FromLang>
      <ToLang>German</ToLang>
   </Book>
</Library>
```

En el apartado siguiente explicamos cómo leer o escribir en dicho archivo mediante programación. Para empezar es necesario generar el código contenedor del esquema (siguiendo las instrucciones del apartado Generar código a partir de esquemas XML o documentos DTD 1142).

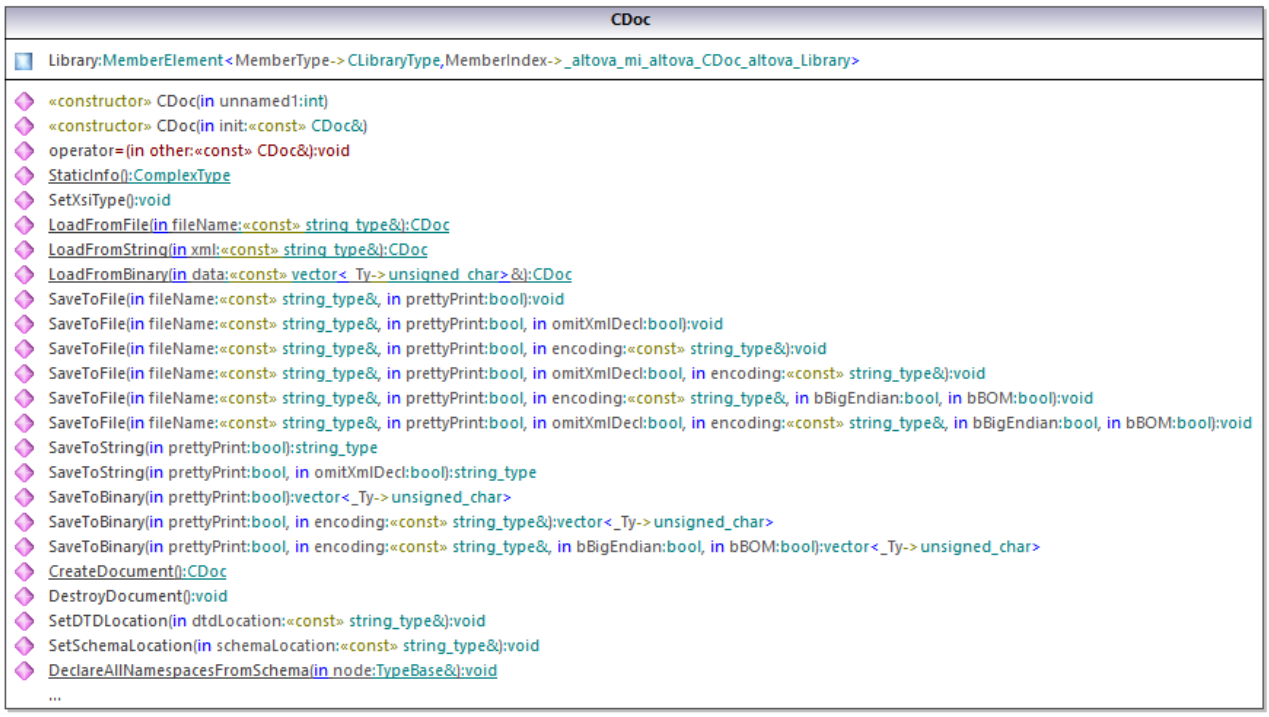
27.2.5.2 Leer y escribir documentos XML (C++)

Tras generarse el código a partir del esquema de ejemplo 1156 se crea una aplicación de prueba C++ junto con varias bibliotecas Altova secundarias.

 © 2016-2022 Altova GmbH

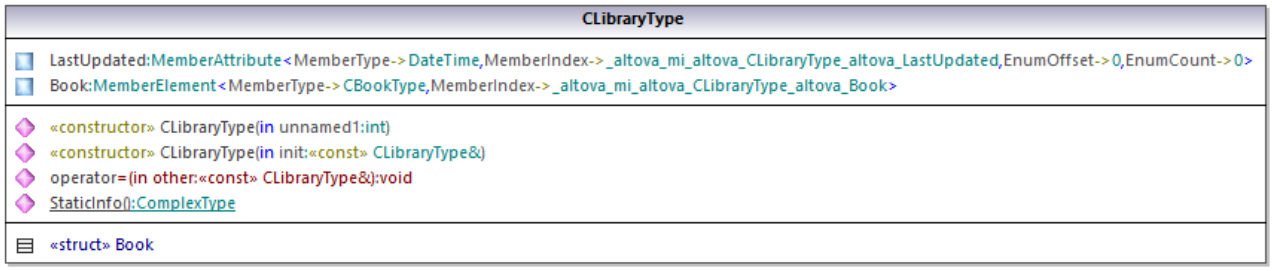
Información sobre las bibliotecas C++ generadas

La clase central del código generado es la clase `CDoc`, que representa el documento XML. Dicha clase se genera por cada esquema y su nombre depende del nombre del archivo de esquema (en nuestro ejemplo es **Doc.xsd**). Como puede verse en el diagrama, esta clase aporta métodos para cargar documentos desde archivos, secuencias binarias o cadenas de texto (o para guardar documentos en archivos, secuencias y cadenas). Para ver una descripción de los miembros que expone esta clase consulte la referencia de la clase (`[SuEsquema]::[CDoc]`[1198]).

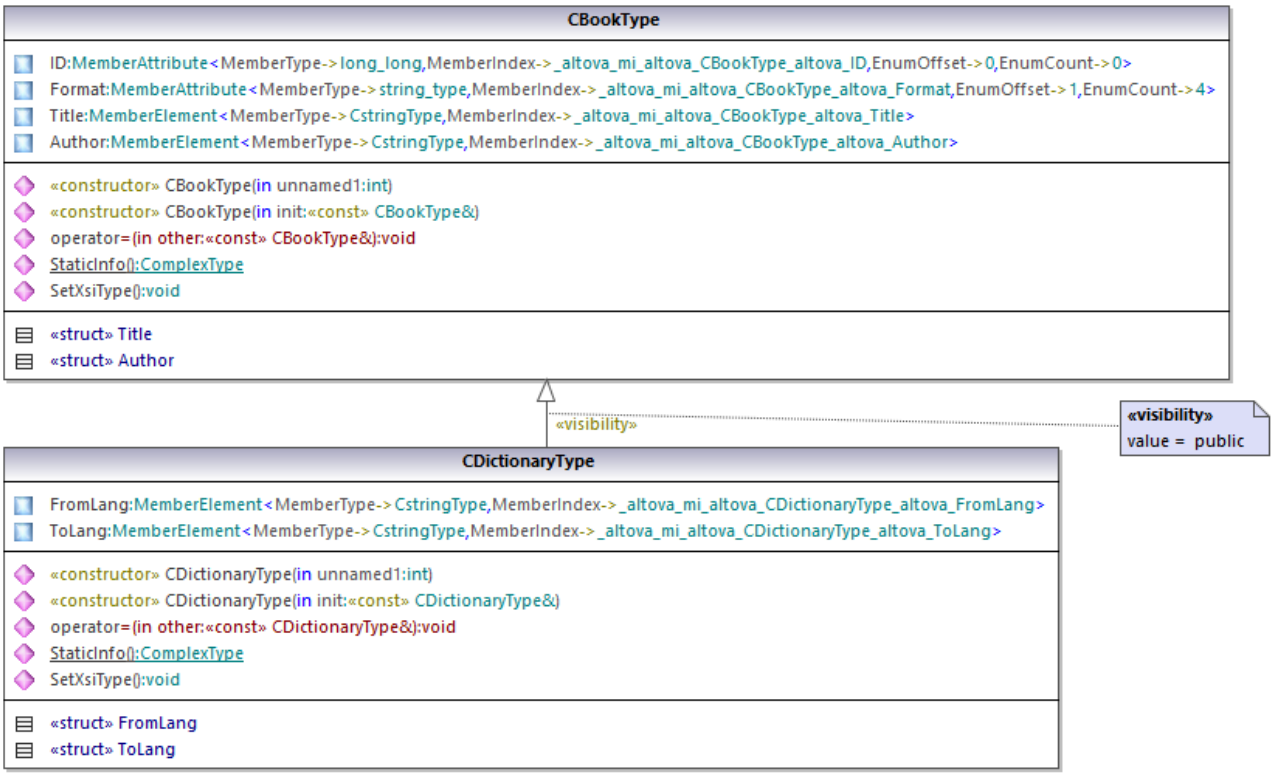


El campo `Library` de la clase `CDoc` representa la raíz real del documento. **Library** es un elemento del archivo XML, de modo que en el código C++ tiene una clase plantilla como tipo (`MemberElement`). La clase plantilla expone métodos y propiedades para interactuar con el elemento **Library**. En general, cada atributo y cada elemento de un tipo del esquema recibe un tipo en el código generado con las clases plantilla `MemberAttribute` y `MemberElement` respectivamente. Para más información consulte la referencia de las clases `[SuEsquema]::AtributoMiembro`[1202] y `[SuEsquema]::ElementoMiembro`[1202].

La clase `CLibraryType` se genera a partir del tipo complejo **LibraryType** del esquema. Observe que la clase `CLibraryType` contiene los campos `Book` y `LastUpdated`. Según la lógica mencionada previamente, estos campos corresponden al elemento **Book** y al atributo **LastUpdated** del esquema y permiten manipular elementos y atributos del documento XML de instancia mediante programación (operaciones anexar, eliminar, etc.).



Como **DictionaryType** es un tipo complejo derivado de **BookType** esta relación también se refleja en las clases generadas. Como puede verse en el diagrama, la clase `CDictionaryType` hereda la clase `CBookType`.



Si su esquema XML define tipos simples como enumeraciones, los valores enumerados están disponibles como valores `enum` en el código generado. En el esquema utilizado en este ejemplo, el formato de los libros puede ser tapa dura, bolsillo, libro electrónico y audiolibro. Por tanto, en el código generado estos valores estarán disponibles a través de un `enum` que es miembro de la clase `CBookFormatType`.

Escribir un documento XML

1. En Visual Studio abra la solución **LibraryTest.sln** que se generó a partir del esquema **Library** mencionado anteriormente.

Cuando cree prototipos de aplicaciones a partir de esquemas XML que cambien con frecuencia, a veces

 © 2016-2022 Altova GmbH

será necesario generar código una y otra vez en el mismo directorio para que los cambios en el esquema se reflejen inmediatamente en el código. Recuerde que la aplicación de prueba que se genera y las bibliotecas de Altova se sobrescribirán cada vez que genere código en el mismo directorio de destino. Por tanto, recuerde que no debe añadir código a la aplicación de prueba que se genera, sino que debe integrar las bibliotecas de Altova en el proyecto (ver *Integrar bibliotecas contenedoras de esquemas* 1152).

2. En el explorador de soluciones abra el archivo **LibraryTest.cpp** y edite el método `Example()` como se indica a continuación:

```
#include <ctime> // obligatorio para obtener hora actual
using namespace Library; // obligatorio para trabajar con bibliotecas de Altova

void Example()
{
   // Crear un documento XML nuevo vacío
   CDoc libDoc = CDoc::CreateDocument();

   // Crear el elemento raíz <Library> y agregarlo al documento
   CLibraryType lib = libDoc.Library.append();

   // Obtener la hora actual y establecer el atributo "LastUpdated" usando clases de
Altova
   time_t t = time(NULL);
   struct tm * now = localtime( & t );
   altova::DateTime dt = altova::DateTime(now->tm_year + 1900, now->tm_mon + 1, now-
>tm_mday, now->tm_hour, now->tm_min, now->tm_sec);
   lib.LastUpdated = dt;

   // Crear un nuevo <Book> y agregarlo a la biblioteca
   CBookType book = lib.Book.append();

   // Establecer el atributo "ID" del libro
   book.ID = 1;

   // Establecer el atributo "Format" del <Book> usando una constante de enumeración
   book.Format.SetEnumerationValue( CBookFormatType::k_Paperback );

   // Agregar los elementos <Title> y <Author> y establecer valores
   book.Title.append() = _T("The XML Spy Handbook");
   book.Author.append() = _T("Altova");

   // Anexar un diccionario (libro de tipo derivado) y rellenar sus atributos y elementos
   CDictionaryType dictionary = CDictionaryType(lib.Book.append().GetNode());
   dictionary.ID = 2;
   dictionary.Format.SetEnumerationValue( CBookFormatType::k_E_book);
   dictionary.Title.append() = _T("English-German Dictionary");
   dictionary.Author.append() = _T("John Doe");
   dictionary.FromLang.append() = _T("English");
   dictionary.ToLang.append() = _T("German");

   // Como diccionario es un tipo derivado, establecer el atributo xsi:type del elemento
book
```

```
   dictionary.SetXsiType();

   // De forma opcional establecer la ubicación del esquema
   libDoc.SetSchemaLocation(_T("Library.xsd"));

   // Guardar el documento XML en un archivo con la codificación predeterminada(UTF-8)
   // "true" causes the file to be pretty-printed.
   libDoc.SaveToFile(_T("GeneratedLibrary.xml"), true);

   // Destruir el documento
   libDoc.DestroyDocument();
}
```

3. Pulse **F5** para iniciar la depuración. Si el código se ejecuta correctamente, el archivo **GeneratedLibrary.xml** se crea en el directorio de salida de la solución.

Leer un documento XML

1. Abra la solución **LibraryTest.sln** en Visual Studio.
2. Guarde el código que aparece a continuación como **Library1.xml** en un directorio que el código de programa pueda leer (p. ej. en el mismo directorio que **LibraryTest.sln**).

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nanonull.com/LibrarySample"
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd" LastUpdated="2016-
02-03T17:10:08.4977404">
   <Book ID="1" Format="E-book">
      <Title>The XMLSpy Handbook</Title>
      <Author>Altova</Author>
   </Book>
   <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="n1:DictionaryType">
      <Title>English-German Dictionary</Title>
      <Author>John Doe</Author>
      <FromLang>English</FromLang>
      <ToLang>German</ToLang>
   </Book>
</Library>
```

3. En el explorador de soluciones abra el archivo **LibraryTest.cpp** y edite el método `Example()` como se indica a continuación.

```
using namespace Doc;
void Example()
{
   // Cargar documento XML
   CDoc libDoc = CDoc::LoadFromFile(_T("Library1.xml"));

   // Obtener el primer y único elemento raíz <Library>
   CLibraryType lib = libDoc.Library.first();
```


© 2016-2022 Altova GmbH

```
   // Compruebe si existe un elemento:
   if (!lib.Book.exists())
   {
      tcout << "This library is empty." << std::endl;
      return;
   }

   // iteración: por cada <Book>...
   for (Iterator<CBookType> itBook = lib.Book.all(); itBook; ++itBook)
   {
      // valores de salida del atributo ISBN y del primer y único elemento title
      tcout << "ID: " << itBook->ID << std::endl;
      tcout << "Title: " << tstring(itBook->Title.first()) << std::endl;

      // leer y compara un valor de enumeración
      if (itBook->Format.GetEnumerationValue() == CBookFormatType::k_Paperback)
         tcout << "This is a paperback book." << std::endl;

      // por cada <Author>...
      for (CBookType::Author::iterator itAuthor = itBook->Author.all(); itAuthor; +
+itAuthor)
         tcout << "Author: " << tstring(itAuthor) << std::endl;

      // alternativa: usar recuento e índice
      for (unsigned int j = 0; j < itBook->Author.count(); ++j)
         tcout << "Author: " << tstring(itBook->Author[j]) << std::endl;
   }

   // Destruir el documento
   libDoc.DestroyDocument();
}
```

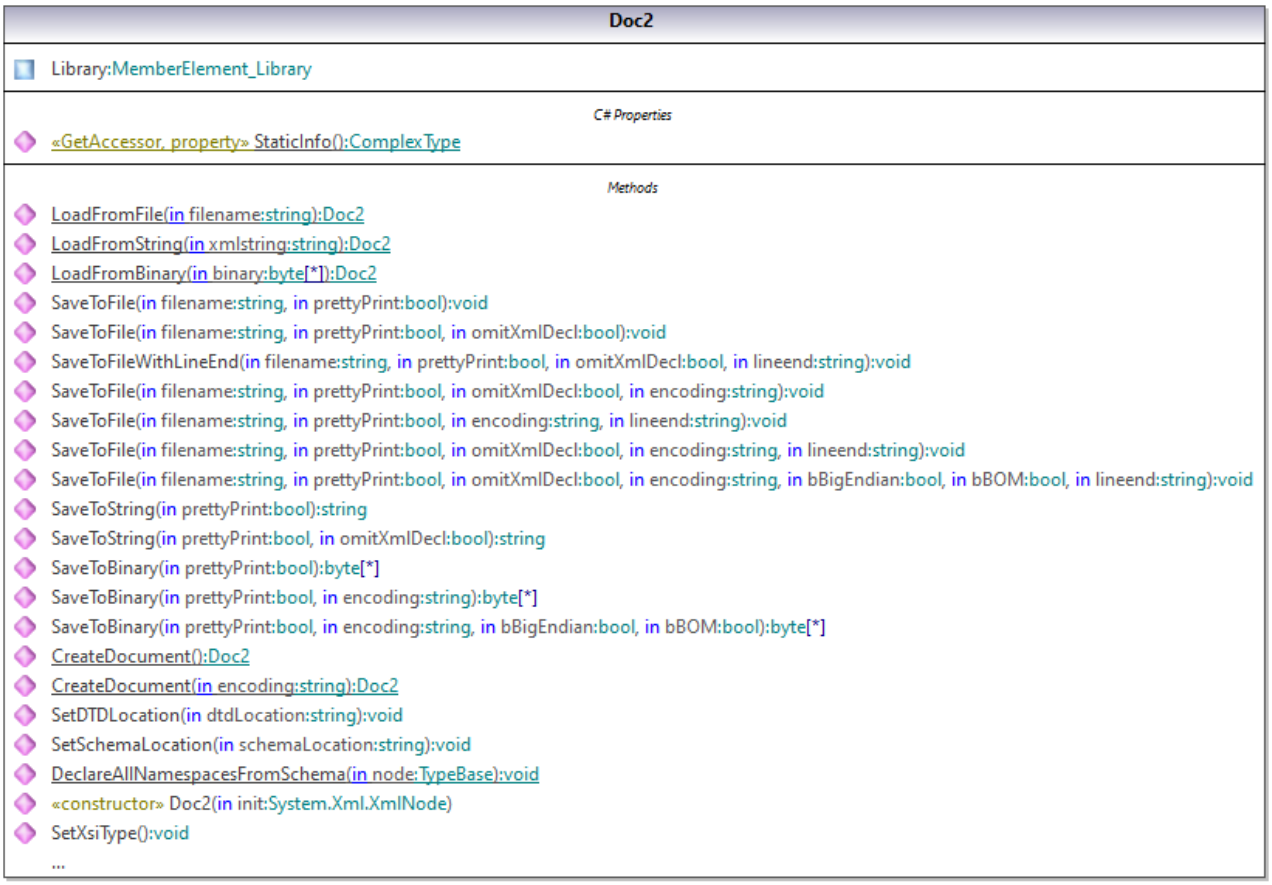
4. Pulse **F5** para iniciar la depuración.

27.2.5.3 Leer y escribir documentos XML (C#)

Tras generarse el código a partir del esquema de ejemplo[1156] se crea una aplicación de prueba C# junto con varias bibliotecas Altova secundarias.

Información sobre las bibliotecas C# generadas

La clase central del código generado es la clase `Doc2`, que representa el documento XML. Dicha clase se genera por cada esquema y su nombre depende del nombre del archivo de esquema (en nuestro ejemplo es **Doc.xsd**). Observe que esta clase se llama `Doc2` para evitar conflictos entre los nombres de espacio de nombres. Como puede verse en el diagrama, esta clase aporta métodos para cargar documentos desde archivos, secuencias binarias o cadenas de texto (o para guardar documentos en archivos, secuencias y cadenas). Para ver una descripción de esta clase consulte la referencia de la clase (`[SuEsquema].[Doc]`[1214]).

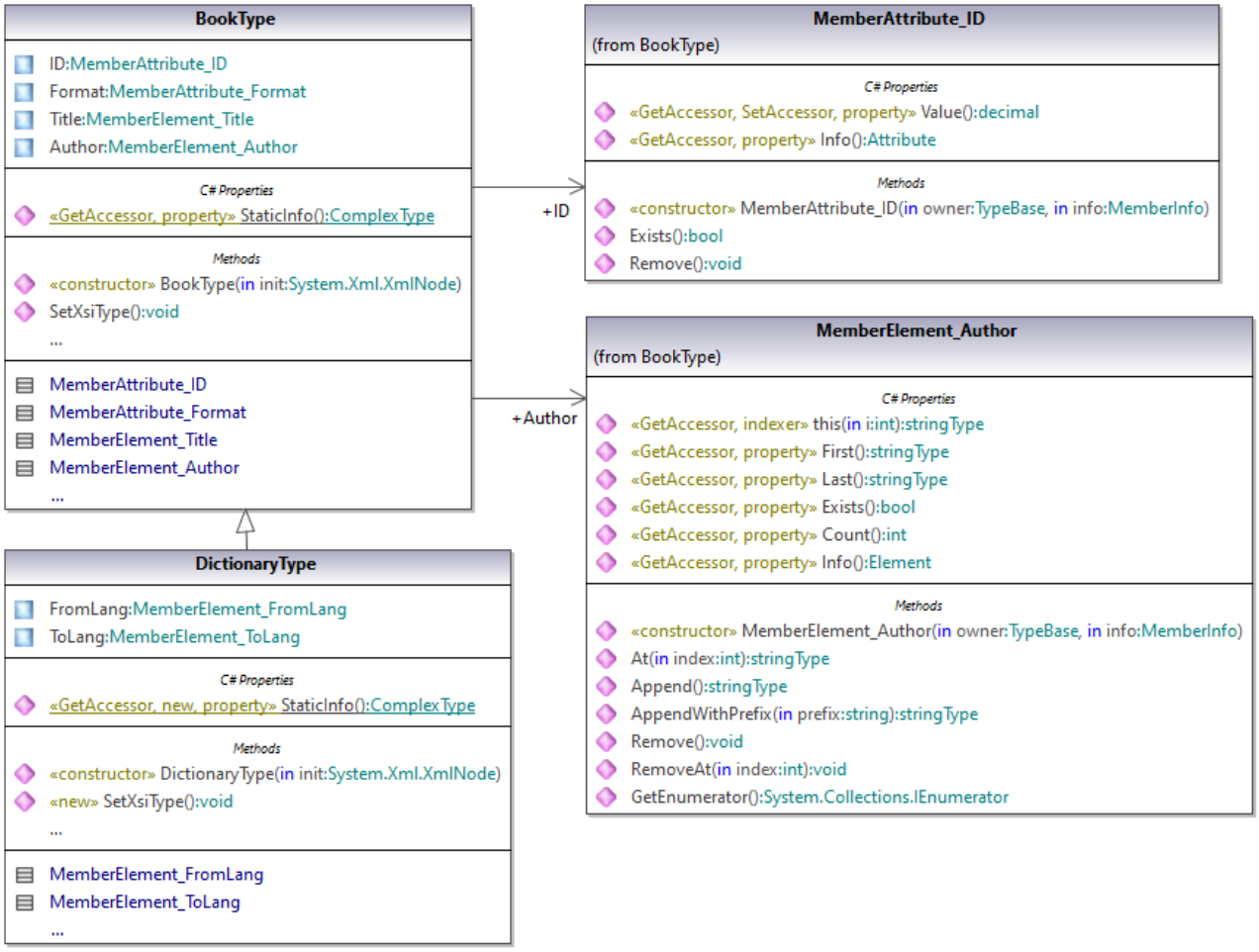


El miembro `Library` de la clase `Doc2` representa la raíz real del documento.

De acuerdo con las reglas de generación de código mencionadas en el apartado Bibliotecas contenedoras de esquemas (C#)[1148], se generan clases miembro por cada atributo y por cada elemento de un tipo. En el código generado el nombre de dichas clases miembros va precedido por el prefijo `MemberAttribute_` y `MemberElement_` respectivamente. Son ejemplos de dichas clases: `MemberAttribute_ID` y `MemberElement_Author`, que se generan a partir del elemento **Author** y del atributo **ID** de un libro respectivamente (en el siguiente diagrama están anidadas bajo `BookType`). Dichas clases permiten manipular los correspondientes elementos y atributos del documento XML de instancia mediante programación (operaciones anexar, eliminar, establecer valor, etc.). Para más información consulte la referencia de las clases `[SuTipoEsquema].MemberAttribute`[1217] y `[SuTipoEsquema].MemberElement`[1218].

Como en el esquema **DictionaryType** es un tipo complejo derivado de **BookType**, esta relación también se refleja en las clases generadas. Como puede verse en el diagrama siguiente, la clase `DictionaryType` hereda la clase `BookType`.

 © 2016-2022 Altova GmbH



Si su esquema XML define tipos simples como enumeraciones, los valores enumerados están disponibles como valores `Enum` en el código generado. En el esquema utilizado en este ejemplo, el formato de los libros puede ser tapa dura, bolsillo, libro electrónico y audiolibro. Por tanto, en el código generado estos valores estarán disponibles a través de un `Enum` que es miembro de la clase `BookFormatType`.

Escribir un documento XML

1. En Visual Studio abra la solución **LibraryTest.sln** que se generó a partir del esquema Library mencionado anteriormente.

Cuando cree prototipos de aplicaciones a partir de esquemas XML que cambien con frecuencia, a veces será necesario generar código una y otra vez en el mismo directorio para que los cambios en el esquema se reflejen inmediatamente en el código. Recuerde que la aplicación de prueba que se genera y las bibliotecas de Altova se sobrescribirán cada vez que genere código en el mismo directorio de destino. Por tanto, recuerde que no debe añadir código a la aplicación de prueba que se genera, sino que debe integrar las bibliotecas de Altova en el proyecto (ver *Integrar bibliotecas contenedoras de esquemas* 1152).

2. En el explorador de soluciones abra el archivo **LibraryTest.cs** y edite el método `Example()` como se indica a continuación:

```
        protected static void Example()
        {
            // Crear una biblioteca XML nueva
            Library2 doc = Library2.CreateDocument();
            // Anexar el elemento raíz
            LibraryType root = doc.Library3.Append();

            // Crear la fecha de generación de biblioteca con ayuda de la clase DateTime
de Altova
            Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);
            // Anexar la fecha a la raíz
            root.LastUpdated.Value = dt;

            // Agregar un libro nuevo
            BookType book = root.Book.Append();
            // Establecer el valor del atributo ID
            book.ID.Value = 1;
            // Establecer el formato del libro (enumeración)
            book.Format.EnumerationValue = BookFormatType.EnumValues.eHardcover;
            // Establecer los elementos Title y Author
            book.Title.Append().Value = "The XMLSpy Handbook";
            book.Author.Append().Value = "Altova";

            // Anexar un diccionario (libro de tipo derivado) y rellenar sus atributos y
elementos
            DictionaryType dictionary = new DictionaryType(root.Book.Append().Node);
            dictionary.ID.Value = 2;
            dictionary.Title.Append().Value = "English-German Dictionary";
            dictionary.Format.EnumerationValue = BookFormatType.EnumValues.eE_book;
            dictionary.Author.Append().Value = "John Doe";
            dictionary.FromLang.Append().Value = "English";
            dictionary.ToLang.Append().Value = "German";
            // Como es un tipo derivado, establecer el atributo xsi:type del elemento
book
            dictionary.SetXsiType();

            // De forma opcional, establecer la ubicación del esquema (ajustar la ruta de
acceso
            // si el esquema no está en la misma carpeta que el archivo de instancia que
se generó)
            doc.SetSchemaLocation("Library.xsd");

            // Guardar el documento XML con la opción pretty print habilitada
            doc.SaveToFile("GeneratedLibrary.xml", true);
        }
```

3. Pulse **F5** para iniciar la depuración. Si el código se ejecuta correctamente, el archivo **GeneratedLibrary.xml** se crea en el directorio de salida de la solución (que suele ser **bin/Debug**).

© 2016-2022 Altova GmbH

Leer un documento XML

1. Abra la solución **LibraryTest.sln** en Visual Studio.
2. Guarde el código que aparece a continuación en un archivo llamado **Library.xml** en el directorio de salida del proyecto (predeterminado: **bin/Debug**). Este es el archivo que leerá el código de programa.

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nanonull.com/LibrarySample"
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd" LastUpdated="2016-
02-03T17:10:08.4977404">
   <Book ID="1" Format="E-book">
      <Title>The XMLSpy Handbook</Title>
      <Author>Altova</Author>
   </Book>
   <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="n1:DictionaryType">
      <Title>English-German Dictionary</Title>
      <Author>John Doe</Author>
      <FromLang>English</FromLang>
      <ToLang>German</ToLang>
   </Book>
</Library>
```

3. En el explorador de soluciones abra el archivo **LibraryTest.cs** y edite el método `Example()` como se indica a continuación:

```
protected static void Example()
{
    // Cargar el archivo XML en una instancia nueva de Library
    Doc2 doc = Doc2.LoadFromFile("Library.xml");
    // Obtener el elemento raíz
    LibraryType root = doc.Library.First;

    // Leer la fecha de generación de Library
    Altova.Types.DateTime dt = root.LastUpdated.Value;
    string dt_as_string = dt.ToString(DateTimeFormat.W3_dateTime);
    Console.WriteLine("The library generation date is: " + dt_as_string);

    // Iteración: por cada <Book>...
    foreach (BookType book in root.Book)
    {
      // Valores de salida del atributo ID y del (primer y único) elemento title
      Console.WriteLine("ID:    " + book.ID.Value);
      Console.WriteLine("Title: " + book.Title.First.Value);

      // Leer y comparar un valor de enumeración
      if (book.Format.EnumerationValue == BookFormatType.EnumValues.ePaperback)
         Console.WriteLine("This is a paperback book.");
```

```
      // Iteración: por cada <Author>
      foreach (xs.stringType author in book.Author)
         Console.WriteLine("Author: " + author.Value);

      // Determinar si se trata de un libro de tipo derivado
      if (book.Node.Attributes.GetNamedItem("xsi:type") != null)
      {
         // Buscar el valor del atributo xsi:type
         string xsiTypeValue = book.Node.Attributes.GetNamedItem("xsi:type").Value;
         // Obtener el URI de espacio de nombres
        // y el prefijo de consulta de este espacio de nombres
         string namespaceUri = book.Node.NamespaceURI;
         string prefix = book.Node.GetPrefixOfNamespace(namespaceUri);

         // si este libro tiene DictionaryType
         if (namespaceUri == "http://www.nanonull.com/LibrarySample" &&
xsiTypeValue.Equals(prefix + ":DictionaryType"))
         {
            // generar campos adicionales
            DictionaryType dictionary = new DictionaryType(book.Node);
            Console.WriteLine("Language from: " + dictionary.FromLang.First.Value);
            Console.WriteLine("Language to: " + dictionary.ToLang.First.Value);
         }
         else
         {
            throw new Exception("Unexpected book type");
         }
      }
    }

    Console.ReadLine();
}
```

4. Pulse **F5** para iniciar la depuración. Si el código se ejecuta correctamente, el código de programa leerá el archivo **GeneratedLibrary.xml** y su contenido aparecerá en el resultado de la consola.

Leer y escribir elementos y atributos

El acceso a los atributos y elementos se consigue con la propiedad `Value` de la clase de elemento o atributo miembro que se genera. Por ejemplo:

```
// Valores de salida del atributo ID y del (primer y único) elemento title
Console.WriteLine("ID:    " + book.ID.Value);
Console.WriteLine("Title: " + book.Title.First.Value);
```

Para obtener el valor del elemento **Title** en este ejemplo concreto también se utilizó el método `First()`. Esto se debe a que este es el primer (y único) elemento **Title** de un libro. Cuando necesite seleccionar un elemento concreto de una lista por medio del índice, utilice el método `At()`.

© 2016-2022 Altova GmbH

La clase que se genera por cada elemento miembro de un tipo implementa la interfaz estándar `System.Collections.IEnumerable`. Esto permite recorrer varios elementos del mismo tipo. En este ejemplo concreto puede recorrer todos los libros de un objeto Library como se puede ver a continuación:

```
// Iteración: por cada <Book>...
foreach (BookType book in root.Book)
{
    // your code here...
}
```

Para agregar un elemento nuevo utilice el método `Append()`. Por ejemplo, a continuación se anexa el elemento raíz al documento:

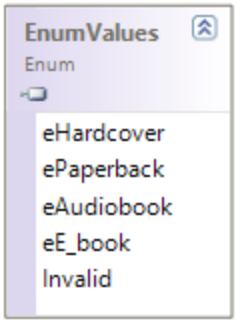
```
// Anexar el elemento raíz a la biblioteca
LibraryType root = doc.Library.Append();
```

Puede establecer el valor de un atributo (como ID) de la siguiente manera:

```
// Establecer el valor del atributo ID
book.ID.Value = 1;
```

Leer y escribir valores de enumeración

Si el esquema XML define tipos simples como enumeraciones, los valores enumerados estarán disponibles como valores `Enum` en el código generado. En el esquema utilizado en este ejemplo el formato de un libro puede ser tapa dura, bolsillo, libro electrónico y audiolibro. Por tanto, en el código generado estos valores estarán disponibles a través de una `Enum`:



Para asignar valores de enumeración a un objeto utilice código como este:

```
// Establecer el formato del libro (enumeración)
book.Format.EnumerationValue = BookFormatType.EnumValues.eHardcover
```

Puede leer dichos valores de enumeración desde documento XML de instancia de la siguiente manera:

```
// Leer y comparar un valor de enumeración
if (book.Format.EnumerationValue == BookFormatType.EnumValues.ePaperback)
Console.WriteLine("This is a paperback book.");
```

Cuando la condición IF no sea suficiente, cree un modificador para determinar cada valor de enumeración y procesarlos según corresponda en cada caso.

Trabajar con tipos xs:dateTime y xs:duration

Si el esquema desde el que se genera código utiliza tipos de hora y duración como `xs:dateTime` o `xs:duration`, estos tipos se convierten en clases nativas de Altova en el código generado. Por tanto, para escribir un valor de fecha o duración en el documento XML debe seguir estas instrucciones:

1. Construya un objeto `Altova.Types.DateTime`[1204] o `Altova.Types.Duration`[1209] (bien desde `System.DateTime` o utilizando partes como horas y minutos como en `Altova.Types.DateTime`[1204] y `Altova.Types.Duration`[1209]).
2. Establezca el objeto como valor del elemento o atributo que necesita. Por ejemplo:

```
// Crear la fecha de generación de la biblioteca usando la clase DateTime de Altova
Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);
// Anexar la fecha a la raíz
root.LastUpdated.Value = dt;
```

Para leer una fecha o duración de un documento XML:

1. Declare el valor de elemento (o atributo) como objeto `Altova.Types.DateTime`[1204] o `Altova.Types.Duration`[1209].
2. Aplique formato al elemento o atributo que necesita. Por ejemplo:

```
// Leer la fecha de generación de la biblioteca
Altova.Types.DateTime dt = root.LastUpdated.Value;
string dt_as_string = dt.ToString(DateTimeFormat.W3_dateTime);
Console.WriteLine("The library generation date is:
```

Para más información consulte la referencia de las clases `Altova.Types.DateTime`[1204] y `Altova.Types.Duration`[1209].

Trabajar con tipos derivados

Si su esquema XML define tipos derivados puede conservar la derivación de tipos en los documentos XML que cree o cargue mediante programación. Tomando el esquema utilizado en este ejemplo, el fragmento de código que aparece a continuación explica cómo se crea un libro nuevo de tipo derivado `DictionaryType`:

```
// Anexar un diccionario (libro de tipo derivado) y rellenar sus atributos y elementos
DictionaryType dictionary = new DictionaryType(root.Book.Append().Node);
dictionary.ID.Value = 2;
dictionary.Title.Append().Value = "English-German Dictionary";
dictionary.Author.Append().Value = "John Doe";
dictionary.FromLanguage.Append().Value = "English";
dictionary.ToLanguage.Append().Value = "German";
```


© 2016-2022 Altova GmbH

```
// Como es un tipo derivado, asegúrese de establecer el atributo xsi:type del elemento
book
dictionary.SetXsiType();
```

Recuerde que es importante establecer el atributo `xsi:type` del libro recién creado. Esto garantiza que el esquema interprete correctamente el tipo de libro a la hora de validar el documento XML.

El fragmento de código que aparece a continuación identifica un libro de tipo derivado `DictionaryType` en la instancia XML que se carga. Primero se busca el valor del atributo `xsi:type` del nodo book. Si el URI de espacio de nombres de este nodo es `http://www.nanonull.com/LibrarySample` y el prefijo de consulta URI y el tipo coinciden con el valor del atributo `xsi:type`, entonces sabemos que se trata de un diccionario.

```
      // Determinar si este libro es de tipo derivado
      if (book.Node.Attributes.GetNamedItem("xsi:type") != null)
      {
          // Buscar el valor del atributo xsi:type
          string xsiTypeValue = book.Node.Attributes.GetNamedItem("xsi:type").Value;
          // Obtener el URI de espacio de nombres y el prefijo de consulta de este
espacio de nombres
          string namespaceUri = book.Node.NamespaceURI;
          string prefix = book.Node.GetPrefixOfNamespace(namespaceUri);

          // Si este libro tiene DictionaryType
          if (namespaceUri == "http://www.nanonull.com/LibrarySample" &&
xsiTypeValue.Equals(prefix + ":DictionaryType"))
          {
             // generar campos adicionales
             DictionaryType dictionary = new DictionaryType(book.Node);
             Console.WriteLine("Language from: " + dictionary.FromLang.First.Value);
             Console.WriteLine("Language to: " + dictionary.ToLang.First.Value);
          }
          else
          {
              throw new Exception("Unexpected book type");
          }
      }
```

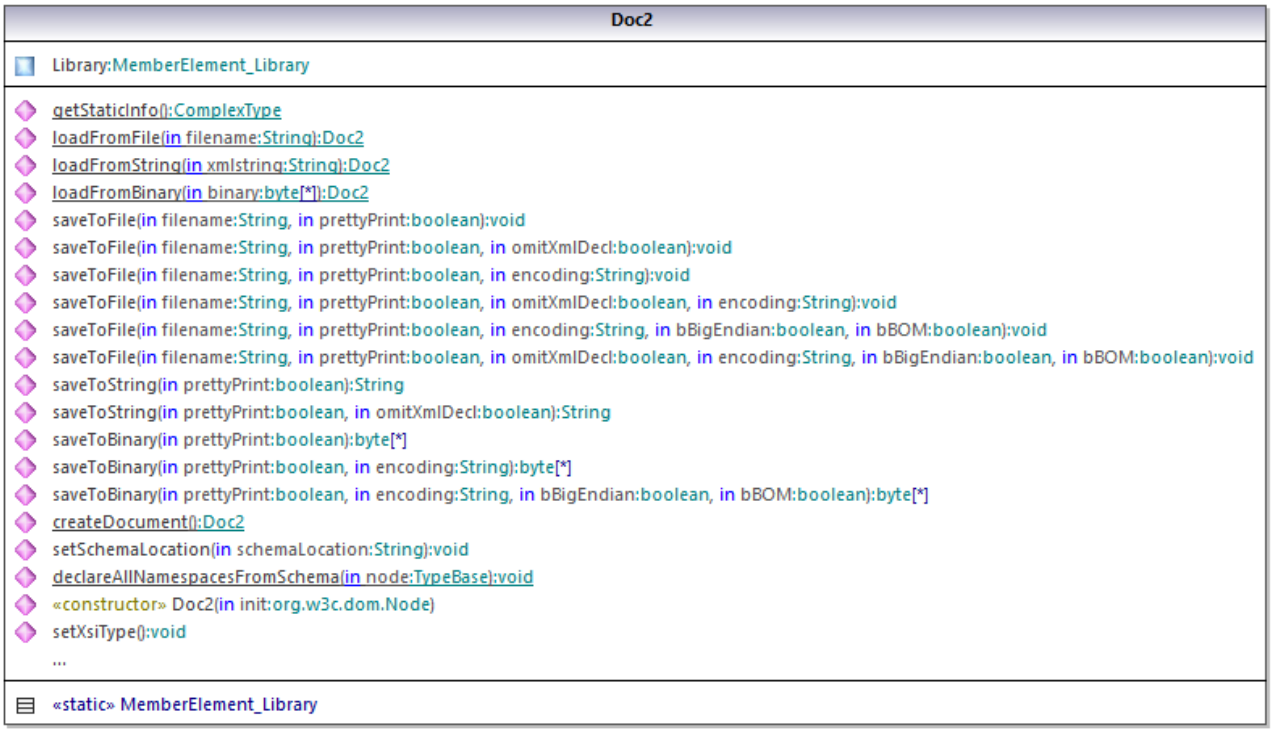
27.2.5.4 Leer y escribir documentos XML (Java)

Tras generarse el código a partir del esquema de ejemplo[1156] se crea un proyecto de prueba Java junto con varias bibliotecas Altova secundarias.

Información sobre las bibliotecas Java generadas

La clase central del código generado es la clase Doc2, que representa el documento XML. Dicha clase se genera por cada esquema y su nombre depende del nombre del archivo de esquema (en nuestro ejemplo es **Doc.xsd**). Observe que esta clase se llama `Doc2` para evitar conflictos entre los nombres de espacio de nombres. Como puede verse en el diagrama, esta clase aporta métodos para cargar documentos desde archivos, secuencias binarias o cadenas de texto (o para guardar documentos en archivos, secuencias y

cadenas). Para ver una descripción de esta clase consulte la referencia de la clase `com.[YourSchema].[Doc]`[1232].

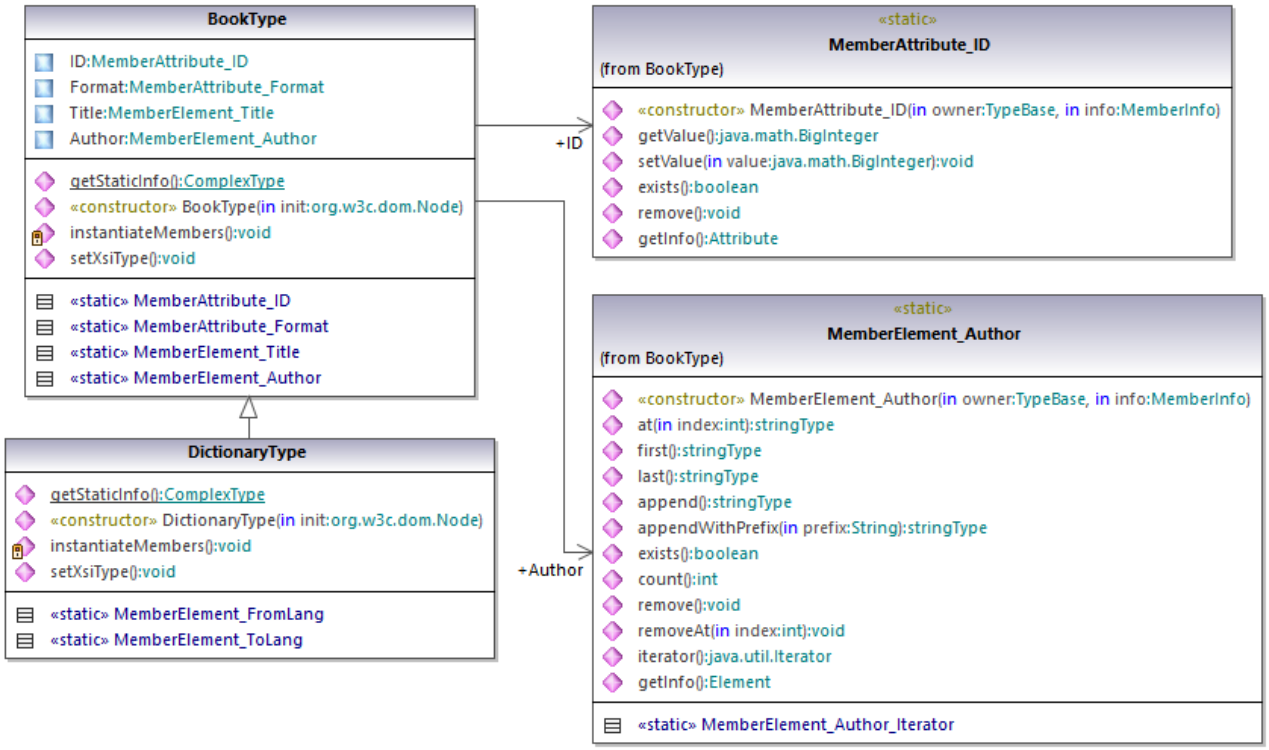


El miembro `Library` de la clase `Doc2` representa la raíz real del documento.

De acuerdo con las reglas de generación de código mencionadas en el apartado Bibliotecas contenedoras de esquemas (Java)[1150], se generan clases miembro por cada atributo y por cada elemento de un tipo. En el código generado el nombre de dichas clases miembros va precedido por el prefijo `MemberAttribute_` y `MemberElement_` respectivamente. En el diagrama anterior puede ver ejemplos de dichas clases: `MemberAttribute_ID` y `MemberElement_Author`, que se generan a partir del elemento **Author** y del atributo **ID** de un libro respectivamente. Dichas clases permiten manipular los correspondientes elementos y atributos del documento XML de instancia mediante programación (operaciones anexar, eliminar, establecer valor, etc.). Para más información consulte la referencia de las clases `com.[SuEsquema].[SuTipoEsquema].MemberAttribute`[1235] y `com.[SuEsquema].[SuTipoEsquema].MemberElement`[1236].

Como en el esquema **DictionaryType** es un tipo complejo derivado de **BookType**, esta relación también se refleja en las clases generadas. Como puede verse en el diagrama, la clase `DictionaryType` hereda la clase `BookType`.

 © 2016-2022 Altova GmbH



Si su esquema XML define tipos simples como enumeraciones, los valores enumerados están disponibles como valores `Enum` en el código generado. En el esquema utilizado en este ejemplo, el formato de los libros puede ser tapa dura, bolsillo, libro electrónico y audiolibro. Por tanto, en el código generado estos valores estarán disponibles a través de un `Enum` que es miembro de la clase `BookFormatType`.

Escribir un documento XML

1. En el menú **File** de Eclipse haga clic en el comando **Import** y seleccione **Existing Projects into Workspace** (Proyectos en el espacio de trabajo). Después haga clic en **Next** para continuar.
2. Junto a **Select root directory** (Seleccionar directorio raíz) haga clic en el botón **Browse**. Después seleccione el directorio donde generó el código Java y haga clic en **Finish** para finalizar.
3. En el explorador de paquetes de Eclipse expanda el paquete **com.LibraryTest** y abra el archivo **LibraryTest.java**.

Cuando cree prototipos de aplicaciones a partir de esquemas XML que cambien con frecuencia, a veces será necesario generar código una y otra vez en el mismo directorio para que los cambios en el esquema se reflejen inmediatamente en el código. Recuerde que la aplicación de prueba que se genera y las bibliotecas de Altova se sobrescribirán cada vez que genere código en el mismo directorio de destino. Por tanto, recuerde que no debe añadir código a la aplicación de prueba que se genera, sino que debe integrar las bibliotecas de Altova en el proyecto (ver *Integrar bibliotecas contenedoras de esquemas* 1152).

4. Edite el método `Example()` como se indica a continuación:

```
protected static void example() throws Exception {
      // crear un documento XML nuevo vacío
      Doc2 libDoc = Doc2.createDocument();

      // crear el elemento raíz <Library> y agregarlo al documento
      LibraryType lib = libDoc.Library.append();

      // establecer el atributo "LastUpdated"
      com.altova.types.DateTime dt = new com.altova.types.DateTime(DateTime.now());
      lib.LastUpdated.setValue(dt);

      // crear un nuevo <Book> y rellenar sus elementos y atributos
      BookType book = lib.Book.append();
      book.ID.setValue(java.math.BigInteger.valueOf(1));
      book.Format.setEnumerationValue( BookFormatType.EPAPERBACK );
      book.Title.append().setValue("The XML Spy Handbook");
      book.Author.append().setValue("Altova");

      // crear un diccionario (libro de tipo derivado) y rellenar sus elementos y
atributos
      DictionaryType dict = new DictionaryType(lib.Book.append().getNode());
      dict.ID.setValue(java.math.BigInteger.valueOf(2));
      dict.Title.append().setValue("English-German Dictionary");
      dict.Format.setEnumerationValue(BookFormatType.EE_BOOK);
      dict.Author.append().setValue("John Doe");
      dict.FromLang.append().setValue("English");
      dict.ToLang.append().setValue("German");
      dict.setXsiType();

      // establecer la ubicación del esquema (opcional)
      libDoc.setSchemaLocation("Library.xsd");

      // guardar el documento XML en un archivo con la codificación predeterminada (UTF-
8). el valor "true" aplica formato pretty-print al archivo.
      libDoc.saveToFile("Library1.xml", true);
   }
```

5. Compile y ejecute el proyecto Java. Si el código se ejecuta correctamente, se crea el archivo **Library1.xml** en el directorio del proyecto.

Leer un documento XML

1. En el menú **File** de Eclipse haga clic en el comando **Import** y seleccione **Existing Projects into Workspace** (Proyectos en el espacio de trabajo). Después haga clic en **Next** para continuar.
2. Junto a **Select root directory** (Seleccionar directorio raíz) haga clic en el botón **Browse**. Después seleccione el directorio donde generó el código Java y haga clic en **Finish** para finalizar.
3. Guarde el código que aparece a continuación en un archivo llamado **Library1.xml** en un directorio local (deberá referirse a la ruta de acceso del archivo **Library1.xml** desde el código).

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

 © 2016-2022 Altova GmbH

```
xmlns="http://www.nanonull.com/LibrarySample"
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd" LastUpdated="2016-
02-03T17:10:08.4977404">
   <Book ID="1" Format="E-book">
      <Title>The XMLSpy Handbook</Title>
      <Author>Altova</Author>
   </Book>
   <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="n1:DictionaryType">
      <Title>English-German Dictionary</Title>
      <Author>John Doe</Author>
      <FromLang>English</FromLang>
      <ToLang>German</ToLang>
   </Book>
</Library>
```

4. En el explorador de paquetes de Eclipse expanda el paquete **com.LibraryTest** y abra el archivo **LibraryTest.java**.
5. Edite el método `Example()` como se indica a continuación:

```
protected static void example() throws Exception {
     // cargar documento XML desde una ruta de acceso (ajustar la ruta si procede)
     Doc2 libDoc = Doc2.loadFromFile("Library1.xml");

     // obtener el primer (y único) elemento raíz <Library>
     LibraryType lib = libDoc.Library.first();

     // comprobar si existe un elemento:
     if (!lib.Book.exists()) {
        System.out.println("This library is empty.");
        return;
     }

     // leer el tipo DateTime del esquema
     com.altova.types.DateTime dt = lib.LastUpdated.getValue();
     System.out.println("The library was last updated on: " + dt.toDateString());

     // iteración: por cada <Book>...
     for (java.util.Iterator itBook = lib.Book.iterator(); itBook.hasNext();) {
        BookType book = (BookType) itBook.next();
        // valores de salida del atributo ID y del (primer y único) elemento title
        System.out.println("ID: " + book.ID.getValue());
        System.out.println("Title: " + book.Title.first().getValue());

        // leer y comparar un valor de enumeración
        if (book.Format.getEnumerationValue() == BookFormatType.EPAPERBACK)
           System.out.println("This is a paperback book.");

        // por cada <Author>...
        for (java.util.Iterator itAuthor = book.Author.iterator(); itAuthor
              .hasNext();)
           System.out.println("Author: " + ((com.Doc.xs.stringType)
```

```
itAuthor.next()).getValue());

        // buscar el tipo derivado de este libro
         // usando el valor del atributo xsi:type y usando DOM
        org.w3c.dom.Node bookNode = book.getNode();
        if (bookNode.getAttributes().getNamedItem("xsi:type") != null) {
           // obtener el valor del atributo xsi:type
           String xsiTypeValue =
bookNode.getAttributes().getNamedItem("xsi:type").getNodeValue();

           // obtener el URI de espacio de nombres y prefijo de consulta de este espacio
de nombres
           String namespaceUri = bookNode.getNamespaceURI();
           String lookupPrefix = bookNode.lookupPrefix(namespaceUri);

           // si xsi:type coincide con el URI de espacio de nombres y con el tipo del
nodo book
           if (namespaceUri == "http://www.nanonull.com/LibrarySample"
                 && (  xsiTypeValue.equals(lookupPrefix + ":DictionaryType" )))    {
              // ...entonces se trata de un libro de tipo derivado (diccionario)

              DictionaryType dictionary = new DictionaryType(   book.getNode());
              // generar el valor de los elementos "FromLang" y "ToLang" elements
              System.out.println("From language: " +
dictionary.FromLang.first().getValue());
              System.out.println("To language: " + dictionary.ToLang.first().getValue());
           }
           else
           {
              // emitir un error
              throw new java.lang.Error("This book has an unknown type.");
           }
        }
     }
   }
```

6. Compile y ejecute el proyecto Java. Si el código se ejecuta correctamente, el código de programa leerá el archivo **Library1.xml** y su contenido aparecerá en la vista de consola *Console*.

Leer y escribir elementos y atributos

El acceso a los atributos y elementos se consigue con el método `getValue()` de la clase de elemento o atributo miembro que se genera. Por ejemplo:

```
// valores de salida del atributo ID y del (primer y único) elemento title
System.out.println("ID: " + book.ID.getValue());
System.out.println("Title: " + book.Title.first().getValue());
```

Para obtener el valor del elemento **Title** en este ejemplo concreto también se utilizó el método `first()`. Esto se debe a que este es el primer (y único) elemento **Title** de un libro. Cuando necesite seleccionar un elemento concreto de una lista por medio del índice, utilice el método `at()`.

© 2016-2022 Altova GmbH

Para recorrer varios elementos utilice la iteración basada en índice o `java.util.Iterator`. Por ejemplo, puede recorrer los libros de la biblioteca de la siguiente manera:

```
// iteración basada en índice
for (int j = 0; j < lib.Book.count(); ++j ) {
   // your code here
}

// iteración alternativa usando java.util.Iterator
for (java.util.Iterator itBook = lib.Book.iterator(); itBook.hasNext();) {
      // introduzca aquí su código
   }
```

Para agregar un elemento nuevo utilice el método `append()`. Por ejemplo, este código anexa un elemento raíz **Library** vacío al documento:

```
// crear el elemento raíz <Library> y agregarlo al documento
LibraryType lib = libDoc.Library
```

Tras anexar el elemento, puede establecer el valor de cualquiera de sus elementos o atributos con ayuda del método `setValue()`.

```
// establecer el valor del elemento Title
book.Title.append().setValue("The XML Spy Handbook");
// establecer el valor del atributo ID
book.ID.setValue(java.math.BigInteger.valueOf(1));
```

Leer y escribir valores de enumeración

Si el esquema XML define tipos simples como enumeraciones, los valores enumerados estarán disponibles como valores `Enum` en el código generado. En el esquema utilizado en este ejemplo el formato de un libro puede ser tapa dura, bolsillo, libro electrónico y audiolibro. Por tanto, en el código generado estos valores estarán disponibles a través de una `Enum` (véase el diagrama de la clase `BookFormatType` que aparece más arriba). Para asignar valores de enumeración a un objeto utilice código parecido a este:

```
// establecer un valor de enumeración
book.Format.setEnumerationValue( BookFormatType.EPAPERBACK
```

Dichos valores de enumeración se pueden leer desde documentos XML de instancia de la siguiente manera:

```
// leer un valor de enumeración
if (book.Format.getEnumerationValue() == BookFormatType.EPAPERBACK)
        System.out.println("This is a paperback book."
```

Cuando la condición IF no sea suficiente, cree un modificador para determinar cada valor de enumeración y procesarlo según corresponda en cada caso.

Trabajar con tipos xs:dateTime y xs:duration

Si el esquema desde el que se genera código utiliza tipos de hora y duración como `xs:dateTime` o `xs:duration`, estos tipos se convierten en clases nativas de Altova en el código generado. Por tanto, para escribir un valor de fecha o duración en el documento XML debe seguir estas instrucciones:

1. Construya un objeto `com.altova.types.DateTime`[1221] o `com.altova.types.Duration`[1226].
2. Establezca el objeto como valor del elemento o atributo que necesita. Por ejemplo:

```
// establecer el valor de un atributo de tipo DateTime
com.altova.types.DateTime dt = new com.altova.types.DateTime(DateTime.now());
lib.LastUpdated.setValue(dt);
```

Para leer una fecha o duración de un documento XML:

1. Declare el valor de elemento (o atributo) como objeto `com.altova.types.DateTime`[1221] o `com.altova.types.Duration`[1226].
2. Aplique formato al elemento o atributo que necesita. Por ejemplo:

```
// leer un tipo DateTime
com.altova.types.DateTime dt = lib.LastUpdated.getValue();
   System.out.println("The library was last updated on: " + dt.toDateString());
```

Para más información consulte la referencia de las clases `com.altova.types.DateTime`[1221] y `com.altova.types.Duration`[1226].

Trabajar con tipos derivados

Si su esquema XML define tipos derivados puede conservar la derivación de tipos en los documentos XML que cree o cargue mediante programación. Tomando el esquema utilizado en este ejemplo, el fragmento de código que aparece a continuación explica cómo se crea un libro nuevo de tipo derivado `DictionaryType`:

```
// crear un diccionario (libro de tipo derivado) y rellenar sus elementos y atributos
DictionaryType dict = new DictionaryType(lib.Book.append().getNode());
dict.ID.setValue(java.math.BigInteger.valueOf(2));
dict.Title.append().setValue("English-German Dictionary");
dict.Format.setEnumerationValue(BookFormatType.EE_BOOK);
dict.Author.append().setValue("John Doe");
dict.FromLang.append().setValue("English");
dict.ToLang.append().setValue("German");
dict.setXsiType();
```

Recuerde que es importante establecer el atributo `xsi:type` del libro recién creado. Esto garantiza que el esquema interprete correctamente el tipo de libro a la hora de validar el documento XML.

El fragmento de código que aparece a continuación identifica un libro de tipo derivado `DictionaryType` en la instancia XML que se carga. Primero se busca el valor del atributo `xsi:type` del nodo book. Si el URI de

© 2016-2022 Altova GmbH

espacio de nombres de este nodo es `http://www.nanonull.com/LibrarySample` y el prefijo de consulta URI y el tipo coinciden con el valor del atributo `xsi:type`, entonces sabemos que se trata de un diccionario.

```
          // buscar el tipo derivado de este libro
         // usando el valor del atributo xsi:type y usando DOM
         org.w3c.dom.Node bookNode = book.getNode();
         if (bookNode.getAttributes().getNamedItem("xsi:type") != null) {
            // obtener el valor del atributo xsi:type
            String xsiTypeValue =
bookNode.getAttributes().getNamedItem("xsi:type").getNodeValue();

            // obtener el URI de espacio de nombres y el prefijo de consulta del nodo
book
            String namespaceUri = bookNode.getNamespaceURI();
            String lookupPrefix = bookNode.lookupPrefix(namespaceUri);

            // si xsi:type coincide con el URI de espacio de nombres y el tipo del nodo
book
            if (namespaceUri == "http://www.nanonull.com/LibrarySample"
                  && (  xsiTypeValue.equals(lookupPrefix + ":DictionaryType" )))    {
               // ...entonces se trata de un libro de tipo derivado (diccionario)

               DictionaryType dictionary = new DictionaryType(   book.getNode());
               // generar el valor de los elementos "FromLang" y "ToLang" elements
               System.out.println("From language: " +
dictionary.FromLang.first().getValue());
               System.out.println("To language: " +
dictionary.ToLang.first().getValue());
            }
            else
            {
               // emitir un error
               throw new java.lang.Error("This book has an unknown type.");
            }
         }
```

27.2.6 Ejemplo: orden de compra

Este ejemplo explica cómo trabajar con código de programa generado a partir de un esquema XML "principal" que importa otros esquemas. Cada uno de los esquemas importados tiene un espacio de nombres de destino distinto. El objetivo es crear de forma programática un documento XML en el que todos los elementos estén prefijados de acuerdo a su espacio de nombres. Más concretamente, el documento XML creado a partir de su código C++, C# o Java debería tener un aspecto parecido a este:

```
<?xml version="1.0" encoding="utf-8"?>
<p:Purchase xsi:schemaLocation="http://NamespaceTest.com/Purchase Main.xsd"
            xmlns:p="http://NamespaceTest.com/Purchase"
            xmlns:o="http://NamespaceTest.com/OrderTypes"
            xmlns:c="http://NamespaceTest.com/CustomerTypes"
            xmlns:cmn="http://NamespaceTest.com/CommonTypes"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
    <p:OrderDetail>
        <o:Item>
            <o:ProductName>Lawnmower</o:ProductName>
            <o:Quantity>1</o:Quantity>
            <o:UnitPrice>148.42</o:UnitPrice>
        </o:Item>
    </p:OrderDetail>
    <p:PaymentMethod>VISA</p:PaymentMethod>
    <p:CustomerDetails>
        <c:Name>Alice Smith</c:Name>
        <c:DeliveryAddress>
            <cmn:Line1>123 Maple Street</cmn:Line1>
            <cmn:Line2>Mill Valley</cmn:Line2>
        </c:DeliveryAddress>
        <c:BillingAddress>
            <cmn:Line1>8 Oak Avenue</cmn:Line1>
            <cmn:Line2>Old Town</cmn:Line2>
        </c:BillingAddress>
    </p:CustomerDetails>
</p:Purchase>
```

El esquema principal que usamos en este ejemplo se llama **Main.xsd**. Como se ve en el extracto de código siguiente, el esquema importa otros tres esquemas: **CommonTypes.xsd**, **CustomerTypes.xsd** y **OrderTypes.xsd**. Para obtener los mismos resultados que en este ejemplo, guarde los extractos de código siguientes en archivos y guárdelos con los nombres de archivo que acabamos de mencionar. Tenga en cuenta que el esquema asigna cada uno de los prefijos **ord**, **pur**, **cmn** y **cust** a algún espacio de nombres (Order types, Purchase types, Common types y Customer types, respectivamente). Esto significa que, en el código generado, las clases que corresponden a Orders, Purchases, Customers, etc. están disponibles bajo el espacio de nombres correspondiente.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://NamespaceTest.com/Purchase"
      xmlns:ord="http://NamespaceTest.com/OrderTypes"
      xmlns:pur="http://NamespaceTest.com/Purchase"
      xmlns:cmn="http://NamespaceTest.com/CommonTypes"
      xmlns:cust="http://NamespaceTest.com/CustomerTypes"
      elementFormDefault="qualified">
    <xs:import schemaLocation="CommonTypes.xsd"
namespace="http://NamespaceTest.com/CommonTypes" />
    <xs:import schemaLocation="CustomerTypes.xsd"
namespace="http://NamespaceTest.com/CustomerTypes" />
    <xs:import schemaLocation="OrderTypes.xsd"
namespace="http://NamespaceTest.com/OrderTypes" />
    <xs:element name="Purchase">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="OrderDetail" type="ord:OrderType" />
                <xs:element name="PaymentMethod" type="cmn:PaymentMethodType" />
                <xs:element ref="pur:CustomerDetails" />
            </xs:sequence>
```


© 2016-2022 Altova GmbH

```
        </xs:complexType>
    </xs:element>
    <xs:element name="CustomerDetails" type="cust:CustomerType" />
</xs:schema>
```

Main.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://NamespaceTest.com/CommonTypes"
      elementFormDefault="qualified">
   <xs:complexType name="AddressType">
      <xs:sequence>
         <xs:element name="Line1" type="xs:string"/>
         <xs:element name="Line2" type="xs:string"/>
      </xs:sequence>
   </xs:complexType>
   <xs:simpleType name="PriceType">
      <xs:restriction base="xs:decimal">
         <xs:fractionDigits value="2"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="PaymentMethodType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="VISA"/>
         <xs:enumeration value="MasterCard"/>
         <xs:enumeration value="Cash"/>
         <xs:enumeration value="AMEX"/>
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

CommonTypes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://NamespaceTest.com/CustomerTypes"
      xmlns:cmn="http://NamespaceTest.com/CommonTypes"
      elementFormDefault="qualified">
    <xs:import schemaLocation="CommonTypes.xsd"
namespace="http://NamespaceTest.com/CommonTypes" />
    <xs:complexType name="CustomerType">
        <xs:sequence>
            <xs:element name="Name" type="xs:string" />
            <xs:element name="DeliveryAddress" type="cmn:AddressType" />
            <xs:element name="BillingAddress" type="cmn:AddressType" />
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

CustomerTypes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://NamespaceTest.com/OrderTypes"
      xmlns:cmn="http://NamespaceTest.com/CommonTypes"
      elementFormDefault="qualified">
    <xs:import schemaLocation="CommonTypes.xsd"
namespace="http://NamespaceTest.com/CommonTypes" />
    <xs:complexType name="OrderType">
        <xs:sequence>
            <xs:element maxOccurs="unbounded" name="Item">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="ProductName" type="xs:string" />
                        <xs:element name="Quantity" type="xs:int" />
                        <xs:element name="UnitPrice" type="cmn:PriceType" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

OrderTypes.xsd

Para completar el ejemplo, siga estos pasos:

1. Guarde en disco todos los esquemas de los extractos de código anteriores sin cambiarles el nombre.
2. Genere el código contenedor de esquema a partir del esquema de más arriba **Main.xsd** siguiendo los pasos descritos en Generar código fuente a partir de esquemas XML o DTD[1142]. Si ha seguido todos los pasos hasta ahora, debería haber generado un programa compilable en el lenguaje de su elección (C++, C# o Java).
3. Añada código a su programa C++, C# o Java desde el extracto que corresponda en cada caso:

   - Espacios de nombres y prefijos XML (C++)[1182]
   - Espacios de nombres y prefijos XML (C#)[1184]
   - Espacios de nombres y prefijos XML (Java)[1185]

27.2.6.1 Espacios de nombres y prefijos XML (C++)

Una vez haya generado código a partir del esquema de ejemplo[1179] se crea una aplicación C++ de prueba, junto con varias bibliotecas de Altova de apoyo. Recuerde que el esquema de ejemplo (Main.xsd) tiene varias declaraciones de espacios de nombres. En consecuencia, el código generado incluye espacios de nombres que corresponden a alias de espacios de nombres (prefijos) del esquema, que son: `Main::ord`, `Main::pur`, `Main::cmn` y `Main::cust`.

Por lo general, para controlar espacios de nombres y prefijos XML con ayuda de las bibliotecas contenedoras de esquemas, debe contar con estos métodos:

- DeclareAllNamespacesFromSchema()[1198]. Llame a este método si quiere declarar los mismos espacios de nombres en su instancia XML que en el esquema. En cambio, si necesita distintos espacios de nombres, como en este ejemplo, entonces use `DeclareNamespace()`. El método

 © 2016-2022 Altova GmbH

`DeclareAllNamespacesFromSchema()` no se usa en este ejemplo porque lo que queremos es crear elementos XML con prefijos que sean ligeramente distintos a los que se declaran en el esquema.

- DeclareNamespace()[1200]. Llame a este método para crear o sobrescribir el atributo del prefijo del espacio de nombres de un elemento. Para ello primero debe crear ese elemento, bien con el método `append()` o con `appendWithPrefix()`, como explicamos a continuación.
- appendWithPrefix().[1202] Use este método para anexar un elemento de instancia con un prefijo concreto. Para crear la instancia XML de este ejemplo bastó con llamar a este método solamente para el elemento raíz. Todos los demás elementos se anexaron usando simplemente append()[1202] y sus prefijos se añadieron automáticamente basándose en los espacios de nombres a los que pertenece el nombre del elemento.

El extracto de código siguiente muestra cómo crear un documento XML con varias declaraciones de espacios de nombres y nombres de elementos prefijados. Más concretamente, genera una instancia de orden de compra, como se puede ver en el Ejemplo: orden de compra[1179]. Es importante tener en cuenta que por motivos ilustrativos en el ejemplo algunos prefijos se han sobrescrito en la instancia XML (es decir, no son exactamente los mismos que los que se declaran en el esquema).

```
void Example()
{
   // Create the XML document and append the root element
   Main::pur::CMain doc = Main::pur::CMain::CreateDocument();
   Main::pur::CPurchaseType purchase = doc.Purchase.appendWithPrefix(_T("p"));

   // Set schema location
   doc.SetSchemaLocation(_T("Main.xsd"));

   // Declare namespaces on root element
   purchase.DeclareNamespace(_T("o"), _T("http://NamespaceTest.com/OrderTypes"));
   purchase.DeclareNamespace(_T("c"), _T("http://NamespaceTest.com/CustomerTypes"));
   purchase.DeclareNamespace(_T("cmn"), _T("http://NamespaceTest.com/CommonTypes"));

   // Append the OrderDetail element
   Main::ord::COrderType order = purchase.OrderDetail.append();
   Main::ord::CItemType item = order.Item.append();
   item.ProductName.append() = _T("Lawnmower");
   item.Quantity.append() = 1;
   item.UnitPrice.append() = 148.42;

   // Append the PaymentMethod element
   Main::cmn::CPaymentMethodTypeType paymentMethod = purchase.PaymentMethod.append();
   paymentMethod.SetEnumerationValue(Main::cmn::CPaymentMethodTypeType::k_VISA);

   // Append the CustomerDetails element
   Main::cust::CCustomerType customer = purchase.CustomerDetails.append();
   customer.Name.append() = _T("Alice Smith");
   Main::cmn::CAddressType deliveryAddress = customer.DeliveryAddress.append();
   deliveryAddress.Line1.append() = _T("123 Maple Street");
   deliveryAddress.Line2.append() = _T("Mill Valley");
   Main::cmn::CAddressType billingAddress = customer.BillingAddress.append();
   billingAddress.Line1.append() = _T("8 Oak Avenue");
   billingAddress.Line2.append() = _T("Old Town");

   // Save to file and release object from memory
```

```
   doc.SaveToFile(_T("Main1.xml"), true);
   doc.DestroyDocument();
}
```

27.2.6.2 Espacios de nombres y prefijos XML (C#)

Una vez haya generado código a partir del esquema de ejemplo[1179] se crea una aplicación C# de prueba, junto con varias bibliotecas de Altova de apoyo. Recuerde que el esquema de ejemplo (Main.xsd) tiene varias declaraciones de espacios de nombres. En consecuencia, el código generado incluye espacios de nombres que corresponden a alias de espacios de nombres (prefijos) del esquema, que son: **`Main::ord`**, **`Main::pur`**, **`Main::cmn`** y **`Main::cust`**.

Por lo general, para controlar espacios de nombres y prefijos XML con ayuda de las bibliotecas contenedoras de esquemas, debe contar con estos métodos:

- DeclareAllNamespacesFromSchema()[1198]: llame a este método si quiere declarar los mismos espacios de nombres en su instancia XML que en el esquema. En cambio, si necesita distintos espacios de nombres, como en este ejemplo, entonces use **`DeclareNamespace()`**. El método **`DeclareAllNamespacesFromSchema()`** no se usa en este ejemplo porque lo que queremos es crear elementos XML con prefijos que sean ligeramente distintos a los que se declaran en el esquema.
- DeclareNamespace()[1216]: llame a este método para crear o sobrescribir el atributo del prefijo del espacio de nombres de un elemento. Para ello primero debe crear ese elemento, bien con el método **`Append()`** o con **`AppendWithPrefix()`**, como explicamos a continuación.
- AppendWithPrefix()[1218]: use este método para anexar un elemento de instancia con un prefijo concreto. Para crear la instancia XML de este ejemplo bastó con llamar a este método solamente para el elemento raíz. Todos los demás elementos se anexaron usando simplemente Append()[1218] y sus prefijos se añadieron automáticamente basándose en los espacios de nombres a los que pertenece el nombre del elemento.

El extracto de código siguiente muestra cómo crear un documento XML con varias declaraciones de espacios de nombres y nombres de elementos prefijados. Más concretamente, genera una instancia de orden de compra, como se puede ver en el Ejemplo: orden de compra[1179]. Es importante tener en cuenta que por motivos ilustrativos en el ejemplo algunos prefijos se han sobrescrito en la instancia XML (es decir, no son exactamente los mismos que los que se declaran en el esquema).

```
protected static void Example()
{
   // Create the XML document and append the root element
   pur.Main2 doc = pur.Main2.CreateDocument();
   pur.PurchaseType purchase = doc.Purchase.AppendWithPrefix("p");

   // Set schema location
   doc.SetSchemaLocation(@"Main.xsd");

   // Declare namespaces on root element
   purchase.DeclareNamespace("o", "http://NamespaceTest.com/OrderTypes");
   purchase.DeclareNamespace("c", "http://NamespaceTest.com/CustomerTypes");
   purchase.DeclareNamespace("cmn", "http://NamespaceTest.com/CommonTypes");

   // Append the OrderDetail element
```


© 2016-2022 Altova GmbH

```
    ord.OrderType order = purchase.OrderDetail.Append();
    ord.ItemType item = order.Item.Append();
    item.ProductName.Append().Value = "Lawnmower";
    item.Quantity.Append().Value = 1;
    item.UnitPrice.Append().Value = 148.42M;

    // Append the PaymentMethod element
    cmn.PaymentMethodTypeType paymentMethod = purchase.PaymentMethod.Append();
    paymentMethod.EnumerationValue = cmn.PaymentMethodTypeType.EnumValues.eVISA;

    // Append the CustomerDetails element
    cust.CustomerType customer = purchase.CustomerDetails.Append();
    customer.Name.Append().Value = "Alice Smith";
    cmn.AddressType deliveryAddress = customer.DeliveryAddress.Append();
    deliveryAddress.Line1.Append().Value = "123 Maple Street";
    deliveryAddress.Line2.Append().Value = "Mill Valley";
    cmn.AddressType billingAddress = customer.BillingAddress.Append();
    billingAddress.Line1.Append().Value = "8 Oak Avenue";
    billingAddress.Line2.Append().Value = "Old Town";

    // Save to file
    doc.SaveToFile("PurchaseOrder.xml", true);
}
```

27.2.6.3 Espacios de nombres y prefijos XML (Java)

Una vez haya generado código a partir del esquema de ejemplo[1179] se crea una aplicación Java de prueba, junto con varias bibliotecas de Altova de apoyo. Recuerde que el esquema de ejemplo (Main.xsd) tiene varias declaraciones de espacios de nombres. En consecuencia, el código generado incluye espacios de nombres que corresponden a alias de espacios de nombres (prefijos) del esquema, que son: **com.Main.ord**, **com.Main.pur**, **com.Main.cmn** y **com.Main.cust**. .

Por lo general, para controlar espacios de nombres y prefijos XML con ayuda de las bibliotecas contenedoras de esquemas, debe contar con estos métodos:

- declareAllNamespacesFromSchema()[1232]: llame a este método si quiere declarar los mismos espacios de nombres en su instancia XML que en el esquema. En cambio, si necesita distintos espacios de nombres, como en este ejemplo, entonces use **DeclareNamespace()**. El método **DeclareAllNamespacesFromSchema()** no se usa en este ejemplo porque lo que queremos es crear elementos XML con prefijos que sean ligeramente distintos a los que se declaran en el esquema.
- declareNamespace()[1234]: llame a este método para crear o sobrescribir el atributo del prefijo del espacio de nombres de un elemento. Para ello primero debe crear ese elemento, bien con el método **append()** o con **appendWithPrefix()**, como explicamos a continuación.
- appendWithPrefix()[1236]: Use este método para anexar un elemento de instancia con un prefijo concreto. Para crear la instancia XML de este ejemplo bastó con llamar a este método solamente para el elemento raíz. Todos los demás elementos se anexaron usando simplemente append()[1202] y sus prefijos se añadieron automáticamente basándose en los espacios de nombres a los que pertenece el nombre del elemento.

El extracto de código siguiente muestra cómo crear un documento XML con varias declaraciones de espacios de nombres y nombres de elementos prefijados. Más concretamente, genera una instancia de orden de

compra, como se puede ver en el Ejemplo: orden de compra[1179]. Es importante tener en cuenta que por motivos ilustrativos en el ejemplo algunos prefijos se han sobrescrito en la instancia XML (es decir, no son exactamente los mismos que los que se declaran en el esquema).

```
protected static void example() throws Exception {
   // Create the XML document and append the root element
   com.Main.pur.Main2 doc = com.Main.pur.Main2.createDocument();
   com.Main.pur.PurchaseType purchase = doc.Purchase.appendWithPrefix("p");

   // Set schema location
   doc.setSchemaLocation("Main.xsd");

   // Declare namespaces on root element
   purchase.declareNamespace("o", "http://NamespaceTest.com/OrderTypes");
   purchase.declareNamespace("c", "http://NamespaceTest.com/CustomerTypes");
   purchase.declareNamespace("cmn", "http://NamespaceTest.com/CommonTypes");

   // Append the OrderDetail element
   com.Main.ord.OrderType order = purchase.OrderDetail.append();
   com.Main.ord.ItemType item = order.Item.append();
   item.ProductName.append().setValue("Lawnmower");
   item.Quantity.append().setValue(1);
   java.math.BigDecimal price = new java.math.BigDecimal("148.42");
   item.UnitPrice.append().setValue(price);

   // Append the PaymentMethod element
   com.Main.cmn.PaymentMethodTypeType paymentMethod = purchase.PaymentMethod.append();
   paymentMethod.setEnumerationValue(com.Main.cmn.PaymentMethodTypeType.EVISA);

   // Append the CustomerDetails element
   com.Main.cust.CustomerType customer = purchase.CustomerDetails.append();
   customer.Name.append().setValue("Alice Smith");
   com.Main.cmn.AddressType deliveryAddress = customer.DeliveryAddress.append();
   deliveryAddress.Line1.append().setValue("123 Maple Street");
   deliveryAddress.Line2.append().setValue("Mill Valley");
   com.Main.cmn.AddressType billingAddress = customer.BillingAddress.append();
   billingAddress.Line1.append().setValue("8 Oak Avenue");
   billingAddress.Line2.append().setValue("Old Town");

   // Save to file
   doc.saveToFile("PurchaseOrder.xml", true);
}
```


© 2016-2022 Altova GmbH

27.3 Referencia de las clases generadas (C++)

Esta sección describe las clases C++ que se generan con XMLSpy a partir de una DTD o esquema XML (*véase* *Generar código a partir de esquemas XML o DTD* 1142). Puede integrar estas clases en su código para leer, modificar y escribir documentos XML.

Nota: el código generado incluye otras clases compatibles que no se enumeran en este apartado y que pueden cambiar en el futuro.

27.3.1 altova::DateTime

Esta clase permite procesar atributos o elementos XML de tipo hora y fecha (p. ej. `xs:dateTime`).

Constructores

Nombre	Descripción
`DateTime()`	Inicializa una instancia nueva de la clase `DateTime` en la medianoche 12:00:00 del 1 de enero de 0001.
`DateTime(__int64 value, short timezone)`	Inicializa una instancia nueva de la clase `DateTime`. El parámetro `value` representa el número de ciclos (intervalos de 100 nanosegundos) que han pasado desde la medianoche 12:00:00 del 1 de enero de 0001.
`DateTime(int year, unsigned char month, unsigned char day, unsigned char hour, unsigned char minute, double second)`	Inicializa una instancia nueva de la clase `DateTime` en el año, mes, día, hora, minuto y segundo dados como argumento.
`DateTime(int year, unsigned char month, unsigned char day, unsigned char hour, unsigned char minute, double second, short timezone)`	Inicializa una instancia nueva de la clase `DateTime` en el año, mes, día, hora, minuto, segundo y zona horaria dados como argumento. La zona horaria se expresa en minutos y puede ser positiva o negativa. Por ejemplo, la zona horaria "UTC-01:00" se expresa como "-60".

Métodos

Nombre	Descripción
`unsigned char Day() const`	Devuelve el día del mes del objeto `DateTime` actual. Los valores devueltos están comprendidos entre 1 y 31.
`int DayOfYear() const`	Devuelve el día del año del objeto `DateTime` actual. Los valores devueltos están comprendidos entre 1 y 366.

Nombre	Descripción
`bool HasTimezone() const`	Devuelve el valor binario **true** si el objeto `DateTime` actual tiene definida una zona horaria. De lo contrario devuelve **false**.
`unsigned char Hour() const`	Devuelve la hora del objeto `DateTime` actual. Los valores devueltos están comprendidos entre 0 y 23.
`static bool IsLeapYear(int year)`	Devuelve el valor binario **true** si el año de la clase `DateTime` es un año bisiesto. De lo contrario devuelve **false**.
`unsigned char Minute() const`	Devuelve el minuto del objeto `DateTime` actual. Los valores devueltos están comprendidos entre 0 y 59.
`unsigned char Month() const`	Devuelve el mes del objeto `DateTime` actual. Los valores devueltos están comprendidos entre 1 y 12.
`__int64 NormalizedValue() const`	Devuelve el valor del objeto `DateTime` expresado como UTC.
`double Second() const`	Devuelve el segundo del objeto `DateTime` actual. Los valores devueltos están comprendidos entre 0 y 59.
`void SetTimezone(short tz)`	Establece la zona horaria del objeto `DateTime` actual en el valor de zona horaria dado como argumento. El argumento **tz** se expresa en minutos y puede ser positivo o negativo.
`short Timezone() const`	Devuelve la zona horaria en minutos del objeto `DateTime` actual. Antes de usar este método asegúrese de que el objeto realmente tiene una zona horaria (llamando al método `HasTimezone()`).
`__int64 Value() const`	Devuelve el valor del objeto `DateTime` expresado en el número de ciclos (intervalos de 100 nanosegundos) que han pasado desde la medianoche 12:00:00 del 1 de enero de 0001.
`int Weekday() const`	Devuelve el día de la semana del objeto `DateTime` actual como entero. Los valores están comprendidos entre 0 y 6, siendo 0 el lunes (ISO-8601).
`int Weeknumber() const`	Devuelve el número de la semana del año del objeto `DateTime` actual. Los valores devueltos son conformes a ISO-8601.
`int WeekOfMonth() const`	Devuelve el número de la semana del mes del objeto `DateTime` actual. Los valores devueltos son conformes a ISO-8601.
`int Year() const`	Devuelve el año del objeto `DateTime` actual.

Ejemplo

```
void Example()
{
```


© 2016-2022 Altova GmbH

```
	// inicializar una nueva instancia de DateTime en la medianoche 12:00:00 del 1 de
enero de 0001
	altova::DateTime dt1 = altova::DateTime();

	// inicializar una nueva instancia de DateTime usando el año, mes, día, hora,
minuto y segundo
	altova::DateTime dt2 = altova::DateTime(2015, 11, 10, 9, 8, 7);

	// inicializar una nueva instancia de DateTime usando el año, mes, día, hora,
minuto, segundo y la zona horaria UTC +01:00
	 altova::DateTime dt = altova::DateTime(2015, 11, 22, 13, 53, 7, 60);

	// Obtener el valor de este objeto DateTime
	std::cout << "El número de ciclos del objeto DateTime es: " << dt.Value() <<
std::endl;

	// Obtener el año
	cout << "El año es: " << dt.Year() << endl;
	// Obtener el mes
	cout << "El mes es: " << (int)dt.Month() << endl;
	// Obtener el día del mes
	cout << "El día del mes es: " << (int) dt.Day() << endl;
	// Obtener el día del año
	cout << "El día del año es: " << dt.DayOfYear() << endl;
	// Obtener la hora
	cout << "La hora es: " << (int) dt.Hour() << endl;
	// Obtener el minuto
	cout << "El minuto es: " << (int) dt.Minute() << endl;
	// Obtener el segundo
	cout << "El segundo es: " << dt.Second() << endl;
	// Obtener el día de la semana
	cout << "El día de la semana es: " << dt.Weekday() << endl;
	// Obtener el número de la semana
	cout << "La semana del año es: " << dt.Weeknumber() << endl;
	// Obtener la semana del mes
	cout << "La semana del mes es: " << dt.WeekOfMonth() << endl;

	// Comprobar si una instancia de DateTime tiene una zona horaria
	if (dt.HasTimezone() == TRUE)
	{
		// generar el valor de la zona horaria
		cout <<  "La zona horaria es: " << dt.Timezone() << endl;
	}
	else
	{
		cout <<  "No se definió la zona horaria." << endl;
	}

	// Construir un objeto DateTime con una zona horaria UTC+01:00 (Viena)
	altova::DateTime vienna_dt = DateTime(2015, 11, 23, 14, 30, 59, +60);
	// Generar el resultado en formato legible
	cout << "La hora de Viena: "
		 << (int) vienna_dt.Month()
		 << "-" << (int) vienna_dt.Day()
		 << " " << (int) vienna_dt.Hour()
```

```
               << ":" << (int) vienna_dt.Minute()
               << ":" << (int) vienna_dt.Second()
               << endl;

      // Convertir el valor en hora UTC
      DateTime utc_dt = DateTime(vienna_dt.NormalizedValue());
      // Generar el resultado en formato legible
      cout << "La hora UTC:    "
             << (int) utc_dt.Month()
             << "-" << (int) utc_dt.Day()
             << " " << (int) utc_dt.Hour()
             << ":" << (int) utc_dt.Minute()
             << ":" << (int) utc_dt.Second()
             << endl;

      // Comprobar si el año es bisiest
      int year = 2016;
      if( altova::DateTime::IsLeapYear(year) )
      { cout << year << " es año bisiesto" << endl; }
      else
      { cout << year << " no es año bisiesto" << endl; }
}
```


© 2016-2022 Altova GmbH

27.3.2 altova::Duration

Esta clase permite procesar atributos o elementos XML de tipo `xs:duration`.

Constructores

Nombre	Descripción
`Duration()`	Inicializa una instancia nueva de la clase `Duration` en un valor vacío.
`Duration(const DayTimeDuration& dt)`	Inicializa una instancia nueva de la clase `Duration` en una duración definida por el argumento `dt` (*véase* *altova::DayTimeDuration*[1193]).
`Duration(const YearMonthDuration& ym)`	Inicializa una instancia nueva de la clase `Duration` en la duración definida por el argumento `ym` (véase *altova::YearMonthDuration*[1194]).
`Duration(const YearMonthDuration& ym, const DayTimeDuration& dt)`	Inicializa una instancia nueva de la clase `Duration` en la duración definida por los argumentos `dt` y `ym` (*véase* *altova::YearMonthDuration*[1194] *y* *altova::DayTimeDuration*[1193]).

Métodos

Nombre	Descripción
`int Days() const`	Devuelve el número de días de la instancia de `Duration` actual.
`DayTimeDuration DayTime() const`	Devuelve el día y la duración en horas de la instancia de `Duration` actual expresada como objeto `DayTimeDuration` *(véase* *altova::DayTimeDuration*[1193]).
`int Hours() const`	Devuelve el número de horas de la instancia de `Duration` actual.
`bool IsNegative() const`	Devuelve el valor binario **true** si la instancia de `Duration` actual es negativa.
`bool IsPositive() const`	Devuelve el valor binario **true** si la instancia de `Duration` actual es positiva.
`int Minutes() const`	Devuelve el número de minutos de la instancia de `Duration` actual.
`int Months() const`	Devuelve el número de meses de la instancia de `Duration` actual.

Nombre	Descripción
`double Seconds() const`	Devuelve el número de segundos de la instancia de `Duration` actual.
`YearMonthDuration YearMonth() const`	Devuelve el año y la duración en meses de la instancia de `Duration` actual expresada como objeto `YearMonthDuration` (*véase* *altova::YearMonthDuration*[1194]).
`int Years() const`	Devuelve el número de años de la instancia de `Duration` actual.

Ejemplo

En este ejemplo se crea un objeto `Duration` nuevo y se leen sus valores.

```
void ExampleDuration()
{
	// Crear un objeto Duration vacío
	altova::Duration empty_duration = altova::Duration();

	// Crear un objeto Duration usando un valor de duración actual
	altova::Duration duration1 = altova::Duration(empty_duration);

	// Crear una duración YearMonth de seis años y cinco meses
	altova::YearMonthDuration yrduration = altova::YearMonthDuration(6, 5);

	// Crear una duración DayTime de cuatro días, tres horas, dos minutos y un segundo
	altova::DayTimeDuration dtduration = altova::DayTimeDuration(4, 3, 2, 1);

	// Crear un objeto Duration combinando las dos duraciones creadas previamente
	altova::Duration duration = altova::Duration(yrduration, dtduration);

	// Obtener el número de años de esta instancia de Duration
	cout << "Años:  " << duration.Years() << endl;

	// Obtener el número de meses de esta instancia de Duration
	cout << "Meses: " << duration.Months() << endl;

	// Obtener el número de días de esta instancia de Duration
	cout << "Días:   " << duration.Days() << endl;

	// Obtener el número de horas de esta instancia de Duration
	cout << "Horas:  " << duration.Hours() << endl;

	// Obtener el número de minutos de esta instancia de Duration
	cout << "Minutos: " << duration.Minutes() << endl;

	// Obtener el número de segundos de esta instancia de Duration
	cout << "Segundos: " << duration.Seconds() << endl;
}
```


© 2016-2022 Altova GmbH

27.3.3 altova::DayTimeDuration

Esta clase permite procesar tipos XML Schema de duración compuestos por un día y una hora.

Constructores

Nombre	Descripción
`DayTimeDuration()`	Inicializa una instancia nueva de la clase `DayTimeDuration` en un valor vacío.
`DayTimeDuration(int days, int hours, int minutes, double seconds)`	Inicializa una instancia nueva de la clase `DayTimeDuration` en el número de días, horas, minutos y segundos dados como argumento.
`explicit DayTimeDuration(__int64 value)`	Inicializa una instancia nueva de la clase `DayTimeDuration` en una duración compuesta por tantos ciclos (intervalos de 100 nanosegundos) como aporta el argumento **value**.

Métodos

Nombre	Descripción
`int Days() const`	Devuelve el número de días de la instancia de `DayTimeDuration` actual.
`int Hours() const`	Devuelve el número de horas de la instancia de `DayTimeDuration` actual.
`bool IsNegative() const`	Devuelve el valor binario **true** si la instancia de `DayTimeDuration` actual es negativa.
`bool IsPositive() const`	Devuelve el valor binario **true** si la instancia de `DayTimeDuration` actual es positiva.
`int Minutes() const`	Devuelve el número de minutos de la instancia de `DayTimeDuration` actual.
`double Seconds() const`	Devuelve el número de segundos de la instancia de `DayTimeDuration` actual.
`__int64 Value() const`	Devuelve el valor (en ciclos) de la instancia de `DayTimeDuration` actual.

27.3.4 altova::YearMonthDuration

Esta clase permite procesar tipos XML Schema de duración compuestos por un año y un mes.

Constructores

Nombre	Descripción
`YearMonthDuration()`	Inicializa una instancia nueva de la clase `YearMonthDuration` en un valor vacío.
`YearMonthDuration(int years, int months)`	Inicializa una instancia nueva de la clase `YearMonthDuration` en el número de años y meses dados por los argumentos **years** y **months**.
`explicit YearMonthDuration(int value)`	Inicializa una instancia nueva de la clase `YearMonthDuration` en una duración compuesta por tantos ciclos (intervalos de 100 nanosegundos) como da el argumento **value**.

Métodos

Nombre	Descripción
`bool IsNegative() const`	Devuelve el valor binario **true** si la instancia de `YearMonthDuration` actual es negativa.
`bool IsPositive() const`	Devuelve el valor binario **true** si la instancia de `YearMonthDuration` actual es positiva.
`int Months() const`	Devuelve el número de meses de la instancia de `YearMonthDuration` actual.
`int Value() const`	Devuelve el valor (en ciclos) de la instancia de `YearMonthDuration` actual.
`int Years()`	Devuelve el número de años de la instancia de `YearMonthDuration` actual.

27.3.5 altova::meta::Attribute

Esta clase permite acceder a la información del esquema sobre clases generadas a partir de atributos. Recuerde que esta clase no tiene como objetivo aportar información dinámica sobre las distintas instancias de un atributo de un documento XML. Por el contrario, permite obtener información mediante programación sobre cualquier atributo que esté definido en el esquema XML.

© 2016-2022 Altova GmbH

Métodos

Nombre	Descripción
`SimpleType GetDataType()`	Devuelve el tipo del contenido del atributo.
`string_type GetLocalName()`	Devuelve el nombre local del atributo.
`string_type GetNamespaceURI()`	Devuelve el URI de espacio de nombres del atributo.
`bool IsRequired()`	Devuelve true si el atributo es obligatorio.

Operadores

Nombre	Descripción
`bool operator()`	Devuelve true si este no es el atributo NULL.
`bool operator!()`	Devuelve true si este es el atributo NULL.

27.3.6 altova::meta::ComplexType

Esta clase permite acceder a información del esquema sobre clases generadas a partir de tipos complejos. Recuerde que esta clase no tiene como objetivo aportar información dinámica sobre las distintas instancias de un tipo complejo de un documento XML. Por el contrario, permite obtener información mediante programación sobre cualquier tipo complejo que esté definido en el esquema XML.

Métodos

Nombre	Descripción
`Attribute FindAttribute(const char_type* localName, const char_type* namespaceURI)`	Busca el atributo que tiene el nombre local y el URI de espacio de nombres especificados.
`Element FindElement(const char_type* localName, const char_type* namespaceURI)`	Busca el elemento que tiene el nombre local y el URI de espacio de nombres especificados.
`std::vector<Attribute> GetAttributes()`	Devuelve una lista de todos los atributos.
`ComplexType GetBaseType()`	Devuelve el tipo base de este tipo.
`SimpleType GetContentType()`	Devuelve el tipo simple del contenido.

Nombre	Descripción
`std::vector<Element> GetElements()`	Devuelve una lista de todos los elementos.
`string_type GetLocalName()`	Devuelve el nombre local del tipo.
`string_type GetNamespaceURI()`	Devuelve el URI de espacio de nombres del tipo.

Operadores

Nombre	Descripción
`bool operator()`	Devuelve true si este no es el tipo complejo NULL.
`bool operator!()`	Devuelve true si este es el tipo complejo NULL.

27.3.7 altova::meta::Element

Esta clase permite acceder a la información del esquema sobre clases generadas a partir de elementos. Recuerde que esta clase no tiene como objetivo aportar información dinámica sobre las distintas instancias de un elemento de un documento XML. Por el contrario, permite obtener información mediante programación sobre cualquier elemento que esté definido en el esquema XML.

Métodos

Nombre	Descripción
`ComplexType GetDataType()`	Devuelve el tipo del elemento. Recuerde que este siempre es un tipo complejo incluso si se declaró como tipo simple en el esquema original. Use GetContentType() del objeto devuelto para obtener el tipo de contenido simple.
`string_type GetLocalName()`	Devuelve el nombre local del elemento.
`unsigned int GetMaxOccurs()`	Devuelve el valor maxOccurs que está definido en el esquema.
`unsigned int GetMinOccurs()`	Devuelve el valor minOccurs que está definido en el esquema.
`string_type GetNamespaceURI()`	Devuelve el URI de espacio de nombres del elemento.

© 2016-2022 Altova GmbH

Operadores

Nombre	Descripción
`bool operator()`	Devuelve false si este no es el elemento NULL.
`bool operator!()`	Devuelve true si este es el elemento NULL.

27.3.8 altova::meta::SimpleType

Esta clase permite acceder a la información del esquema sobre clases generadas a partir de tipos simples. Recuerde que esta clase no tiene como objetivo aportar información dinámica sobre las distintas instancias de un tipo simple de un documento XML. Por el contrario, permite obtener información mediante programación sobre cualquier tipo simple que esté definido en el esquema XML.

Métodos

Nombre	Descripción
`SimpleType GetBaseType()`	Devuelve el tipo base de este tipo.
`std::vector<string_type> GetEnumerations()`	Devuelve una lista de todas las facetas de enumeración.
`unsigned int GetFractionDigits()`	Devuelve el valor de esta faceta.
`unsigned int GetLength()`	Devuelve el valor de esta faceta.
`string_type GetLocalName()`	Devuelve el nombre local de este tipo.
`string_type GetMaxExclusive()`	Devuelve el valor de esta faceta.
`string_type GetMaxInclusive()`	Devuelve el valor de esta faceta.
`unsigned int GetMaxLength()`	Devuelve el valor de esta faceta.
`string_type GetMinExclusive()`	Devuelve el valor de esta faceta.
`string_type GetMinInclusive()`	Devuelve el valor de esta faceta.
`unsigned int GetMinLength()`	Devuelve el valor de esta faceta.
`string_type GetNamespaceURI()`	Devuelve el URI de espacio de nombres del tipo.

Nombre	Descripción
`std::vector<string_type> GetPatterns()`	Devuelve una lista de todas las facetas de patrón.
`unsigned int GetTotalDigits()`	Devuelve el valor de esta faceta.
`WhitespaceType GetWhitespace()`	Devuelve el valor de la faceta de espacio en blanco, que es uno de estos: • Whitespace_Unknown • Whitespace_Preserve • Whitespace_Replace • Whitespace_Collapse

Operadores

Nombre	Descripción
`bool operator()`	Devuelve true si este no es el SimpleType NULL.
`bool operator!()`	Devuelve true si este es el SimpleType NULL.

27.3.9 [SuEsquema]::[CDoc]

Cuando se genera código a partir de un esquema XML, el código generado ofrece una clase de documento que se llama igual que el esquema. Esta clase tiene como miembros todos los elementos raíz posibles, así como los métodos que aparecen a continuación. Observe que en los nombres de método, `CDoc` corresponde al nombre de la clase de documento generada.

Métodos

Nombre	Descripción
`static CDoc CreateDocument()`	Crea un documento XML nuevo vacío. Debe publicarse con DestroyDocument().
`static void DeclareAllNamespacesFromSchema(TypeBase& node)`	Declara todos los espacios de nombres del esquema XML en el elemento dado como argumento (que suele ser el elemento XML raíz). Es útil llamar a este método si su esquema tiene varias declaraciones de espacios de nombres, cada una de ellas está asignada a un prefijo y quiere declararlas todas en el elemento dado como argumento.
`void DestroyDocument()`	Destruye un documento. Todas las referencias al documento y a sus nodos se invalidarán. Esta llamada debe hacerse cuando se termina de trabajar con un documento.

© 2016-2022 Altova GmbH

Nombre	Descripción
`static CDoc LoadFromBinary(const std:vector<unsigned char>& xml)`	Carga un documento XML desde una matriz de bytes.
`static CDoc LoadFromFile(const string_type& fileName)`	Carga un documento XML desde un archivo.
`static CDoc LoadFromString(const string_type& xml)`	Carga un documento XML desde una cadena de texto.
`std::vector<unsigned char> SaveToBinary(bool prettyPrint)`	Guarda un documento XML en una matriz de bytes. Si su valor es true, el argumento `prettyPrint` ajusta el formato del documento XML para facilitar su lectura.
`std::vector<unsigned char> SaveToBinary(bool prettyPrint, const string_type & encoding)`	Guarda un documento XML en una matriz de bytes (con formato pretty-print opcional) con la codificación especificada.
`std::vector<unsigned char> SaveToBinary(bool prettyPrint, const string_type & encoding, bool bBigEndian, bool bBOM)`	Guarda un documento XML en una matriz de bytes (con formato pretty-print opcional) con la codificación especificada. Para codificaciones Unicode se puede especificar el orden de bytes y la marca BOM Unicode.
`void SaveToFile(const string_type & fileName, bool prettyPrint)`	Guarda un documento XML en un archivo, con formato pretty-print opcional.
`void SaveToFile(const string_type & fileName, bool omitXmlDecl)`	Guarda un documento XML en un archivo. Si el argumento `omitXmlDecl` tiene el valor true, la declaración XML no se escribirá.
`void SaveToFile(const string_type & fileName, bool omitXmlDecl, const string_type & encoding)`	Guarda un documento XML en un archivo. Si el argumento `omitXmlDecl` tiene el valor true, la declaración XML no se escribirá.
`void SaveToFile(const string_type & fileName, bool prettyPrint, bool omitXmlDecl, const string_type & encoding, bool bBigEndian, bool bBOM)`	Guarda un documento XML en un archivo (con formato pretty-print opcional) con la codificación especificada. Para codificaciones Unicode se puede especificar el orden de bytes y la marca BOM Unicode.
`void SaveToFile(const string_type & fileName, bool prettyPrint, bool omitXmlDecl, const string_type & encoding, bool bBigEndian, bool bBOM, const string_type & lineend)`	Guarda un documento XML en un archivo(con formato pretty-print opcional) con la codificación y el final de línea especificados. Para codificaciones Unicode se puede especificar el orden de bytes y la marca BOM Unicode. Este método solo está disponible si generó código para la biblioteca XML Xerces3 (*véase* *Opciones del generador de código* 1239).
`void SaveToFile(const string_type& fileName, bool prettyPrint, bool omitXmlDecl, const string_type & encoding, const string_type & lineend)`	Guarda un documento XML en un archivo, (con formato pretty-print opcional) con la codificación y el final de línea especificados.

Nombre	Descripción
	Este método solo está disponible si generó código para la biblioteca XML Xerces3 (*véase* Opciones del generador de código[1239]).
`void SaveToFile(const string_type & fileName, bool prettyPrint, const string_type & encoding)`	Guarda un documento XML en un archivo, (con formato pretty-print opcional) con la codificación especificada.
`void SaveToFile(const string_type& fileName, bool prettyPrint, const string_type & encoding, bool bBigEndian, bool bBOM)`	Guarda un documento XML en un archivo, (con formato pretty-print opcional) con la codificación especificada. Para codificaciones Unicode se puede especificar el orden de bytes y la marca BOM Unicode.
`void SaveToFile(const string_type& fileName, bool prettyPrint, const string_type & encoding, bool bBigEndian, bool bBOM, const string_type & lineend)`	Guarda un documento XML en un archivo con la codificación y el final de línea especificados. Para codificaciones Unicode se puede especificar el orden de bytes y la marca BOM Unicode. Este método solo está disponible si generó código para la biblioteca XML Xerces3 (*véase* Opciones del generador de código[1239]).
`void SaveToFile(const string_type& fileName, bool prettyPrint, const string_type & encoding, const string_type & lineend)`	Guarda un documento XML en un archivo, (con formato pretty-print opcional) con la codificación y el final de línea especificados. Este método solo está disponible si generó código para la biblioteca XML Xerces3 (véase Opciones del generador de código[1239]).
`string_type SaveToString(bool prettyPrint)`	Guarda un documento XML en una cadena de texto, con formato pretty-print opcional.
`string_type SaveToString(bool prettyPrint, bool omitXmlDecl)`	Guarda un documento XML en una cadena de texto, con formato pretty-print opcional. Si el argumento `omitXmlDecl` tiene el valor true, la declaración XML no se escribirá.
`void SetDTDLocation(const string_type & dtdLocation)`	Agrega una declaración DOCTYPE con el ID de sistema especificado. Es obligatorio que ya exista un elemento raíz. Este método es incompatible con MSXML porque no se puede agregar una declaración DOCTYPE a un documento que está en memoria.
`void SetSchemaLocation(const string_type & schemaLocation)`	Agrega un atributo `xsi:schemaLocation` o `xsi:noNamespaceSchemaLocation` al elemento raíz. Es obligatorio que ya exista un elemento raíz.

27.3.10 [SuEsquema]::[ElementType]

Esta clase permite manipular elementos XML desde el esquema. Los métodos de esta clase se pueden llamar en elementos, pero no en el documento XML. Observe que no necesita instanciar la clase directamente para

© 2016-2022 Altova GmbH

poder llamar a los métodos de esta clase. Cualquier elemento creado con los métodos **append()** o **appendWithPrefix()** es de tipo **append()** or **appendWithPrefix()**.

Métodos

Nombre	Descripción

`void DeclareNamespace(const string_type prefix, const string_type nsURI)`	Observe que para llamar al método no necesita instanciar la clase TypeBase directamente. Cualquier elemento que cree usando los métodos **append()** o **appendWithPrefix()** hereda del tipo TypeBase.
	
	Este método toma dos argumentos de tipo cadena: el prefijo y el URI de espacio de nombres que quiere usar. El prefijo dado como argumento se asigna al valor del URI de espacio de nombres dado como argumento. Si el prefijo dado como argumento está vacío, el método crea o sobrescribe la declaración de espacio de nombres predeterminada del elemento.
	
	Por ejemplo, imagine que el documento XML tiene un elemento XML llamado "purchase". Si llama a
	
	`purchase.DeclareNamespace(_T("ord"), _T("http://OrderTypes"));`
	
	entonces el documento XML se convierte en
	
	`<purchase xmlns:ord="http://OrderTypes" />`
	
	Otro ejemplo. Si llama a:
	
	`purchase.DeclareNamespace(_T(""), _T("http://OrderTypes"));`
	
	entonces el documento XML se convierte en
	
	`<purchase xmlns="http://OrderTypes" />`
	
	Nota: el espacio de nombres declarado se usa al anexar elementos secundarios o atributos posteriormente, conforme a estas normas:
	
	1. Si el espacio de nombres secundario es el predeterminado, use un prefijo vacío.
	2. Si el espacio de nombres secundario es igual que el del elemento matriz, use el prefijo matriz.
	3. De lo contrario, busque el prefijo más cercano desde el

Nombre	Descripción
	elemento matriz hasta el principio usando el algoritmo de búsqueda que se describe en la sección "B.2: Namespace Prefix Lookup" de https://www.w3.org/TR/2002/WD-DOM-Level-3-Core-20021022/namespaces-algorithms.html. 4. Si no se encuentra ningún prefijo para el espacio de nombres del elemento, entonces use un prefijo vacío.

27.3.11 [SuEsquema]::MemberAttribute

Cuando se genera código a partir de un esquema XML, se crea una clase como esta por cada atributo miembro de un tipo.

Métodos

Nombre	Descripción
`bool exists()`	Devuelve true si el atributo existe.
`int GetEnumerationValue()`	Solo se genera para tipos de enumeración. Devuelve una de las constantes generadas para los posibles valores o "no válido" si el valor no coincide con ninguno de los valores enumerados en el esquema.
`altova::meta::Attribute info()`	Devuelve un objeto para consultar información del esquema (*véase* `altova::meta::Attribute` 1194).
`void remove()`	Quita el atributo de su elemento primario.
`void SetEnumerationValue(int)`	Solo se genera para tipos de enumeración. Pase una de las constantes generadas para los posibles valores a este método para establecer el valor.

27.3.12 [SuEsquema]::MemberElement

Cuando se genera código a partir de un esquema XML, se crea una clase como esta por cada elemento miembro de un tipo. En la tabla que aparece a continuación `MemberType` corresponde al nombre del elemento miembro propiamente dicho.

© 2016-2022 Altova GmbH

Métodos

Nombre	Descripción
`Iterator<MemberType> all()`	Devuelve un objeto para recorrer instancias del elemento miembro.
`MemberType append()`	Crea un elemento nuevo y lo anexa a su primario.
`MemberType appendWithPrefix(string_type prefix)`	Crea un elemento nuevo que tiene el prefijo dado como argumento y lo anexa a su elemento superior. Para ver un ejemplo consulte Ejemplo: orden de compra 1179.
`unsigned int count()`	Devuelve el recuento de elementos.
`int GetEnumerationValue()`	Solo se genera para tipos de enumeración. Devuelve una de las constantes generadas para los posibles valores o "no válido" si el valor no coincide con ninguno de los valores enumerados en el esquema.
`bool exists()`	Devuelve true si existe al menos un elemento.
`MemberType first()`	Devuelve la primera instancia del elemento miembro.
`MemberType operator[](unsigned int index)`	Devuelve el elemento miembro especificado por el índice.
`altova::meta::Element info()`	Devuelve un objeto para consultar información del esquema (*véase* *altova::meta::Element* 1196).
`MemberType last()`	Devuelve la última instancia del elemento miembro.
`void remove()`	Elimina todas las instancias del elemento en su primario.
`void removeAt(unsigned int index)`	Elimina la instancia del elemento especificada por el índice.
`void SetEnumerationValue(int)`	Solo se genera para tipos de enumeración. Pase una de las constantes generadas para los posibles valores a este método para establecer el valor.

27.4 Referencia de las clases generadas (C#)

Esta sección describe las clases C# que se generan con XMLSpy a partir de una DTD o esquema XML (*véase* *Generar código a partir de esquemas XML o DTD* 1142). Puede integrar estas clases en su código para leer, modificar y escribir documentos XML.

Nota: el código generado incluye otras clases compatibles que no se enumeran en este apartado y que pueden cambiar en el futuro.

27.4.1 Altova.Types.DateTime

Esta clase permite procesar atributos o elementos XML que tienen tipos de fecha y hora (p. ej. `xs:dateTime`).

Constructores

	Nombre	Descripción
	`DateTime(DateTime obj)`	Inicializa una instancia nueva de la clase `DateTime` en el objeto `DateTime` dado como argumento.
	`DateTime(System.DateTime newvalue)`	Inicializa una instancia nueva de la clase `DateTime` en el objeto `System.DateTime` dado como argumento.
	`DateTime(int year, int month, int day, int hour, int minute, double second, int offsetTZ)`	Inicializa una instancia nueva de la clase `DateTime` en el año, mes, día, hora, minuto, segundo y diferencia de zona horaria dados como argumentos.
	`DateTime(int year, int month, int day, int hour, int minute, double second)`	Inicializa una instancia nueva de la clase `DateTime` en el año, mes, día, hora, minuto y segundo dados como argumentos.
	`DateTime(int year, int month, int day)`	Inicializa una instancia nueva de la clase `DateTime` en el año, mes y día dados como argumentos.

Propiedades

	Nombre	Descripción
	`bool HasTimezone`	Obtiene un valor booleano que indica si `DateTime` tiene una zona horaria.
	`static DateTime Now`	Obtiene un objeto `DateTime` que está definido con la fecha y hora actuales de este equipo.
	`short TimezoneOffset`	Obtiene o establece la diferencia de zona horaria, en minutos, del objeto `DateTime`.

© 2016-2022 Altova GmbH

	Nombre	Descripción
	`System.DateTime Value`	Obtiene o establece el valor del objeto `DateTime` como valor de un `System.DateTime`.

Métodos

	Nombre	Descripción
	`int CompareTo(object obj)`	La clase `DateTime` implementa la interfaz `IComparable`. Este método compara la instancia actual de `DateTime` con otro objeto y devuelve un entero que indica si la posición de la instancia actual es anterior, posterior o igual que la del otro objeto en el criterio de ordenación. Consulte también https://msdn.microsoft.com/es-es/library/system.icomparable.compareto(v=vs.110).aspx
	`override bool Equals(object obj)`	Devuelve **true** si el objeto especificado es igual que el objeto actual. De lo contrario devuelve **false**.
	`System.DateTime GetDateTime(bool correctTZ)`	Devuelve un objeto `System.DateTime` de la instancia `Altova.Types.DateTime` actual. El argumento booleano `correctTZ` especifica si la hora del objeto devuelto debe ajustarse a la zona horaria de la instancia `Altova.Types.DateTime` actual.
	`override int GetHashCode()`	Devuelve el código hash de la instancia actual.
	`int GetWeekOfMonth()`	Devuelve el número de la semana del mes como entero.

	`static DateTime Parse( string s )`	Crea un objeto `DateTime` a partir de la cadena dada como argumento. Por ejemplo, estos valores de cadena de muestra se convertirían correctamente en un objeto `DateTime`:
		
		`2015-01-01T23:23:23`
		`2015-01-01`
		`2015-11`
		`23:23:23`
		
		Si la cadena no se puede convertir en un objeto `DateTime`, entonces se genera una excepción.
		
		Tenga en cuenta que este método es estático y solo se le puede llamar en la clase `Altova.Types.DateTime` propiamente dicha y no en una instancia de la clase.

	`static DateTime Parse(string s, DateTimeFormat format)`	Crea un objeto `DateTime` a partir de una cadena usando el formato dado como argumento. Para ver una lista de formatos posibles consulte Altova.Types.DateTimeFormat 1207.

	Nombre	Descripción
		Si la cadena no se puede convertir en un objeto `DateTime`, entonces se genera una excepción. Tenga en cuenta que este método es estático y solo se le puede llamar en la clase `Altova.Types.DateTime` propiamente dicha y no en una instancia de la clase.
	`override string ToString()`	Convierte el objeto `DateTime` en una cadena.
	`string ToString(DateTimeFormat format)`	Convierte el objeto `DateTime` en una cadena usando el formato dado como argumento. Para ver una lista de formatos posibles consulte Altova.Types.DateTimeFormat 1207.

Operadores

Nombre	Descripción
`!=`	Determina si `DateTime` a no es igual que `DateTime` b.
`<`	Determina si `DateTime` a es menor que `DateTime` b.
`<=`	Determina si `DateTime` a es menor o igual que `DateTime` b.
`==`	Determina si `DateTime` a es igual que `DateTime` b.
`>`	Determina si `DateTime` a es mayor que `DateTime` b.
`>=`	Determina si `DateTime` a es mayor o igual que `DateTime` b.

Ejemplos

Antes de usar estos ejemplos en su programa debe importar los tipos de Altova:

```
using Altova.Types;
```

Este ejemplo muestra cómo se pueden crear objetos `DateTime`:

```
protected static void DateTimeExample1()
{
   // Crear un objeto DateTime a partir de la hora actual del sistema
   Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);
   Console.WriteLine("La hora actual es: " + dt.ToString());

   // Crear un objeto Altova DateTime a partir de las partes (sin zona horaria)
```

 © 2016-2022 Altova GmbH

```
   Altova.Types.DateTime dt1 = new Altova.Types.DateTime(2015, 10, 12, 10, 50, 33);
   Console.WriteLine("Mi hora es : " + dt1.ToString());

   // Crear un objeto Altova DateTime a partir de las partes (con zona horaria UTC+60
minutos)
   Altova.Types.DateTime dt2 = new Altova.Types.DateTime(2015, 10, 12, 10, 50, 33, 60);
   Console.WriteLine("Mi hora con zona horaria es : " + dt2.ToString());

   // Crear un objeto Altova DateTime analizando una cadena
   Altova.Types.DateTime dt3 = Altova.Types.DateTime.Parse("2015-01-01T23:23:23");
   Console.WriteLine("La hora se creó a partir de la cadena: " + dt3.ToString());

   // Crear un objeto Altova DateTime analizando una cadena con formato de fecha de
esquema
   Altova.Types.DateTime dt4 = Altova.Types.DateTime.Parse("2015-01-01",
DateTimeFormat.W3_date);
   Console.WriteLine("La hora se creó a partir de una cadena con formato de fecha de
esquema: " + dt4.ToString());
}
```

Este ejemplo muestra cómo se puede dar formato a objetos `DateTime`:

```
protected static void DateTimeExample2()
{
   // Crear un objeto DateTime a partir de la hora actual del sistema
   Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);

   // Generar la fechaHora sin formato
   Console.WriteLine("Unformatted time: " + dt.ToString());

   // Generar este DateTime con formato usando lo siguiente
   Console.WriteLine("S_DateTime:       " + dt.ToString(DateTimeFormat.S_DateTime));
   Console.WriteLine("S_Days:           " + dt.ToString(DateTimeFormat.S_Days));
   Console.WriteLine("S_Seconds:        " + dt.ToString(DateTimeFormat.S_Seconds));
   Console.WriteLine("W3_date:          " + dt.ToString(DateTimeFormat.W3_date));
   Console.WriteLine("W3_dateTime:      " + dt.ToString(DateTimeFormat.W3_dateTime));
   Console.WriteLine("W3_gDay:          " + dt.ToString(DateTimeFormat.W3_gDay));
   Console.WriteLine("W3_gMonth:        " + dt.ToString(DateTimeFormat.W3_gMonth));
   Console.WriteLine("W3_gMonthDay:     " + dt.ToString(DateTimeFormat.W3_gMonthDay));
   Console.WriteLine("W3_gYear:         " + dt.ToString(DateTimeFormat.W3_gYear));
   Console.WriteLine("W3_gYearMonth:    " + dt.ToString(DateTimeFormat.W3_gYearMonth));
   Console.WriteLine("W3_time:          " + dt.ToString(DateTimeFormat.W3_time));
}
```

27.4.2 Altova.Types.DateTimeFormat

El tipo de enumeración `DateTimeFormat` tiene estos valores de constante:

Valor	Descripción	Ejemplo
S_DateTime	Da al valor el formato de un dateTime estándar, con una precisión de una décimo millonésima de segundo, incluida la zona horaria.	`2015-11-12 12:19:03.9019132+01:00`
S_Days	Da al valor el formato del número de días que han pasado desde la época UNIX.	`735913.631897345108796296 2963`
S_Seconds	Da al valor el formato del número de segundos que han pasado desde la épica UNIX, con una precisión de una décimo millonésima de segundo.	`63582937678.0769062`
W3_date	Da al valor el formato de fecha de esquema.	`2015-11-12`
W3_dateTime	Da al valor el formato de fechaHora de esquema.	`2015-11-12T15:12:14.5194251`
W3_gDay	Da al valor el formato de gDay de esquema.	`---12` (si la fecha es el día 12 del mes)
W3_gMonth	Da al valor el formato de gMonth de esquema.	`--11` (si el mes es noviembre)
W3_gMonthDay	Da al valor el formato de gMonthDay de esquema.	`--11-12` (si la fecha es el día 12 de noviembre)
W3_gYear	Da al valor el formato de gYear de esquema.	`2015` (si el año es 2015)
W3_gYearMonth	Da al valor el formato de gYearMonth de esquema.	`2015-11` (si el años es 2015 y el mes es noviembre)
W3_time	Da al valor el formato de hora de esquema, con una precisión de una décimo millonésima de segundo.	`15:19:07.5582719`

© 2016-2022 Altova GmbH

27.4.3 Altova.Types.Duration

Esta clase permite procesar atributos o elementos XML de tipo `xs:duration`.

Constructores

	Nombre	Descripción
	`Duration(Duration obj)`	Inicializa una instancia nueva de la clase `Duration` en el objeto `Duration` dado como argumento.
	`Duration(System.TimeSpan newvalue)`	Inicializa una instancia nueva de la clase `Duration` en el objeto `System.TimeSpan` dado como argumento.
	`Duration(long ticks)`	Inicializa una instancia nueva de la clase `Duration` en el número de ciclos dado como argumento.
	`Duration(int newyears, int newmonths, int days, int hours, int minutes, int seconds, double partseconds, bool bnegative)`	Inicializa una instancia nueva de la clase `Duration` en una duración compuesta por las partes dadas como argumentos.

Propiedades

	Nombre	Descripción
	`int Months`	Obtiene o establece el número de meses de la instancia actual de `Duration`.
	`System.TimeSpan Value`	Obtiene o establece el valor (como `System.TimeSpan`) de la instancia actual de `Duration`.
	`int Years`	Obtiene o establece el número de años de la instancia actual de `Duration`.

Métodos

	Nombre	Descripción
	`override bool Equals(object other)`	Devuelve **true** si el objeto especificado es igual al objeto actual. De lo contrario devuelve **false**.
	`override int GetHashCode()`	Devuelve el código hash de la instancia actual.

	Nombre	Descripción
	`bool IsNegative()`	Devuelve **true** si la instancia actual de `Duration` representa una duración negativa.
	`static Duration Parse( string s, ParseType pt )`	Devuelve un objeto `Altova.Types.Duration` analizado desde una cadena dada como argumento, usando el tipo de análisis dado como argumento. Son valores válidos de tipo de análisis: **DURATION** Analizar duración asumiendo que existen las partes año, mes, día y duración. **YEARMONTH** Analizar duración asumiendo que solo existen las partes año y mes. **DAYTIME** Analizar duración asumiendo que solo existen las partes día y hora. Recuerde que este método es estático solo se le puede llamar en la clase propiamente dicha y no desde una instancia de la clase.
	`override string ToString()`	Convierte la instancia `Duration` actual en una cadena. Por ejemplo, un intervalo de tiempo de 3 horas, 4 minutos y 5 segundos se convertiría en la cadena "PT3H4M5S".
	`string ToYearMonthString()`	Convierte la instancia `Duration` actual en una cadena usando el tipo de análisis YEARMONTH.

Operadores

Nombre	Descripción
`!=`	Determina si `Duration` a no es igual que `Duration` b.
`==`	Determina si `Duration` a es igual que `Duration` b.

Ejemplos

Antes de usar estos ejemplos en su programa debe importar los tipos de Altova:

```
using Altova.Types;
```

Este ejemplo muestra cómo se pueden crear objetos `Duration`:

```
protected static void DurationExample1()
{
   // Crear un intervalo de tiempo nuevo de 3 horas, 4 minutos y 5 segundos
   System.TimeSpan ts = new TimeSpan(3, 4, 5);
```


© 2016-2022 Altova GmbH

```
   // Crear un Duration a partir del intervalo de tiempo
   Duration dr = new Duration(ts);
   // El resultado es: PT3H4M5S
   Console.WriteLine("Duration created from TimeSpan: " + dr.ToString());

   // Crear un Altova.Types.Duration negativo a partir de 6 años, 5 meses, 4 días, 3
horas, 2 minutos, 1 segundo y .33 de un segundo
   Duration dr1 = new Duration(6, 5, 4, 3, 2, 1, .33, true);
   // El resultado es: -P6Y5M4DT3H2M1.33S
   Console.WriteLine("Duration created from parts: " + dr1.ToString());

   // Crear un Duration a partir de una cadena usando el tipo de análisis DAYTIME
   Duration dr2 = Altova.Types.Duration.Parse("-P4DT3H2M1S", Duration.ParseType.DAYTIME);
   // El resultado es: -P4DT3H2M1S
   Console.WriteLine("Duration created from string: " + dr2.ToString());

   // Crear una duración a partir de ciclos
   Duration dr3 = new Duration(System.DateTime.UtcNow.Ticks);
   // Generar el resultado
   Console.WriteLine("Duration created from ticks: " + dr3.ToString());
}
```

Este ejemplo muestra cómo se pueden obtener valores de los objetos `Duration`:

```
protected static void DurationExample2()
{
   // Crear un Altova.Types.Duration negativo a partir de 6 años, 5 meses, 4 días, 3
horas, 2 minutos, 1 segundo y .33 de un segundo
   Duration dr = new Duration(6, 5, 4, 3, 2, 1, .33, true);
   // El resultado es: -P6Y5M4DT3H2M1.33S
   Console.WriteLine("The complete duration is: " + dr.ToString());

   // Obtener solo la parte año y mes como cadena
   string dr1 = dr.ToYearMonthString();
   Console.WriteLine("The YEARMONTH part is: " + dr1);

   // Obtener el número de años de la duración
   Console.WriteLine("Years: " + dr.Years);

   // Obtener el número de meses de la duración
   Console.WriteLine("Months: " + dr.Months);
}
```

27.4.4 Altova.Xml.Meta.Attribute

Esta clase permite acceder a la información del esquema sobre clases generadas a partir de atributos. Recuerde que esta clase no tiene como objetivo aportar información dinámica sobre las distintas instancias de un atributo de un documento XML. Por el contrario, permite obtener información mediante programación sobre cualquier atributo que esté definido en el esquema XML.

Propiedades

	Nombre	Descripción
	`SimpleType DataType`	Devuelve el tipo del contenido del atributo.
	`string LocalName`	Devuelve el nombre local del atributo.
	`string NamespaceURI`	Devuelve el URI de espacio de nombres del atributo.
	`XmlQualifiedName QualifiedName`	Devuelve el nombre completo del atributo.
	`bool Required()`	Devuelve true si el atributo es obligatorio.

27.4.5 Altova.Xml.Meta.ComplexType

Esta clase permite acceder a información del esquema sobre clases generadas a partir de tipos complejos. Recuerde que esta clase no tiene como objetivo aportar información dinámica sobre las distintas instancias de un tipo complejo de un documento XML. Por el contrario, permite obtener información mediante programación sobre cualquier tipo complejo que esté definido en el esquema XML.

Propiedades

	Nombre	Descripción
	`Attribute[] Attributes`	Devuelve una lista de todos los atributos.
	`ComplexType BaseType`	Devuelve el tipo base de este tipo o null si no existe el tipo base.
	`SimpleType ContentType`	Devuelve el tipo simple del contenido.
	`Element[] Elements`	Devuelve una lista de todos los elementos.
	`string LocalName`	Devuelve el nombre local de este tipo.
	`string NamespaceURI`	Devuelve el URI de espacio de nombres del tipo.
	`XmlQualifiedName QualifiedName`	Devuelve el nombre completo de este tipo.

Métodos

	Nombre	Descripción
	`ComplexType BaseType`	Devuelve el tipo base de este tipo.

© 2016-2022 Altova GmbH

	Nombre	Descripción
	`bool Equals(obj)`	Comprueba si dos objetos de información remiten al mismo tipo, basándose en la comparación de los nombres completos. Devuelve true si el tipo tiene el mismo nombre completo.
	`Attribute FindAttribute(string localName, string namespaceURI)`	Busca el atributo con el nombre local y el URI de espacio de nombres especificados.
	`Element FindElement(string localName, string namespaceURI)`	Busca el elemento con el nombre local y el URI de espacio de nombres especificados.

27.4.6 Altova.Xml.Meta.Element

Esta clase permite acceder a la información del esquema sobre clases generadas a partir de elementos. Recuerde que esta clase no tiene como objetivo aportar información dinámica sobre las distintas instancias de un elemento de un documento XML. Por el contrario, permite obtener información mediante programación sobre cualquier elemento que esté definido en el esquema XML.

Propiedades

	Nombre	Descripción
	`ComplexType DataType`	Devuelve el tipo del elemento. Recuerde que este siempre es un tipo complejo incluso si se declaró como simple en el esquema original. Use `ContentType` del objeto devuelto para obtener el tipo de contenido simple.
	`string` `LocalName`	Devuelve el nombre local del elemento.
	`int` `MaxOccurs`	Devuelve el valor `maxOccurs` definido en el esquema.
	`int` `MinOccurs`	Devuelve el valor `minOccurs` definido en el esquema.
	`string` `NamespaceURI`	Devuelve el URI de espacio de nombres del elemento.
	`XmlQualifiedName QualifiedName`	Devuelve el nombre completo del elemento.

27.4.7 Altova.Xml.Meta.SimpleType

Esta clase permite acceder a la información del esquema sobre clases generadas a partir de tipos simples. Recuerde que esta clase no tiene como objetivo aportar información dinámica sobre las distintas instancias de un tipo simple de un documento XML. Por el contrario, permite obtener información mediante programación

sobre cualquier tipo simple que esté definido en el esquema XML.

Propiedades

	Nombre	Descripción
	`SimpleType BaseType`	Devuelve el tipo base de este tipo.
	`string``[] Enumerations`	Devuelve una lista de todas las facetas de enumeración.
	`int` `FractionDigits`	Devuelve el valor de esta faceta.
	`int` `Length`	Devuelve el valor de esta faceta.
	`string` `LocalName`	Devuelve el nombre local de este tipo.
	`string` `MaxExclusive`	Devuelve el valor de esta faceta.
	`string` `MaxInclusive`	Devuelve el valor de esta faceta.
	`int` `MaxLength`	Devuelve el valor de esta faceta.
	`string` `MinExclusive`	Devuelve el valor de esta faceta.
	`string` `MinInclusive`	Devuelve el valor de esta faceta.
	`int` `MinLength`	Devuelve el valor de esta faceta.
	`string` `NamespaceURI`	Devuelve el URI de espacio de nombres del tipo.
	`string``[] Patterns`	Devuelve las facetas de patrón o null si no hay patrones especificados.
	`XmlQualifiedName QualifiedName`	Devuelve el nombre completo de este tipo.
	`int` `TotalDigits`	Devuelve el valor de esta faceta.
	`WhitespaceType Whitespace`	Devuelve la faceta de normalización de espacios en blanco.

27.4.8 [SuEsquema].[Doc]

Cuando se genera código a partir de un esquema XML, el código generado ofrece una clase de documento que se llama igual que el esquema. Esta clase tiene como miembros todos los elementos raíz posibles, así como los métodos que aparecen a continuación. Observe que en los nombres de método, `CDoc` corresponde al nombre de la clase de documento generada.

© 2016-2022 Altova GmbH

Métodos

	Nombre	Descripción
	`static Doc CreateDocument()`	Crea un documento XML nuevo vacío.
	`static Doc CreateDocument(string encoding)`	Crea un documento XML nuevo vacío, con codificación de tipo "encoding"
	`static void DeclareAllNamespacesFromSchema(Altova.Xml.TypeBase node)`	Declara todos los espacios de nombres del esquema XML en el elemento dado como argumento (que suele ser el elemento XML raíz). Es útil llamar a este método si su esquema tiene varias declaraciones de espacios de nombres, cada una de ellas está asignada a un prefijo y quiere declararlas todas en el elemento dado como argumento.
	`static Doc LoadFromBinary(byte[] binary)`	Carga un documento XML desde una matriz de bytes.
	`static Doc LoadFromFile(string filename)`	Carga un documento XML desde un archivo.
	`static Doc LoadFromString(string xmlstring)`	Carga un documento XML desde una cadena de texto.
	`byte[] SaveToBinary(bool prettyPrint)`	Guarda un documento XML en una matriz de bytes, con formato pretty-print opcional.
	`byte[] SaveToBinary(bool prettyPrint, string encoding)`	Guarda un documento XML en una matriz de bytes (con formato pretty-print opcional) con la codificación especificada.
	`byte[] SaveToBinary(bool prettyPrint, string encoding, bool bBigEndian, bool bBOM)`	Guarda un documento XML en una matriz de bytes, (con formato pretty-print opcional) con la codificación, el orden de bytes y la marca BOM especificados.
	`void SaveToFile(string fileName, bool prettyPrint)`	Guarda un documento XML en un archivo, con formato pretty-print opcional.
	`void SaveToFile(string fileName, bool prettyPrint, bool omitXmlDecl)`	Guarda un documento XML en un archivo, con formato pretty-print opcional. Si el argumento `omitXmlDecl` tiene el valor true, la declaración XML no se escribirá.
	`void SaveToFile(string fileName, bool prettyPrint, bool omitXmlDecl, string encoding)`	Guarda un documento XML en un archivo, (con formato pretty-print opcional) con la codificación especificada. Si el argumento `omitXmlDecl` tiene el valor true, la declaración XML no se escribirá.
	`void SaveToFile(string fileName, bool prettyPrint,`	Guarda un documento XML en un archivo, (con formato pretty-print opcional) con la codificación y los caracteres de final de línea especificados.

	Nombre	Descripción
	string encoding, **string** lineend)	
	void SaveToFile(**string** fileName, **bool** prettyPrint, **bool** omitXmlDecl, **string** encoding, **string** lineend)	Guarda un documento XML en un archivo, (con formato pretty-print opcional) con la codificación y los caracteres de final de línea especificados. Si el argumento omitXmlDecl tiene el valor true, la declaración XML no se escribirá.
	void SaveToFile(**string** fileName, **bool** prettyPrint, **bool** omitXmlDecl, **string** encoding, **bool** bBigEndian, **bool** bBOM, **string** lineend)	Guarda un documento XML en un archivo, (con formato pretty-print opcional) con la codificación, el orden de bytes, la marca BOM y los caracteres de final de línea especificados. Si el argumento omitXmlDecl tiene el valor true, la declaración XML no se escribirá.
	void SaveToFileWithLineEnd(**string** fileName, **bool** prettyPrint, **bool** omitXmlDecl, **string** lineend)	Guarda un documento XML en un archivo, con formato pretty-print opcional y con caracteres de final de línea. Si el argumento omitXmlDecl tiene el valor true, la declaración XML no se escribirá.
	string SaveToString(**bool** prettyPrint)	Guarda un documento XML en un archivo, con formato pretty-print opcional.
	string SaveToString(**bool** prettyPrint, **bool** omitXmlDecl)	Guarda un documento XML en un archivo, con formato pretty-print opcional. Si el argumento omitXmlDecl tiene el valor true, la declaración XML no se escribirá.
	void SetDTDLocation(**string** dtdLocation)	Agrega una declaración DOCTYPE con el ID de sistema especificado. Es obligatorio que ya exista un elemento raíz.
	void SetSchemaLocation(**string** schemaLocation)	Agrega un atributo xsi:schemaLocation o xsi:noNamespaceSchemaLocation al elemento raíz. Es obligatorio que ya exista un elemento raíz.

27.4.9 [SuEsquema].[ElementType]

Esta clase permite manipular elementos XML desde el esquema. Los métodos de esta clase se pueden llamar en elementos, pero no en el documento XML. Observe que no necesita instanciar la clase directamente para poder llamar a los métodos de esta clase. Cualquier elemento creado con los métodos **append()** o **appendWithPrefix()** es de tipo **append()** or **appendWithPrefix()**.

Métodos

	Nombre	Descripción
	void DeclareNamespace(string prefix, string nsURI)	Este método toma dos argumentos de tipo cadena: el prefijo y el URI de espacio de nombres que quiere usar. El prefijo dado como argumento se asigna al valor del URI de espacio de nombres dado como argumento. Si el prefijo dado como

© 2016-2022 Altova GmbH

	Nombre	Descripción

		argumento está vacío, el método crea o sobrescribe la declaración de espacio de nombres predeterminada del elemento.
		
		Por ejemplo, imagine que el documento XML tiene un elemento XML llamado "purchase". Si llama a
		
		`purchase.DeclareNamespace("ord", "http://OrderTypes");`
		
		entonces el documento XML se convierte en
		
		`<purchase xmlns:ord="http://OrderTypes" />`
		
		Otro ejemplo. Si llama a:
		
		`purchase.DeclareNamespace("", "http://OrderTypes");`
		
		entonces el documento XML se convierte en
		
		`<purchase xmlns="http://OrderTypes" />`
		
		Nota: el espacio de nombres declarado se usa al anexar elementos secundarios o atributos posteriormente, conforme a estas normas:
		
		1. Si el espacio de nombres secundario es el predeterminado, use un prefijo vacío.
		2. Si el espacio de nombres secundario es igual que el del elemento matriz, use el prefijo matriz.
		3. De lo contrario, busque el prefijo más cercano desde el elemento matriz hasta el principio usando el algoritmo de búsqueda que se describe en la sección "B.2: Namespace Prefix Lookup" de https://www.w3.org/TR/2002/WD-DOM-Level-3-Core-20021022/namespaces-algorithms.html.
		4. Si no se encuentra ningún prefijo para el espacio de nombres del elemento, entonces use un prefijo vacío.

27.4.10 [SuTipoEsquema].MemberAttribute

Cuando se genera código a partir de un esquema XML, se crea una clase como esta por cada atributo miembro de un tipo. En las tablas que aparecen a continuación `AttributeType` corresponde al tipo del atributo miembro propiamente dicho.

Métodos

	Nombre	Descripción
	`bool Exists()`	Devuelve true si el atributo existe.
	`void Remove()`	Quita el atributo de su elemento primario.

Propiedades

	Nombre	Descripción
	`int` `EnumerationValue`	Solo se genera para tipos de enumeración. Establece u obtiene el valor del atributo con ayuda de una de las constantes generadas para los posibles valores. Devuelve "no válido" si el valor no coincide con ninguno de los valores enumerados en el esquema.
	`Altova.Xml.Meta.Attribute Info`	Devuelve un objeto para consultar información del esquema (*véase* *`Altova.Xml.Meta.Attribute`* 1211).
	`AttributeType Value`	Establece u obtiene el valor del atributo.

27.4.11 [SuTipoEsquema].MemberElement

Cuando se genera código a partir de un esquema XML, se crea una clase como esta por cada elemento miembro de un tipo. La clase implementa la interfaz estándar `System.Collections.IEnumerable` para que pueda utilizarse con la instrucción `foreach`.

En la tabla que aparece a continuación `MemberType` corresponde al tipo del elemento miembro propiamente dicho.

Métodos

	Nombre	Descripción
	`MemberType Append()`	Crea un elemento nuevo y lo anexa a su primario.

© 2016-2022 Altova GmbH

	Nombre	Descripción
	`MemberType AppendWithPrefix(string prefix)`	Crea un elemento nuevo que tiene el prefijo dado como argumento y lo anexa a su elemento superior. Para ver un ejemplo consulte Ejemplo: orden de compra[1179].
	`MemberType At(int index)`	Devuelve el elemento miembro especificado por el índice.
	`System.Collections.IEnumerator GetEnumerator()`	Devuelve un objeto para recorrer instancias del elemento miembro.
	`void Remove()`	Elimina todas las instancias del elemento en su primario.
	`void RemoveAt(int index)`	Elimina la instancia del elemento especificada por el índice.

Propiedades

	Nombre	Descripción
	`int Count`	Devuelve el recuento de elementos.
	`int EnumerationValue`	Solo se genera para tipos de enumeración. Establece u obtiene el valor del elemento con ayuda de una de las constantes generadas para los posibles valores. Devuelve "no válido" si el valor no coincide con ninguno de los valores enumerados en el esquema.
	`bool Exists`	Devuelve true si existe al menos un elemento.
	`MemberType First`	Devuelve la primera instancia del elemento miembro.
	`Altova.Xml.Meta.Element Info`	Devuelve un objeto para consultar información del esquema (véase `Altova.Xml.Meta.Element`[1213]).
	`MemberType Last`	Devuelve la última instancia del elemento miembro.
	`MemberType this[int index]`	Devuelve el elemento miembro especificado por el índice.
	`MemberType Value`	Establece u obtiene el contenido del elemento (solo se genera si el elemento puede tener contenido mixto o simple).

27.5 Referencia de las clases generadas (Java)

Esta sección describe las clases Java que se generan con XMLSpy a partir de una DTD o esquema XML (*véase* *Generar código a partir de esquemas XML o DTD* 1142). Puede integrar estas clases en su código para leer, modificar y escribir documentos XML.

Nota: el código generado incluye otras clases compatibles que no se enumeran en este apartado y que pueden cambiar en el futuro.

© 2016-2022 Altova GmbH

27.5.1 com.altova.types.DateTime

Esta clase permite procesar atributos o elementos XML que tienen tipos de fecha y hora (p. ej. `xs:dateTime`).

Constructores

	Nombre	Descripción
C	`public DateTime()`	Inicializa una instancia nueva de la clase `DateTime` en un valor vacío.
C	`public DateTime(DateTime newvalue)`	Inicializa una instancia nueva de la clase `DateTime` en el valor `DateTime` dado como argumento.
C	`public DateTime(int newyear, int newmonth, int newday, int newhour, int newminute, int newsecond, double newpartsecond, int newoffsetTZ)`	Inicializa una instancia nueva de la clase `DateTime` en el año, mes, día, hora, minuto, segundo, la parte fraccionaria del segundo y la zona horaria dados como argumentos. La parte fraccionaria del segundo newpartsecond debe estar comprendida entre 0 y 1. La diferencia de zona horaria newoffsetTZ puede ser negativa o positivo y se expresa en minutos.
C	`public DateTime(int newyear, int newmonth, int newday, int newhour, int newminute, int newsecond, double newpartsecond)`	Inicializa una instancia nueva de la clase `DateTime` en el año, mes, día, hora, minuto, segundo y la parte fraccionaria de un segundo dados como argumentos.
C	`public DateTime(int newyear, int newmonth, int newday)`	Inicializa una instancia nueva de la clase `DateTime` en el año, mes y día dados como argumentos.
C	`public DateTime(Calendar newvalue)`	Inicializa una instancia nueva de la clase `DateTime` en el valor `java.util.Calendar` dado como argumento.

Métodos

	Nombre	Descripción
S	`static DateTime now()`	Devuelve la hora actual como un objeto `DateTime`.

S	`static DateTime parse(String s)`	Devuelve un objeto `DateTime` analizado desde el valor de cadena dado como argumento. Por ejemplo, estos valores de cadena de muestra se convertirían correctamente en un objeto `DateTime`:
		
		`2015-11-24T12:54:47.969+01:00`
		`2015-11-24T12:54:47`
		`2015-11-24`

	Nombre	Descripción
	int getDay()	Devuelve el día de la instancia DateTime actual.
	int getHour()	Devuelve la hora de la instancia DateTime actual.
	int getMillisecond()	Devuelve el milisegundo de la instancia DateTime actual, como valor entero.
	int getMinute()	Devuelve el minuto de la instancia DateTime actual.
	int getMonth()	Devuelve el mes de la instancia DateTime actual.
	double getPartSecond()	Devuelve la parte fraccionaria del segundo de la instancia DateTime actual, como valor **double**. El valor devuelto es mayor que cero y menor que uno. Por ejemplo: 0.313
	int getSecond()	Devuelve el segundo de la instancia DateTime actual.
	int getTimezoneOffset()	Devuelve la diferencia de zona horaria, en minutos, de la instancia DateTime actual. Por ejemplo, la zona horaria "UTC-01:00" se devolvería como: -60
	Calendar getValue()	Devuelve la instancia DateTime actual como un valor java.util.Calendar.
	int getWeekday()	Devuelve el día de la semana de la instancia DateTime actual. Los valores están comprendidos entre el 0 y el 6, siendo 0 el lunes (ISO-8601).
	int getYear()	Devuelve el año de la instancia DateTime actual.
	int hasTimezone()	Devuelve información sobre la zona horaria de la instancia DateTime actual. Los valores devueltos posibles son: CalendarBase.TZ_MISSING — No está definida la diferencia de zona horaria. CalendarBase.TZ_UTC — La zona horaria es UTC. CalendarBase.TZ_OFFSET — Está definida una diferencia de zona horaria.
	void setDay(**int** nDay)	Establece el día de la instancia DateTime actual en el valor dado como argumento.
	void setHasTimezone(**int** nHasTZ)	Establece la información de zona horaria de la instancia DateTime actual en el valor dado como argumento. Este método puede utilizarse para eliminar la información de zona horaria o establecer la zona horaria en UTC (hora

© 2016-2022 Altova GmbH

	Nombre	Descripción
		universal coordinada). Los valores válidos para el argumento nHasTZ son: `CalendarBase.TZ_MISSING` Establecer la diferencia de zona horaria como indefinida. `CalendarBase.TZ_UTC` Establecer la zona horaria en UTC. `CalendarBase.TZ_OFFSET` Si el objeto actual tiene una diferencia de zona horaria, dejarla sin cambiar.
	void setHour(**int** nHour)	Establece la hora de la instancia `DateTime` actual en el valor dado como argumento.
	void setMinute(**int** nMinute)	Establece el minuto de la instancia `DateTime` actual en el valor dado como argumento.
	void setMonth(**int** nMonth)	Establece el mes de la instancia `DateTime` actual en el valor dado como argumento.
	void setPartSecond(**double** nPartSecond)	Establece la parte fraccionaria del segundo de la instancia `DateTime` actual en el valor dado como argumento.
	void setSecond(**int** nSecond)	Establece el segundo de la instancia `DateTime` actual en el valor dado como argumento.
	void setTimezoneOffset(**int** nOffsetTZ)	Establece la diferencia de zona horaria de la instancia `DateTime` actual en el valor dado como argumento. El valor nOffsetTZ debe ser un entero (negativo o positivo) y debe expresarse en minutos.
	void setYear(**int** nYear)	Establece el año de la instancia `DateTime` actual en el valor dado como argumento.
	String toString()	Devuelve la representación de cadena de la instancia `DateTime` actual. Por ejemplo: `2015-11-24T15:50:56.968+01:00`

Ejemplos

Antes de usar este ejemplo en su programa debe importar los tipos de Altova:

```
import com.altova.types.*;
```

Este ejemplo muestra cómo se pueden crear objetos `DateTime`:

```
protected static void DateTimeExample1()
{
   // Inicializar una instancia nueva de la clase DateTime en la hora actual
   DateTime dt = new DateTime(DateTime.now());
   System.out.println("DateTime created from current date and time: " + dt.toString());

   // Inicializar una instancia nueva de la clase DateTime aportando las partes
   DateTime dt1 = new DateTime(2015, 11, 23, 14, 30, 24, .459);
   System.out.println("DateTime from parts (no timezone): " + dt1.toString());

   // Inicializar una instancia nueva de la clase DateTime aportando las partes
   DateTime dt2 = new DateTime(2015, 11, 24, 14, 30, 24, .459, -60);
   System.out.println("DateTime from parts (with negative timezone): " + dt2.toString());

   // Inicializar una instancia nueva de la clase DateTime analizando un valor de cadena
       DateTime dt3 = DateTime.parse("2015-11-24T12:54:47.969+01:00");
   System.out.println("DateTime parsed from string: " + dt3.toString());
}
```

Este ejemplo muestra cómo se pueden obtener valores de objetos `DateTime`:

```
protected static void DateTimeExample2()
   {
      // Inicializar una instancia nueva de la clase DateTime en la hora actual
      DateTime dt = new DateTime(DateTime.now());

      // Generar el año, mes y día de esta instancia DateTime con formato
      String str1 = String.format("Year: %d; Month: %d; Day: %d;", dt.getYear(),
dt.getMonth(), dt.getDay());
      System.out.println(str1);

      // Generar la hora, el minuto y el segundo de esta instancia DateTime con formato
      String str2 = String.format("Hour: %d; Minute: %d; Second: %d;", dt.getHour(),
dt.getMinute(), dt.getSecond());
      System.out.println(str2);

      // Devolver la zona horaria (en minutos) de esta instancia DateTime
      System.out.println("Timezone: " + dt.getTimezoneOffset());

      // Obtener DateTime como valor java.util.Calendar
      java.util.Calendar dt_java = dt.getValue();
      System.out.println("" + dt_java.toString());

      // Devolver el día de la semana de esta instancia DateTime
      System.out.println("Weekday: " + dt.getWeekday());

      // Comprobar si la instancia DateTime tiene definida una zona horaria
      switch(dt.hasTimezone())
      {
         case CalendarBase.TZ_MISSING:
            System.out.println("Sin zona horaria.");
            break;
         case CalendarBase.TZ_UTC:
```


© 2016-2022 Altova GmbH

```
            System.out.println("La zona horaria es UTC.");
            break;
         case CalendarBase.TZ_OFFSET:
            System.out.println("Este objeto tiene una zona horaria.");
            break;
         default:
            System.out.println("Imposible determinar si se definió una zona horaria.");
            break;
      }
   }
```

Este ejemplo muestra cómo se puede cambiar la diferencia horaria de un objeto `DateTime`:

```
protected static void DateTimeExample3()
{
   // Crear un objeto DateTime nuevo con zona horaria -0100 UTC
   DateTime dt = new DateTime(2015, 11, 24, 14, 30, 24, .459, -60);
   // Generar el valor antes del cambio
   System.out.println("Before: " + dt.toString());
   // Cambiar la diferencia a +0100 UTC
   dt.setTimezoneOffset(60);
   // Generar el valor después del cambio
   System.out.println("After:  " + dt.toString());
}
```

27.5.2 com.altova.types.Duration

Esta clase permite procesar atributos o elementos XML de tipo `xs:duration`.

Constructores

	Nombre	Descripción
C	`Duration(Duration newvalue)`	Inicializa una instancia nueva de la clase `Duration` en el objeto `Duration` dado como argumento.
C	`Duration(int newyear, int newmonth, int newday, int newhour, int newminute, int newsecond, double newpartsecond, boolean newisnegative)`	Inicializa una instancia nueva de la clase `Duration` en una duración construida con las partes dadas como argumentos.

Métodos

	Nombre	Descripción
S	`static Duration getFromDayTime( int newday, int newhour, int newminute, int newsecond, double newpartsecond )`	Devuelve un objeto `Duration` creado a partir del número de días, horas, minutos, segundos y partes fraccionarias de segundo dados como argumentos.
S	`static Duration getFromYearMonth( int newyear, int newmonth )`	Devuelve un objeto `Duration` creado a partir del número de años y meses dados como argumentos.
S	`static Duration parse( String s )`	Devuelve un objeto `Duration` creado a partir de la cadena dada como argumento. Por ejemplo, la cadena -P1Y1M1DT1H1M1.333S puede utilizarse para crear una duración negativa de un año, un mes, un día, una hora, un minuto, un segundo y 0.333 partes fraccionarias de un segundo. Para crear una duración negativa basta con anexar el signo menos (-) a la cadena.
S	`static Duration parse( String s, ParseType pt )`	Devuelve un objeto `Duration` creado a partir de la cadena dada como argumento, usando un formato de análisis concreto. El formato de análisis puede ser cualquiera de estos: **ParseType.DAYTIME** Puede utilizarse cuando la cadena **s** está compuesta por cualquiera de estos: días, horas, minutos,

© 2016-2022 Altova GmbH

	Nombre	Descripción
		segundos, partes fraccionarias de segundo. Por ejemplo - P4DT4H4M4.774S. **ParseType.DURATION** Puede utilizarse cuando la cadena **s** está compuesta por cualquiera de estos: años, meses, días, horas, minutos, segundos, partes fraccionarias de segundo. Por ejemplo P1Y1M1DT1H1M1.333S. **ParseType.YEARMONTH** Puede utilizarse cuando la cadena **s** está compuesta por cualquiera de estos: años, meses. Por ejemplo: P3Y2M.
	int `getDay()`	Devuelve el número de días de la instancia `Duration` actual.
	long `getDayTimeValue()`	Devuelve el valor de día y hora (en milisegundos) de la instancia `Duration` actual. Años y meses se omiten.
	int `getHour()`	Devuelve el número de horas de la instancia `Duration` actual.
	int `getMillisecond()`	Devuelve el número de milisegundos de la instancia `Duration` actual.
	int `getMinute()`	Devuelve el número de minutos de la instancia `Duration` actual.
	int `getMonth()`	Devuelve el número de meses de la instancia `Duration` actual.
	double `getPartSecond()`	Devuelve el partes fraccionarias de segundo de la instancia `Duration` actual.
	int `getSecond()`	Devuelve el número de segundos de la instancia `Duration` actual.
	int `getYear()`	Devuelve el número de años de la instancia `Duration` actual.
	int `getYearMonthValue()`	Devuelve el valor de año y mes (en meses) de la instancia `Duration` actual. Días, horas, segundos y milisegundos se omiten.
	boolean `isNegative()`	Devuelve el valor booleano **true** si la instancia `Duration` actual es negativa.
	void `setDayTimeValue(long l)`	Establece la duración en el número de milisegundos dados como argumento, afectando solo a la parte día y hora de la duración.

	Nombre	Descripción
	`void setNegative( boolean isnegative )`	Convierte la instancia `Duration` actual en una duración negativa.
	`void setYearMonthValue(int l)`	Establece la duración en el número de meses dados como argumento. Esto solo afecta la parte años y meses de la duración.
	`String toString()`	Devuelve la representación de cadena de la instancia `Duration`. Por ejemplo: `-P4DT4H4M4.774S`
	`String toYearMonthString()`	Devuelve la representación de cadena de la parte YearMonth de la instancia `Duration` actual. Por ejemplo: `P1Y2M`

Ejemplos

Antes de usar este ejemplo en su programa debe importar los tipos de Altova:

```
import com.altova.types.*;
import com.altova.types.Duration.ParseType;
```

Este ejemplo muestra cómo se pueden crear objetos `Duration`:

```
protected static void ExampleDuration()
{
   // Crear una duración negativa de 1 año, 1 mes, 1 día, 1 hora, 1 minuto, 1 segundo y
0.333 partes fraccionarias de segundo
   Duration dr = new Duration(1, 1, 1, 1, 1, 1, .333, true);

   // Crear una duración a partir de un objeto Duration que ya existe
   Duration dr1 = new Duration(dr);

   // Crear una duración de 4 días, 4 horas, 4 minutos, 4 segundos, .774 partes
fraccionarias de segundo
   Duration dr2 = Duration.getFromDayTime(4, 4, 4, 4, .774);

   // Crear una duración de 3 años y 2 meses
   Duration dr3 = Duration.getFromYearMonth(3, 2);

   // Crear una duración a partir de una cadena
   Duration dr4 = Duration.parse("-P4DT4H4M4.774S");

   // Crear una duración a partir de una cadena, usando formatos de análisis concretos
   Duration dr5 = Duration.parse("-P1Y1M1DT1H1M1.333S", ParseType.DURATION);
   Duration dr6 = Duration.parse("P3Y2M", ParseType.YEARMONTH);
```


© 2016-2022 Altova GmbH

```
   Duration dr7 = Duration.parse("-P4DT4H4M4.774S", ParseType.DAYTIME);
}
```

Este ejemplo muestra cómo se puede obtener y establecer el valor de objetos `Duration`:

```
protected static void DurationExample2()
{
   // Crear una duración de 1 año, 2 meses, 3 días, 4 horas, 5 minutos, 6 segundos y 333
milisegundos
   Duration dr = new Duration(1, 2, 3, 4, 5, 6, .333, false);
   // Generar el número de días de esta duración
   System.out.println(dr.getDay());

   // Crear una duración positiva de un año y 333 milisegundos
   Duration dr1 = new Duration(1, 0, 0, 0, 0, 0, .333, false);
   // Generar el valor de día y hora en milisegundos
   System.out.println(dr1.getDayTimeValue());

   // Crear una duración positiva de 1 año, 1 mes, 1 día, 1 hora, 1 minuto, 1 segundo y
333 milisegundos
   Duration dr2 = new Duration(1, 1, 1, 1, 1, 1, .333, false);
   // Generar el valor de año y mes en meses
   System.out.println(dr2.getYearMonthValue());

   // Crear una duración positiva de 1 año y 1 un mes
   Duration dr3 = new Duration(1, 1, 0, 0, 0, 0, 0, false);
   // Generar el valor
   System.out.println("The duration is now: " + dr3.toString());
   // Establecer la parte DayTime de la duración en 1000 milisegundos
   dr3.setDayTimeValue(1000);
   // Generar el valor
   System.out.println("The duration is now: " + dr3.toString());
   // Generar la parte de mes y año de la duración
   dr3.setYearMonthValue(1);
   // Generar el valor
   System.out.println("The duration is now: " + dr3.toString());
   // Generar la parte de mes y año de la duración
   System.out.println("The YearMonth part of the duration is: " +
dr3.toYearMonthString());
}
```

27.5.3 com.altova.xml.meta.Attribute

Esta clase permite acceder a la información del esquema sobre clases generadas a partir de atributos. Recuerde que esta clase no tiene como objetivo aportar información dinámica sobre las distintas instancias de un atributo de un documento XML. Por el contrario, permite obtener información mediante programación sobre cualquier atributo que esté definido en el esquema XML.

Métodos

	Nombre	Descripción
	`SimpleType getDataType()`	Devuelve el tipo del contenido del atributo.
	`String getLocalName()`	Devuelve el nombre local del atributo.
	`String getNamespaceURI()`	Devuelve el URI de espacio de nombres del atributo.
	`boolean isRequired()`	Devuelve true si el atributo es obligatorio.

27.5.4 com.altova.xml.meta.ComplexType

Esta clase permite acceder a información del esquema sobre clases generadas a partir de tipos complejos. Recuerde que esta clase no tiene como objetivo aportar información dinámica sobre las distintas instancias de un tipo complejo de un documento XML. Por el contrario, permite obtener información mediante programación sobre cualquier tipo complejo que esté definido en el esquema XML.

Métodos

	Nombre	Descripción
	`Attribute findAttribute(String localName, String namespaceURI)`	Busca el atributo con el nombre local y el URI de espacio de nombres especificados.
	`Element findElement(String localName, String namespaceURI)`	Busca el elemento con el nombre local y el URI de espacio de nombres especificados.
	`Attribute[] GetAttributes()`	Devuelve una lista de todos los atributos.
	`ComplexType getBaseType()`	Devuelve el tipo base de este tipo.
	`SimpleType getContentType()`	Devuelve el tipo simple del contenido.
	`Element[] GetElements()`	Devuelve una lista de todos los elementos.
	`String getLocalName()`	Devuelve el nombre local de este tipo.
	`String getNamespaceURI()`	Devuelve el URI de espacio de nombres del tipo.

© 2016-2022 Altova GmbH

27.5.5 com.altova.xml.meta.Element

Esta clase permite acceder a la información del esquema sobre clases generadas a partir de elementos. Recuerde que esta clase no tiene como objetivo aportar información dinámica sobre las distintas instancias de un elemento de un documento XML. Por el contrario, permite obtener información mediante programación sobre cualquier elemento que esté definido en el esquema XML.

Métodos

	Nombre	Descripción
	`ComplexType getDataType()`	Devuelve el tipo del elemento. Recuerde que este siempre es un tipo complejo incluso si se declaró como simple en el esquema original. Use `getContentType()` del objeto devuelto para obtener el tipo de contenido simple.
	`String getLocalName()`	Devuelve el nombre local del elemento.
	`int` `getMaxOccurs()`	Devuelve el valor maxOccurs definido en el esquema.
	`int` `getMinOccurs()`	Devuelve el valor minOccurs definido en el esquema.
	`String getNamespaceURI()`	Devuelve el URI de espacio de nombres del elemento.

27.5.6 com.altova.xml.meta.SimpleType

Esta clase permite acceder a la información del esquema sobre clases generadas a partir de tipos simples. Recuerde que esta clase no tiene como objetivo aportar información dinámica sobre las distintas instancias de un tipo simple de un documento XML. Por el contrario, permite obtener información mediante programación sobre cualquier tipo simple que esté definido en el esquema XML.

Métodos

	Nombre	Descripción
	`SimpleType getBaseType()`	Devuelve el tipo base de este tipo.
	`String[] getEnumerations()`	Devuelve una matriz de todas las facetas de enumeración.
	`int` `getFractionDigits()`	Devuelve el valor de esta faceta.
	`int` `getLength()`	Devuelve el valor de esta faceta.
	`String getLocalName()`	Devuelve el nombre local del tipo.
	`String getMaxExclusive()`	Devuelve el valor de esta faceta.
	`String getMaxInclusive()`	Devuelve el valor de esta faceta.

	Nombre	Descripción
	`int getMaxLength()`	Devuelve el valor de esta faceta.
	`String getMinExclusive()`	Devuelve el valor de esta faceta.
	`String getMinInclusive()`	Devuelve el valor de esta faceta.
	`int getMinLength()`	Devuelve el valor de esta faceta.
	`String getNamespaceURI()`	Devuelve el URI de espacio de nombres del tipo.
	`String[] getPatterns()`	Devuelve una matriz de todas las facetas de patrón.
	`int getTotalDigits()`	Devuelve el valor de esta faceta.
	`int getWhitespace()`	Devuelve el valor de la faceta de espacio en blanco, que es uno de estos: `com.altova.typeinfo.WhitespaceType.Whitespace_Unknown` `com.altova.typeinfo.WhitespaceType.Whitespace_Preserve` `com.altova.typeinfo.WhitespaceType.Whitespace_Replace` `com.altova.typeinfo.WhitespaceType.Whitespace_Collapse`

27.5.7 com.[SuEsquema].[Doc]

Cuando se genera código a partir de un esquema XML, el código generado ofrece una clase de documento que se llama igual que el esquema. Esta clase tiene como miembros todos los elementos raíz posibles, así como los miembros que aparecen a continuación. Observe que en los nombres de método, "Doc" corresponde al nombre de la clase de documento generada.

Métodos

	Nombre	Descripción
S	`static Doc createDocument()`	Crea un documento XML nuevo vacío.
S	`static void declareAllNamespacesFromSchema(com.altova.xml.TypeBase node)`	Declara todos los espacios de nombres del esquema XML en el elemento dado como argumento (que suele ser el elemento XML raíz). Es útil llamar a este método si su esquema tiene varias declaraciones de espacios de nombres, cada una de ellas está asignada a un prefijo y quiere declararlas todas en el elemento dado como argumento.
S	`static Doc loadFromBinary(byte[] xml)`	Carga un documento XML desde una matriz de bytes.

© 2016-2022 Altova GmbH

	Nombre	Descripción
S	**static** Doc loadFromFile(String fileName)	Carga un documento XML desde un archivo.
S	**static** Doc loadFromString(String xml)	Carga un documento XML desde una cadena de texto.
	byte[] saveToBinary(**boolean** prettyPrint)	Guarda un documento XML en una matriz de bytes, con formato pretty-print opcional.
	byte[] saveToBinary(**boolean** prettyPrint, String encoding)	Guarda un documento XML en una matriz de bytes (con formato pretty-print opcional) con la codificación especificada.
	byte[] saveToBinary(**boolean** prettyPrint, String encoding, **boolean** bigEndian, **boolean** writeBOM)	Guarda un documento XML en una matriz de bytes (con formato pretty-print opcional) con la codificación especificada. Para codificaciones Unicode se puede especificar el orden de bytes y la marca BOM Unicode.
	void saveToFile(String fileName, **boolean** prettyPrint)	Guarda un documento XML en un archivo, con formato pretty-print opcional.
	void saveToFile(String fileName, **boolean** prettyPrint, **boolean** omitXmlDecl)	Guarda un documento XML en un archivo, con formato pretty-print opcional, con codificación UTF-8. Si el argumento omitXmlDecl tiene el valor true, la declaración XML no se escribirá.
	void saveToFile(String fileName, **boolean** prettyPrint, **boolean** omitXmlDecl, String encoding)	Guarda un documento XML en un archivo (con formato pretty-print opcional) con la codificación especificada. Si el argumento omitXmlDecl tiene el valor true, la declaración XML no se escribirá.
	void saveToFile(String fileName, **boolean** prettyPrint, **boolean** omitXmlDecl, String encoding, **boolean** bBigEndian, **boolean** bBOM)	Guarda un documento XML en un archivo (con formato pretty-print opcional) con la codificación especificada. Si el argumento omitXmlDecl tiene el valor true, la declaración XML no se escribirá. Para codificaciones Unicode se puede especificar el orden de bytes y la marca BOM Unicode.
	void saveToFile(String fileName, **boolean** prettyPrint, String encoding)	Guarda un documento XML en un archivo, (con formato pretty-print opcional) con la codificación especificada.
	void saveToFile(String fileName, **boolean** prettyPrint, String encoding, **boolean** bBigEndian, **boolean** bBOM)	Guarda un documento XML en un archivo (con formato pretty-print opcional) con la codificación especificada. Para codificaciones Unicode se puede especificar el orden de bytes y la marca BOM Unicode.
	String saveToString(**boolean** prettyPrint)	Guarda un documento XML en una cadena de texto, con formato pretty-print opcional.
	String saveToString(**boolean** prettyPrint, **boolean**	Guarda un documento XML en una cadena de texto, con formato pretty-print opcional. Si el argumento omitXmlDecl tiene el valor

	Nombre	Descripción
	omitXmlDecl)	true, la declaración XML no se escribirá.
	void setSchemaLocation(String schemaLocation)	Agrega un atributo xsi:schemaLocation o xsi:noNamespaceSchemaLocation al elemento raíz. Es obligatorio que ya exista un elemento raíz.

27.5.8 com.[SuEsquema].[ElementType]

Esta clase permite manipular elementos XML desde el esquema. Los métodos de esta clase se pueden llamar en elementos, pero no en el documento XML. Observe que no necesita instanciar la clase directamente para poder llamar a los métodos de esta clase. Cualquier elemento creado con los métodos **append()** o **appendWithPrefix()** es de tipo **append()** or **appendWithPrefix()**.

Métodos

	Nombre	Descripción

	void declareNamespace(String prefix, String nsURI)	Este método toma dos argumentos de tipo cadena: el prefijo y el URI de espacio de nombres que quiere usar. El prefijo dado como argumento se asigna al valor del URI de espacio de nombres dado como argumento. Si el prefijo dado como argumento está vacío, el método crea o sobrescribe la declaración de espacio de nombres predeterminada del elemento.
		
		Por ejemplo, imagine que el documento XML tiene un elemento XML llamado "purchase". Si llama a
		
		`purchase.declareNamespace("ord", "http://OrderTypes");`
		
		entonces el documento XML se convierte en
		
		`<purchase xmlns:ord="http://OrderTypes" />`
		
		Otro ejemplo. Si llama a:
		
		`purchase.declareNamespace("", "http://OrderTypes");`
		
		entonces el documento XML se convierte en
		
		`<purchase xmlns="http://OrderTypes" />`

© 2016-2022 Altova GmbH

	Nombre	Descripción
		Nota: el espacio de nombres declarado se usa al anexar elementos secundarios o atributos posteriormente, conforme a estas normas: 1. Si el espacio de nombres secundario es el predeterminado, use un prefijo vacío. 2. Si el espacio de nombres secundario es igual que el del elemento matriz, use el prefijo matriz. 3. De lo contrario, busque el prefijo más cercano desde el elemento matriz hasta el principio usando el algoritmo de búsqueda que se describe en la sección "B.2: Namespace Prefix Lookup" de https://www.w3.org/TR/2002/WD-DOM-Level-3-Core-20021022/namespaces-algorithms.html. 4. Si no se encuentra ningún prefijo para el espacio de nombres del elemento, entonces use un prefijo vacío.

27.5.9 com.[SuEsquema].[SuTipoEsquema].MemberAttribute

Cuando se genera código a partir de un esquema XML, se crea una clase como esta por cada atributo miembro de un tipo. En la tabla que aparece a continuación `AttributeType` corresponde al tipo del atributo miembro propiamente dicho.

Métodos

	Nombre	Descripción
	`boolean exists()`	Devuelve true si el atributo existe.
	`int getEnumerationValue()`	Solo se genera para tipos de enumeración. Devuelve una de las constantes generadas para los posibles valores o "no válido" si el valor no coincide con ninguno de los valores enumerados en el esquema.
	`com.altova.xml.meta.Attribute getInfo()`	Devuelve un objeto para consultar información del esquema (véase `com.altova.xml.meta.Attribute` 1229).
	`AttributeType getValue()`	Obtiene el valor del atributo.
	`void remove()`	Quita el atributo de su elemento primario.
	`void setEnumerationValue(int)`	Solo se genera para tipos de enumeración. Pase una de las constantes generadas para los posibles

	Nombre	Descripción
		valores a este método para establecer el valor.
●	`void setValue(AttributeType value)`	Establece el valor del atributo.

27.5.10 com.[SuEsquema].[SuTipoEsquema].MemberElement

Cuando se genera código a partir de un esquema XML, se crea una clase como esta por cada elemento miembro de un tipo. En la tabla que aparece a continuación `MemberType` corresponde al tipo del elemento miembro propiamente dicho.

Métodos

	Nombre	Descripción
●	`MemberType append()`	Crea un elemento nuevo y lo anexa a su primario.
●	`MemberType appendWithPrefix(String prefix)`	Crea un elemento nuevo que tiene el prefijo dado como argumento y lo anexa a su elemento superior. Para ver un ejemplo consulte Ejemplo: orden de compra 1179.
●	`MemberType at(int index)`	Devuelve la instancia del elemento miembro en el índice especificado.
●	`int count()`	Devuelve el recuento de elementos.
●	`boolean exists()`	Devuelve true si existe al menos un elemento.
●	`MemberType first()`	Devuelve la primera instancia del elemento miembro.
●	`int getEnumerationValue()`	Solo se genera para tipos de enumeración. Devuelve una de las constantes generadas para los posibles valores o "no válido" si el valor no coincide con ninguno de los valores enumerados en el esquema.
●	`com.altova.xml.meta.Element getInfo()`	Devuelve un objeto para consultar información del esquema (véase `com.altova.xml.meta.Element` 1231).
●	`MemberType getValue()`	Obtiene el contexto del elemento (solo se genera si el elemento puede tener contenido simple o mixto).

© 2016-2022 Altova GmbH

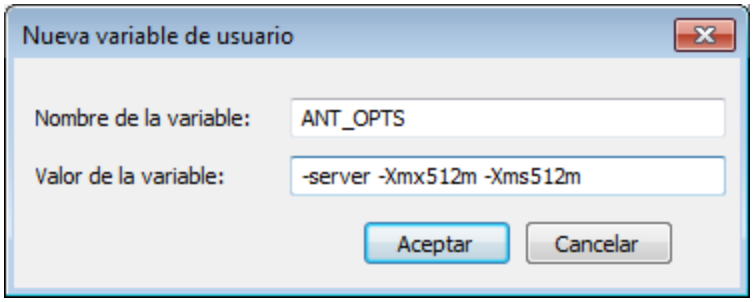
	Nombre	Descripción
	`java.util.Iterator iterator()`	Devuelve un objeto para recorrer instancias del elemento miembro.
	`MemberType last()`	Devuelve la última instancia del elemento miembro.
	`void` `remove()`	Elimina todas las instancias del elemento en su primario.
	`void` `removeAt(`**`int`** `index)`	Elimina la instancia del elemento especificada por el índice.
	`void` `setEnumerationValue(`**`int`** `index)`	Solo se genera para tipos de enumeración. Pase una de las constantes generadas para los posibles valores a este método para establecer el valor.
	`void` `setValue(MemberType value)`	Establece el contexto del elemento (solo se genera si el elemento puede tener contenido simple o mixto).

27.6 Consejos para la generación de código

Solución para las excepciones de memoria insuficiente durante la compilación de Java

Los esquemas complejos pueden producir una gran cantidad de código, lo cual puede dar lugar a una excepción java.lang.OutofMemory durante la compilación con Ant. Para resolver esto:

- Añada la variable de entorno **ANT_OPTS**, que asigna al compilador ciertas opciones de Ant, como la opción de memoria. Configure su valor como se muestra a continuación.



- Para asegurarse de que el compilador y el código generado se ejecutan en el mismo proceso que Ant, cambie el valor del atributo **fork**, de build.xml, por `false`.

Quizás sea necesario personalizar los valores, dependiendo de la cantidad de memoria disponible en su equipo y del tamaño del proyecto en el que esté trabajando. Para más información, consulte su documentación de Java VM.

Es posible que, al ejecutar el comando `ant jar`, reciba un mensaje de error parecido a este: "`[...] el archivo contiene más de 65535 entidades`". Para evitar este error, recomendamos utilizar Ant 1.9 o superior y, además, añadir `zip64mode="as-needed"` al elemento `<jar>` en el archivo **build.xml**.

Reservar nombres de métodos

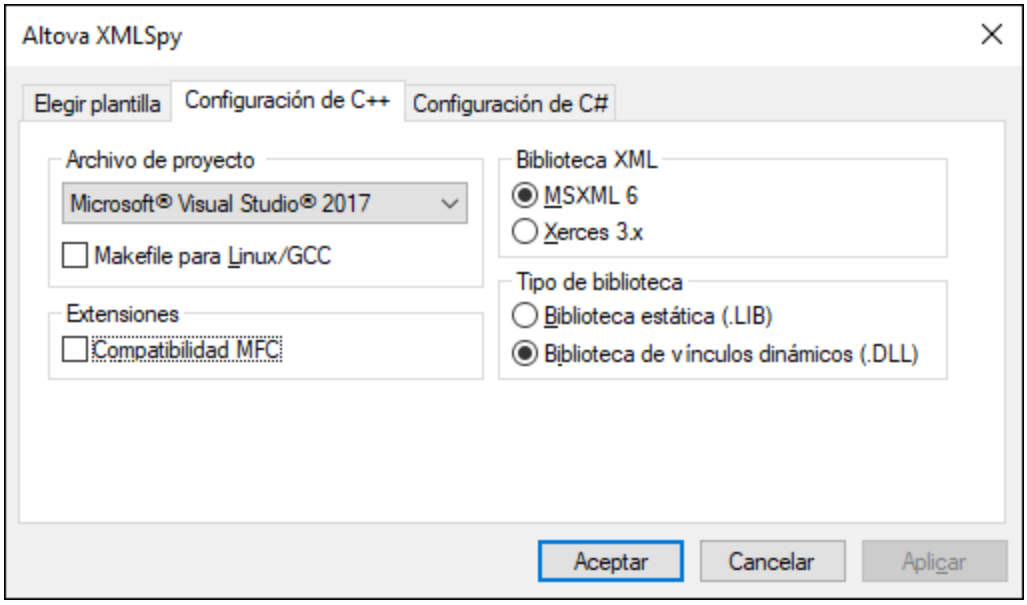
Cuando personalice la generación de código usando los archivos spl que vienen con la aplicación, quizás sea necesario reservar nombres para evitar conflictos con otros símbolos. Para hacerlo:

1. Navegue hasta el directorio de instalación del programa, por ejemplo `C:\Archivos de programa\Altova\XMLSpy2023\spl\java\`.
2. En el subdirectorio **spl** ubique el directorio que corresponda al lenguaje de programación, por ejemplo **..\spl\java**.
3. Abra el archivo **settings.spl** e inserte una línea nueva en la sección de reserva, por ejemplo **`reserve "miPalabraReservada"`**.
4. Vuelva a generar el código de programa.

© 2016-2022 Altova GmbH

27.7 Opciones del generador de código

El comando de menú **DTD/Esquema | Generar código de programa** sirve para especificar el lenguaje de programación de destino y para configurar la generación de código.



Estas son las opciones de configuración del generador de código:

Configuración de C++	Define la configuración del compilador para el entorno C++: • La edición de Visual Studio (2013, 2015, 2017, 2019, 2022) • Si debe generarse o no un archivo Make para Linux con el compilador GCC. • La biblioteca XML (MSXML, Xerces 3.x) • Si deben generarse bibliotecas estáticas o dinámicas • Si debe generarse código compatible con MFC o no Si marca la casilla *Makefile para Linux/GCC,* se añadirán archivos Make al código generado. Los archivos de código fuente C++ se generan de modo que sean portátiles gracias a construcciones `#ifdef` (esto permite la compatibilidad con diferentes compiladores y sistemas operativos). Si tiene pensado compilar el código generado con GCC (GNU Compiler Collection) en Linux debe tener en cuenta que: • Para la compilación con Linux/GCC la única biblioteca XML que es compatible es **Xerces 3.x**. • El estado de la casilla *Compatibilidad MFC* no afecta a la compilación con Linux/GCC.

Configuración de C#	Seleccione respectivamente la opción **Microsoft .NET Core 3.1**, **Microsoft .NET 5.0** o **Microsoft .NET 6.0** para generar una solución de Visual Studio para estas plataformas. Si necesita usar la plataforma **.NET Framework** para un proyecto de Visual Studio concreto, seleccione cualquiera de las opciones de **Microsoft Visual Studio 2010-2019**; en este caso la solución generada apuntará a la versión de .NET Framework que corresponde a la versión de Visual Studio que haya elegido.

© 2016-2022 Altova GmbH

27.8 SPL: el lenguaje de programación Spy

Esta sección ofrece una introducción al lenguaje de programación Spy (en adelante, *SPL*), que es el lenguaje de plantillas del generador de código.

En esta sección asumimos que el usuario tiene cierta experiencia en programación y ciertos conocimientos sobre operadores, variables, funciones y clases, así como sobre conceptos básicos de programación orientada a objetos, que se usa en gran medida en SPL. También asumimos que tiene conocimientos de XML Schema.

Las plantillas utilizadas por XMLSpy están en la carpeta **...\XMLSpy\spl**. Puede usar estos archivos para orientarse a la hora de crear sus propias plantillas.

¿Cómo funciona el generador de código?

Las entradas del generador de código son los archivos de plantilla (`.spl`) y el modelo de objetos que viene con XMLSpy. Los archivos de plantilla contienen instrucciones SPL (para crear archivos, leer información del modelo de objetos y realizar cálculos) intercaladas con fragmentos de código literal en el lenguaje de programación de destino.

El archivo de plantilla lo interpreta el generador de código y produce archivos de código fuente **.cpp**, **.java** y **.cs** , archivos de proyecto o cualquier otro tipo de proyecto, dependiendo de la plantilla. El código fuente se puede compilar después en un archivo ejecutable que accede a los datos XML descritos por el archivo de esquema.

Los archivos SPL tienen acceso a una gran variedad de información que se recopila de los esquemas de origen. Recuerde que un archivo SPL no está unido a ningún esquema concreto, sino que permite acceder a todos los esquemas. Asegúrese de que escribe sus archivos SPL de forma genérica y de que evita estructuras, etc. que pertenezcan a esquemas concretos.

Ejemplo: crear un archivo nuevo en SPL

Se trata de un archivo SPL muy básico. Crea un archivo llamado **test.cpp** y coloca la instrucción `include` dentro del archivo. El comando de cierre (`close`) termina la plantilla.

```
[create "test.cpp"]
#include "stdafx.h"
[close]
```

27.8.1 Estructura básica de SPL

Un archivo SPL contiene texto literal, que se debe reproducir, intercalado con instrucciones del generador de código.

Las instrucciones del generador de código van entre corchetes: "**[**" y "**]**".
Entre cada par de corchetes se pueden introducir varias instrucciones. Las instrucciones se separan con una línea nueva o con dos puntos "**:**".

Estos dos ejemplos son igual de válidos:

Ejemplo nº1

```
[$x = 42
$x = $x + 1]
```

Ejemplo nº2

```
[$x = 42: $x = $x + 1]
```

Agregar texto a los archivos

El texto que no aparece entre corchetes **[ ]** se escribe directamente en el archivo de salida actual. Si no hay un archivo de salida actual, el texto se pasa por alto (consulte el apartado Crear archivos de salida[1246]). Para reproducir corchetes literales, introduzca el carácter de escape \ de la siguiente manera: **\[** texto **\].** Para reproducir el carácter \, use \\.

Comentarios

Los comentarios introducidos en un bloque de instrucciones empiezan con el carácter **'** y terminan en la línea siguiente o con un carácter de cierre de bloque: **]**.

27.8.2 Declaraciones

Estas instrucciones se evalúan mientras se analiza el archivo de plantilla SPL. A estas instrucciones **no** les afectan las instrucciones de control de flujo como condiciones, bucles o subrutinas y siempre se evalúan exactamente una vez.

Estas palabras clave, como todas las palabras clave de SPL, no distinguen entre mayúsculas y minúsculas.

Recuerde que todas estas declaraciones deben estar dentro de un bloque delimitado por corchetes.

map *clave nombreasignación* to valor *[, clave to valor]...*

Esta instrucción añade información a una asignación. Consulte el resto del apartado para ver usos concretos de esta instrucción.

map *schemanativetype tipoesquema* to *tipoindicado*

El tipo integrado de XML Schema indicado se asignará al tipo o clase nativos indicados, usando el formateador indicado. Esta configuración es propia de la versión 2007r3 o posterior. *Tipoindicado* es el nombre de un tipo o clase nativos, seguido de una coma y de la instancia de clase del formateador.

Ejemplo:

```
map schemanativetype "double" to "double,Altova::DoubleFormatter"
```

map type *tipoesquema* to *nombreclase*

El tipo integrado de XML Schema indicado se asignará a la clase indicada. Esta configuración es propia de la generación de código para la versión 2007 o anterior.

© 2016-2022 Altova GmbH

Ejemplo:
```
map type "float" to "CSchemaFloat"
```

default *configuración* is *valor*

Esta instrucción permite determinar cómo se derivan los nombres de clase y miembro desde el esquema XML. Recuerde que los nombres de configuración sí distinguen entre mayúsculas y minúsculas.

Ejemplo:
```
default "InvalidCharReplacement" is "_"
```

Nombre de configuración	Explicación
ValidFirstCharSet	Caracteres permitidos para iniciar un identificador
ValidCharSet	Caracteres permitidos para otros caracteres en un identificador
InvalidCharReplacement	El carácter que reemplazará todos los caracteres de los nombres que no están en la configuración ValidCharSet
AnonTypePrefix	Prefijo para nombres de tipos anónimos*
AnonTypeSuffix	Sufijo para nombres de tipos anónimos*
ClassNamePrefix	Prefijo para los nombres de las clases generadas
ClassNameSuffix	Sufijo para los nombres de las clases generadas
EnumerationPrefix	Prefijo para constantes simbólicas declaradas para valores de enumeración
EnumerationUpperCase	"on" para poner los nombres de constante de enumeración en mayúsculas
FallbackName	Si un nombre tiene sólo caracteres que no están en ValidCharSet, use este

* Los nombres de tipos anónimos se construyen con AnonTypePrefix + nombre de elemento + AnonTypeSuffix

reserve *palabra*

Añade la palabra indicada a la lista de palabras reservadas. Esto garantiza que esa palabra se genere como nombre de clase o miembro.

Ejemplo:
```
reserve "while"
```

include *nombrearchivo*

Ejemplo:
```
include "Module.cpp"
```

incluye el archivo indicado como SPL de origen. Esto permite dividir su plantilla en varios archivos para editarlos y gestionarlos más fácilmente.

27.8.3 Variables

Los archivos SPL importantes exigen el uso de variables. Algunas variables vienen predefinidas 1244 por el generador de código y se pueden crear variables nuevas con solo asignarles valores.

El carácter **$** se usa cuando se **declara** o **usa** una variable, un nombre de variable siempre tiene el prefijo **$**. Los nombres de variable distinguen **entre mayúsculas y minúsculas**.

Tipos de variables:

- entero, que también se usa como binario, siendo 0 equivalente a false y cualquier otro valor equivale a true
- cadena
- objeto, que viene dado por XMLSpy
- iterador, ver instrucción foreach 1248

El tipo de variable se declara en la primera asignación de valor:

```
[$x = 0]
```
ahora x es un entero.

```
[$x = "cadenaDePrueba"]
```
ahora x es una cadena.

Cadenas

Las constantes de cadena siempre aparecen entre comillas dobles, como en el ejemplo anterior. **\n** y **\t** dentro de comillas dobles se considera una línea nueva y una tabulación, respectivamente. **\"** es una comilla doble literal y \\ es una barra diagonal inversa. Las constantes de cadena también pueden ocupar más de una línea.

Para la concatenación de cadenas se usa el carácter **&**:

```
[$BasePath = $outputpath & "/" & $JavaPackageDir]
```

Objetos

Los objetos representan la información que contiene el esquema XML. Los objetos tienen propiedades. Con el operador **.** puede acceder a estas propiedades. En SPL no puede crear objetos nuevos (vienen predefinidos por el generador de código y se derivan del esquema de entrada), pero se pueden asignar objetos a variables.

Ejemplo:

```
class [=$class.Name]
```

Este ejemplo reproduce la palabra "class" seguida de un espacio y el valor de la propiedad **Name** del objeto **$class**.

27.8.4 Variables predefinidas

Después de que el generador de código termine de analizar un archivo de esquema, los objetos de esta tabla existen en el motor de plantillas.

Nombre	Tipo	Descripción
$schematype	entero	1 para DTD, 2 para XML Schema

© 2016-2022 Altova GmbH

Nombre	Tipo	Descripción
$TheLibrary	Biblioteca 1253	La biblioteca que se deriva del esquema XML o de la DTD
$module	cadena	Nombre del esquema de origen sin la extensión
$outputpath	cadena	Ruta de acceso de salida especificada por el usuario o ruta de acceso de salida predeterminada

Para generación de código **C++** solamente:

Nombre	Tipo	Descripción
$domtype	entero	1 para MSXML, 2 para Xerces
$libtype	entero	1 para static LIB, 2 para DLL
$mfc	booleano	True si se marcó la casilla *Compatibilidad con MFC*
$VSVersion	entero	Especifica la versión de Visual Studio. Valores válidos: **0** Sin proyecto de Visual Studio **2010** Visual Studio 2010 **2013** Visual Studio 2013 **2015** Visual Studio 2015 **2017** Visual Studio 2017 **2019** Visual Studio 2019

Para generación de código **C#** solamente:

Nombre	Tipo	Descripción
$VSVersion	entero	Especifica la versión de Visual Studio. Valores válidos: **0** Sin proyecto de Visual Studio **2010** Visual Studio 2010 **2013** Visual Studio 2013 **2015** Visual Studio 2015 **2017** Visual Studio 2017 **2019** Visual Studio 2019

27.8.5 Crear archivos de salida

Estas instrucciones sirven para crear archivos de salida con ayuda de la función de generación de código.

Recuerde que todas estas instrucciones deben estar dentro de un bloque delimitado por corchetes.

create *nombrearchivo*

crea un archivo nuevo. El archivo tiene que cerrarse con la instrucción **close**. El resultado se escribe en el archivo indicado.

Ejemplo:
```
[create $outputpath & "/" & $JavaPackageDir & "/" & $application.Name & ".java"]
package [=$JavaPackageName];

public class [=$application.Name]Application {
...
}
[close]
```

close

cierra el archivo de salida actual

=$*variable*

escribe el valor de la variable indicada en el archivo de salida actual.

Ejemplo:
```
[$x = 20+3]
El resultado de su cálculo es [=$x] - ¡que tenga un buen día!
```

- El archivo de salida incluirá:

```
El resultado de su cálculo es 23 - ¡que tenga un buen día!
```

write *cadena*

escribe la cadena en el archivo de salida actual.

Ejemplo:
```
[write "C" & $name]
```

Esto también se puede escribir de esta manera:
```
C[=$name]
```

filecopy *origen* to *destino*

copia el archivo de origen al archivo de destino, sin interpretarlo.

© 2016-2022 Altova GmbH

Ejemplo:

```
filecopy "java/mapforce/mapforce.png" to $outputpath & "/" & $JavaPackageDir &
"/mapforce.png"
```

27.8.6 Operadores

Los operadores de SPL funcionan como en casi todos los lenguajes de programación.

A continuación ofrecemos la lista de operadores de SPL ordenados por orden de prioridad (de mayor a menor):

.	Acceso a la propiedad del objeto
( )	Agrupación de expresiones
true	Constante binaria "true"
false	Constante binaria "false"
&	Concatenación de cadenas de texto
-	Signo para un número negativo
not	Negación lógica
*	Multiplicación
/	División
%	Resto
+	Suma
-	Resta
<=	Menor o igual que
<	Menor que
>=	Mayor o igual que
>	Mayor que
=	Igual
<>	No igual
and	Conjunción lógica (con evaluación en cortocircuito)
or	Disyunción lógica (con evaluación en cortocircuito)
=	Asignación

27.8.7 Condiciones

En SPL puede usar instrucciones `if` estándar. Esta es la sintaxis para estas instrucciones:

```
if condición
       instrucciones
else
```

```
        instrucciones
endif
```

o sin la parte **else**:

```
if condición
        instrucciones
endif
```

Nota: observe que la condición no aparece entre paréntesis.
Al igual que en otros lenguajes de programación, las condiciones se construyen con operadores[1247] lógicos y de comparación.

Ejemplo:

```
[if $namespace.ContainsPublicClasses and $namespace.Prefix <> ""]
        lo que usted quiera ['inserta lo que usted quiera en el archivo de resultados]
[endif]
```

Instrucción *switch*

SPL también incluye una instrucción de control para múltiples selecciones.

Sintaxis:

```
switch $variable
        case X:
                instrucciones
        case Y:
        case Z:
                instrucciones
        default:
                instrucciones
endswitch
```

Las etiquetas "case" son constantes o variables.

Al igual que C, SPL no admite el paso implícito de una etiqueta case a otra y, por tanto, no es necesario usar la instrucción `break`.

27.8.8 Colecciones y foreach

Colecciones e iteradores

Una colección contiene varios objetos, como una matriz normal y corriente. Los iteradores sirven para almacenar e incrementar índices de matriz al acceder a objetos.

Sintaxis:

```
foreach iterador in colección
        instrucciones
next
```


© 2016-2022 Altova GmbH

Ejemplo nº1:

```
[foreach $class in $classes
       if not $class.IsInternal
              ]      class [=$class.Name];
[      endif
next]
```

Ejemplo 2:

```
[foreach $i in 1 To 3
      Write "// Step " & $i & "\n"
      ' Do some work
next]
```

En la primera línea **$classes** es el **objeto global**[1244] de todos los tipos generados. Es una colección de diferentes objetos de clase.

La instrucción **foreach** recorre todos los elementos de **$classes** y en cada uno de ellos ejecuta el código que sigue a la instrucción hasta llegar a la instrucción **next**.

En cada iteración se asigna **$class** al siguiente objeto de clase. Simplemente se trabaja con el objeto de clase en lugar de usar `classes[i]->Name()` como en C++.

Los iteradores de colección tienen estas propiedades adicionales:

`Index`	El índice actual, empezando por 0
`IsFirst`	true si el objeto actual es el primer objeto de la colección (el índice es 0)
`IsLast`	true si el objeto actual es el último objeto de la colección
`Current`	El objecto actual (este es implícito si no se especifica y se puede omitir)

Ejemplo:

```
[foreach $enum in $facet.Enumeration
       if not $enum.IsFirst
              ], [
       endif
       ]"[=$enum.Value]"[
next]
```

Rutinas para la manipulación de colecciones:

collection **SortByName**(bAscending)
devuelve una colección cuyos elementos están ordenados por nombre (con distinción de mayúsculas y minúsculas) en orden ascendente o descendente.

collection **SortByNameNoCase**(bAscending)
devuelve una colección cuyos elementos están ordenados por nombre (sin distinción de mayúsculas y minúsculas) en orden ascendente o descendente.

Ejemplo:

```
$SortedNestedClassifier = $Class.nestedClassifier.SortByNameNoCase( true )
```

collection **SortByKind**(bAscending)
devuelve una colección cuyos elementos están ordenados por tipo (p. ej. clase, interfaz...) en orden ascendente o descendente.

collection **SortByKindAndName**(bAscendingKind, bAscendingName)
devuelve una colección cuyos elementos están ordenados por tipo (p. ej. clase, interfaz...) en orden ascendente o descendente y si el tipo es idéntico al nombre (con distinción de mayúsculas y minúsculas en orden ascendente o descendente).

collection **SortByKindAndNameNoCase**(bAscending)
devuelve una colección cuyos elementos están ordenados por tipo (p. ej. clase, interfaz...) en orden ascendente o descendente y si el tipo es idéntico al nombre (sin distinción de mayúsculas y minúsculas en orden ascendente o descendente).

27.8.9 Subrutinas

El generador de código admite subrutinas en forma de procedimientos o funciones.

Características:
- Se pasan valores por valor y por referencia
- Parámetros locales/globales (locales dentro de las subrutinas)
- Variables locales
- Invocación
- Invocación recurrente (las subrutinas se pueden llamar a sí mismas)

27.8.9.1 Declaración de subrutinas

Subrutinas

Ejemplo de sintaxis:

```
Sub SimpleSub()
... lines of code
EndSub
```

- **Sub** es la palabra clave que denota el procedimiento.
- **SimpleSub** es el nombre asignado a la subrutina.
- Los **paréntesis** pueden contener una lista de parámetros.
- El bloque de código de una subrutina empieza inmediatamente después del paréntesis de cierre.
- **EndSub** denota el final del bloque de código.

Nota: no se permiten **declaraciones** de subrutinas recursivas o en cascada (es decir, una subrutina no puede contener otra subrutina).

Parámetros

Los parámetros también se pueden pasar con procedimientos usando esta sintaxis:

© 2016-2022 Altova GmbH

- Todos los parámetros deben ser variables.
- Las variables deben tener el prefijo **$**.
- Las variables locales se definen en una subrutina.
- Las variables globales se declaran de forma explícita, fuera de las subrutinas.
- Si hay varios parámetros, se separan con comas dentro de los paréntesis.
- Los parámetros pueden pasar valores.

Parámetros: pasar valores

Los parámetros se pueden pasar de dos maneras: por valor y por referencia, usando las palabras clave **ByVal** y **ByRef** respectivamente.

Sintaxis:

```
' definir sub CompleteSub()
[Sub CompleteSub( $param, ByVal $paramByValue, ByRef $paramByRef )
] ...
```

- **ByVal** especifica que el parámetro se pasa por valor. Recuerde que la mayoría del os objetos solamente se pueden pasar por referencia.
- **ByRef** especifica que el parámetro se pasa por referencia. Se trata del método predeterminado si no se especifica ByVal ni ByRef.

Valores devueltos de funciones

Para devolver un valor desde una subrutina use la instrucción **return**. Este tipo de función se puede llamar desde dentro de una expresión.

Ejemplo:

```
' definir una función
[Sub MakeQualifiedName( ByVal $namespacePrefix, ByVal $localName )
if $namespacePrefix = ""
  return $localName
else
  return $namespacePrefix & ":" & $localName
endif
EndSub
]
```

27.8.9.2 Invocación de subrutinas

Use **call** para invocar una subrutina, seguido del nombre y los parámetros del procedimiento (si procede).

```
      Call SimpleSub()
```

o

```
      Call CompleteSub( "FirstParameter", $ParamByValue, $ParamByRef )
```

Invocación de funciones

Para invocar una función (cualquier subrutina que contenga una instrucción **return**) use su nombre dentro de una expresión. No use la instrucción **call** para llamar a funciones.

Ejemplo:

```
$QName = MakeQualifiedName($namespace, "entry")
```

27.8.9.3 Ejemplo

Este ejemplo muestra la declaración e invocación de subrutinas.

```
[create $outputpath & $module & "output.txt"

' definir sub SimpleSub()
Sub SimpleSub()
]SimpleSub() called
[endsub

' ejecutar sub SimpleSub()
Call SimpleSub()

$ParamByValue        = "Original Value"
]ParamByValue        = [=$ParamByValue]
[$ParamByRef  = "Original Value"
]ParamByRef   = [=$ParamByRef]

' definir sub CompleteSub()
[Sub CompleteSub( $param, ByVal $paramByValue, ByRef $paramByRef )
]CompleteSub called.
       param = [=$param]
       paramByValue = [=$paramByValue]
       paramByRef = [=$paramByRef]
[$ParamByRef = "Local Variable"
$paramByValue = "new value"
$paramByRef = "new value"
]      Set values inside Sub
[$ParamByRef = "Local Variable"
$paramByValue = "new value"
$paramByRef = "new value"
]CompleteSub finished.
[endsub

' ejecutar sub CompleteSub()
Call CompleteSub( "FirstParameter", $ParamByValue, $ParamByRef )
]
ParamByValue=[=$ParamByValue]
ParamByRef=[=$ParamByRef]
[
Close
]
```

27.8.10 Tipos integrados

Esta sección describe las propiedades de los tipos integrados usados en las variables predefinidas[1244] que describen el esquema analizado.

© 2016-2022 Altova GmbH

27.8.10.1 Library

Este objeto representa toda la biblioteca generada a partir del esquema XML o de la DTD.

Propiedad	Tipo	Descripción
SchemaNamespaces	Colección Namespace 1253	Los espacios de nombres de esta biblioteca
SchemaFilename	string	Nombre del XSD o DTD del que se deriva esta biblioteca
SchemaType	integer	1 para la DTD, 2 para el esquema XML
Guid	string	Un ID único global
CodeName	string	El nombre de la biblioteca generada (derivado del nombre del archivo de esquema)

27.8.10.2 Namespace

Se genera un objeto namespace por cada espacio de nombres del esquema XML. Los componentes del esquema que no están en ningún espacio de nombres se incluyen en un objeto namespace especial con un NamespaceURI vacío. Recuerde que para las DTD también se derivan espacios de nombres de los atributos cuyo nombre comienza por "xmlns".

Propiedad	Tipo	Descripción
CodeName	string	Nombre para el código generado (derivado del prefijo)
LocalName	string	Prefijo del espacio de nombres
NamespaceURI	string	URI del espacio de nombres
Types	Colección Type 1253	Todos los tipos incluidos en este espacio de nombres
Library	Library 1253	La biblioteca que contiene este espacio de nombres

27.8.10.3 Type

Este objeto representa un tipo simple o complejo. Se usa para generar una clase en el lenguaje de destino. Hay un tipo adicional por cada biblioteca que representa al documento, que tiene como miembros todos los elementos raíz posibles.
Los tipos anónimos tienen la propiedad LocalName vacía.

Propiedad	Tipo	Descripción
CodeName	string	Nombre para el código generado (derivado del nombre local o de la declaración de objeto primario)
LocalName	string	Nombre original en el esquema
Namespace	Namespace 1253	Espacio de nombres que contiene este tipo

Attributes	Colección Member 1254	Atributos que contiene este tipo*
Elements	Colección Member 1254	Elementos secundarios que contiene este tipo
IsSimpleType	boolean	True para tipos simples y false para tipos complejos
IsDerived	boolean	True si el tipo se deriva de otro tipo, lo cual también se representa con un objeto Type
IsDerivedByExtension	boolean	True si el tipo se deriva por extensión
IsDerivedByRestriction	boolean	True si el tipo se deriva por restricción
IsDerivedByUnion	boolean	True si el tipo se deriva mediante union
IsDerivedByList	boolean	True si el tipo se deriva mediante list
BaseType	Type	El tipo base de este tipo (si IsDerived es true)
IsDocumentRootType	boolean	True si este tipo representa al documento propiamente dicho
Library	Library 1253	La biblioteca que contiene este tipo
IsFinal	boolean	True si se declaró como final en el esquema
IsMixed	boolean	True si este tipo puede tener contenido mixto
IsAbstract	boolean	True si este tipo se declaró como abstracto
IsGlobal	boolean	True si este tipo se declaró globalmente en el esquema
IsAnonymous	boolean	True si este tipo se declaró localmente en un elemento

Para tipos simples solamente:

Propiedad	Tipo	Descripción
IsNativeBound	boolean	True si existe un NativeBinding para este tipo
NativeBinding	NativeBinding 1255	El NativeBinding para este tipo
Facets	Facets 1256	Las facetas de este tipo
Whitespace	string	Acceso rápido a la faceta Whitespace

* Los tipos complejos que tienen contenido de texto (es decir, tipos que tienen contenido mixto y un complexType con simpleContent) tienen un miembro atributo adicional sin nombre que representa el contenido de texto.

27.8.10.4 Member

Este objeto representa a un atributo o elemento del esquema XML. Se usa para crear miembros de clases de tipos.

Propiedad	Tipo	Descripción
CodeName	string	El nombre para el código generado (derivado del nombre local o de la declaración del objeto

© 2016-2022 Altova GmbH

		primario)
LocalName	string	El nombre original en el esquema. Aparece vacía para el miembro especial que representa el contenido de texto de tipos complejos.
NamespaceURI	string	El URI de espacio de nombres de este elemento/atributo dentro de documentos/secuencias de instancia XML.
DeclaringType	Type 1253	El tipo que originalmente declara el miembro (igual a containingType para miembros no heredados)
ContainingType	Type 1253	El tipo del cual este es miembro
DataType	Type 1253	El tipo de datos del contenido de este miembro
Library	Library 1253	La biblioteca que contiene el DataType de este miembro
IsAttribute	boolean	True para atributos, falso para elementos
IsOptional	boolean	True si minOccurs = 0 o si es un atributo opcional
IsRequired	boolean	True si minOccurs > 0 o si es un atributo obligatorio
IsFixed	boolean	True para atributos fijos, el valor está en la propiedad Default
IsDefault	boolean	True para los atributos con valor predeterminado, el valor está en la propiedad Default
IsNillable	boolean	True para elementos nillable
IsUseQualified	boolean	True si la propiedad NamespaceURI no está vacía
MinOccurs	integer	minOccurs como en el esquema. 1 para atributos obligatorios
MaxOccurs	integer	maxOccurs como en el esquema. 0 para atributos prohibidos y -1 para unbounded
Default	string	Valor predeterminado

27.8.10.5 NativeBinding

Este objeto representa el enlace de un tipo simple con un tipo nativo en el lenguaje de programación de destino, tal y como indica la asignación "schemanativetype".

Propiedad	**Tipo**	**Descripción**
ValueType	string	Tipo nativo
ValueHandler	string	Instancia de clase de formateador

27.8.10.6 Facets

Este objeto representa todas las facetas de un tipo simple.
Las facetas heredadas se combinan con las facetas declaradas de forma explícita. Si hay una faceta Length, entonces MinLength y MaxLength tienen el mismo valor.

Propiedad	Tipo	Descripción
DeclaringType	Type	El tipo en el que están declaradas las facetas
Whitespace	string	"preserve", "collapse" o "replace"
MinLength	integer	Valor de faceta
MaxLength	integer	Valor de faceta
MinInclusive	integer	Valor de faceta
MinExclusive	integer	Valor de faceta
MaxInclusive	integer	Valor de faceta
MaxExclusive	integer	Valor de faceta
TotalDigits	integer	Valor de faceta
FractionDigits	integer	Valor de faceta
List	Colección Facet	Todas las facetas en forma de lista

Facet

Este objeto representa una sola faceta cuyo valor calculado está vigente en un tipo concreto.

Propiedad	Tipo	Descripción
LocalName	string	Nombre de la faceta
NamespaceURI	string	Espacio de nombres de la faceta
FacetType	string	Puede ser "normalization", "lexicalspace", "valuespace-length", "valuespace-enum" o "valuespace-range"
DeclaringType	Type 1253	El tipo en el que se declara esta faceta
FacetCheckerName	string	El nombre del Facet Checker (de la asignación schemafacet)
FacetValue	string o integer	El valor real de esta faceta

© 2016-2022 Altova GmbH

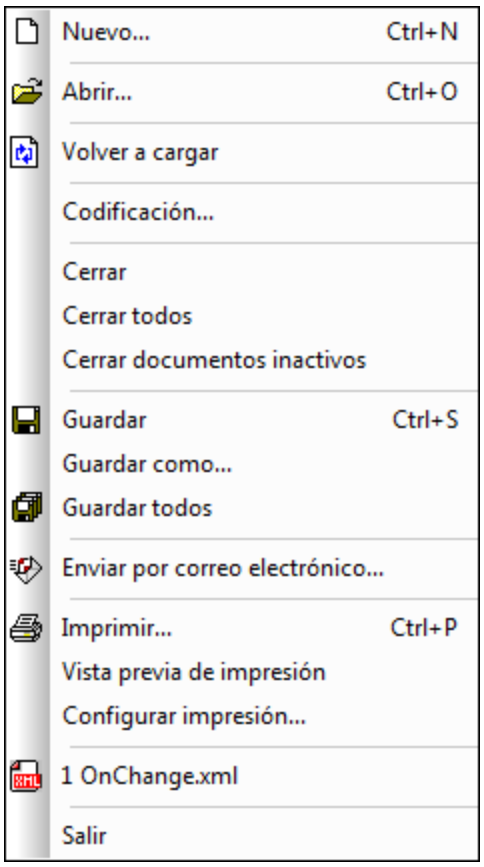
28 Comandos de menú

La sección ***Comandos de menú*** ofrece una descripción completa de todos los comandos y menús de XMLSpy y explica cómo se utilizan. Si tiene alguna duda y no encuentra lo que busca en esta sección ni en el resto de la documentación, consulte las secciones de preguntas frecuentes y foros de debate del sitio web de Altova. Si sigue sin encontrar lo que busca, póngase en contacto con el centro de soporte técnico de Altova[1636].

Los comandos estándar de Windows (**Abrir**, **Guardar**, **Cortar**, **Copiar**, **Pegar**, etc.) están en los menús **Archivo**[1258] y **Edición**[1279]. Estos menús también incluyen comandos relacionados con XML e Internet.

28.1 Menú Archivo

Los comandos del menú **Archivo** sirven para realizar operaciones con archivos. Además de los comandos estándar **Nuevo**[1258], **Abrir**[1262], **Guardar**[1269], **Imprimir**[1275], **Configurar impresión**[1277] y **Salir**[1278], el menú **Archivo** incluye comandos característicos de XMLSpy.



28.1.1 Nuevo

En este apartado:

- Icono y acceso rápido[1259]
- Descripción[1259]
- Plantillas para documentos nuevos[1259]
- Asignar una DTD o un esquema XML a un documento XML nuevo[1260]
- Especificar el elemento raíz de un documento XML nuevo[1261]
- Asignar un archivo SPS a un documento XML nuevo[1261]
- Crear taxonomías XBRL nuevas con el Asistente para taxonomías XBRL[1262]

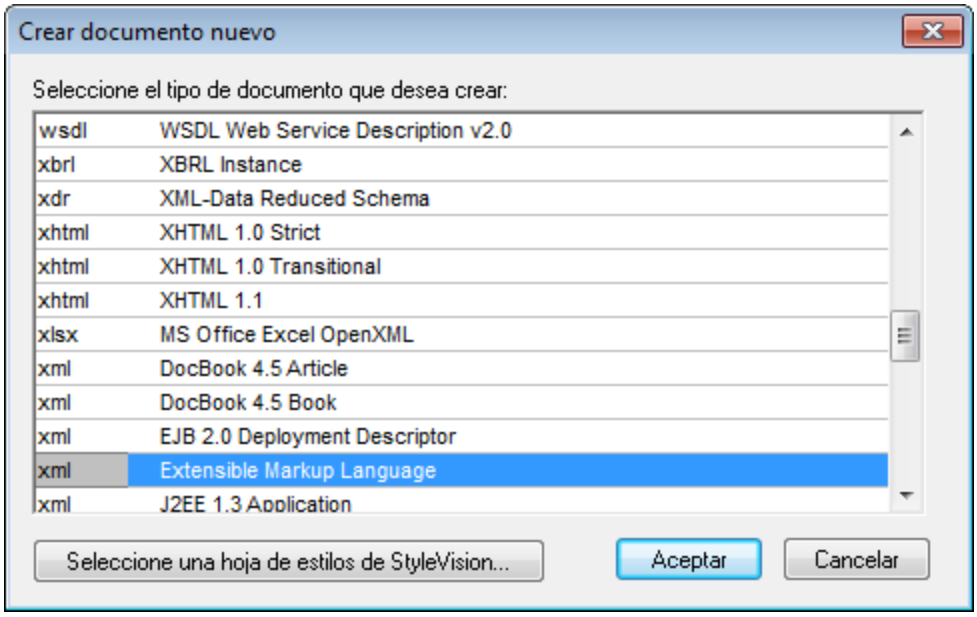
© 2016-2022 Altova GmbH

Icono y acceso rápido

Icono:	
Acceso rápido:	**Ctrl+N**

Descripción

El comando **Nuevo** sirve para crear un documento nuevo. Al hacer clic en este comando aparece el cuadro de diálogo "Crear documento nuevo" (*imagen siguiente*), donde puede seleccionar qué tipo de documento desea crear. Si el tipo de documento que desea crear no aparece en la lista del cuadro de diálogo, seleccione la opción `xml` y cambie la extensión de archivo al guardarlo. Recuerde que puede agregar más tipos de archivo a la lista del cuadro de diálogo (desde **Herramientas | Opciones | Tipos de archivo**[1587]).



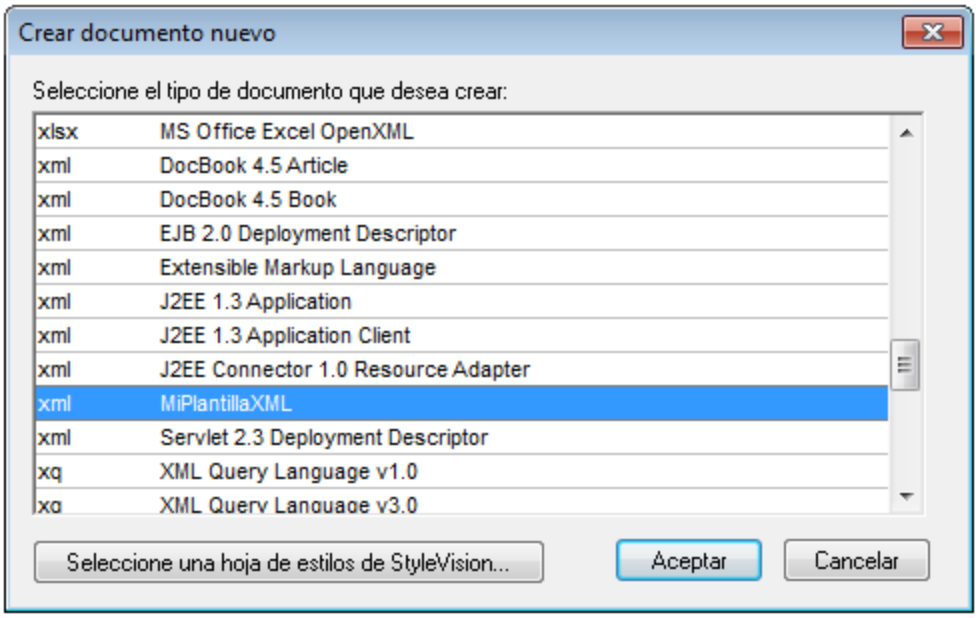
Plantillas para documentos nuevos

La lista de tipos de archivos del cuadro de diálogo "Crear documento nuevo" puede incluir plantillas definidas por el usuario para cualquier tipo de documento. Estas plantillas se pueden abrir directamente desde el cuadro de diálogo "Crear documento nuevo" para editarlas. Para crear su propia plantilla de documento y que aparezca en la lista del cuadro de diálogo "Crear documento nuevo", primero tiene que crear el documento de plantilla y guardarlo en la carpeta que contiene todas las plantillas.

Para crear una plantilla de documento:

1. Abra la carpeta `XMLSpy\Template` de la carpeta de la aplicación[36] desde el explorador de Windows y seleccione un archivo de plantilla de los archivos que se llaman `new.xxx` (siendo `.xxx` la extensión de archivo, como `.xml` o `.xslt`).
2. Abra el archivo en XMLSpy y haga los cambios necesarios. Este archivo será el archivo de plantilla.

3. Cuando termine haga clic en **Archivo | Guardar como...** para guardar el archivo en la carpeta `XMLSpy\Template` y póngale un nombre adecuado, como `MiPlantillaXML.xml`, por ejemplo. Ahora tiene una plantilla llamada `MiPlantillaXML`, que aparece en la lista de archivos del cuadro de diálogo "Crear documento nuevo".



4. Para abrir la plantilla, seleccione **Archivo | Nuevo** y después la plantilla (`MiPlantillaXML` en este caso).

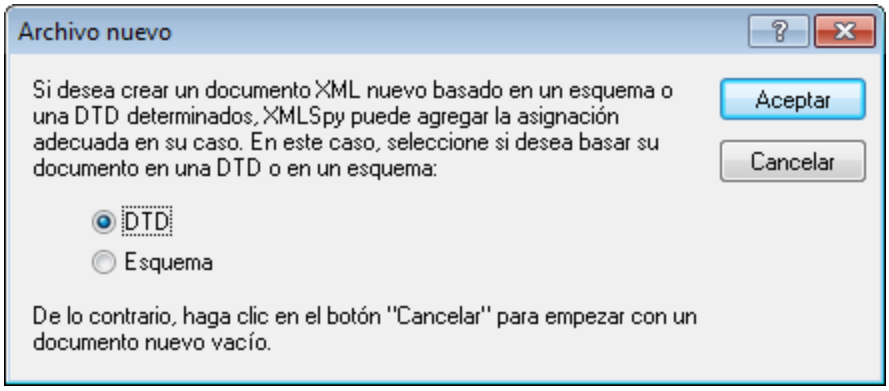
Para eliminar una plantilla, elimine el archivo de plantilla de la carpeta `XMLSpy\Template`.

Asignar una DTD o esquema XML al documento XML nuevo

Cuando cree un documento nuevo desde el cuadro de diálogo "Crear documento nuevo", el documento se abre automáticamente con la asociación de esquema XML o DTD correcta. Por ejemplo, si abre un archivo XHTML, este se abre con la DTD `http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd` asociada, mientras que los archivos de esquema XML (`.xsd`) están asociados al documento de esquema `http://www.w3.org/2001/XMLSchema`.

Si crea un documento nuevo cuyo esquema se desconoce (un archivo XML, por ejemplo), XMLSpy le pide que asocie un esquema (esquema XML o DTD) al nuevo documento (*imagen siguiente*).

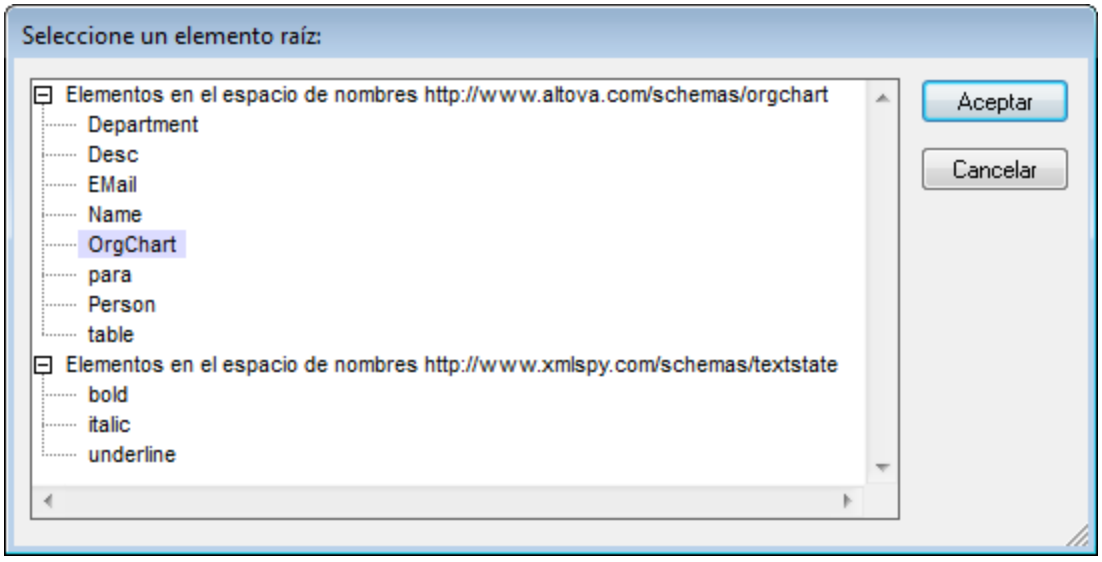
© 2016-2022 Altova GmbH



Si en el cuadro de diálogo selecciona la opción *DTD* o *Esquema* y hace clic en **Aceptar**, en el paso siguiente podrá buscar el esquema que desea usar. Si hace clic en **Cancelar**, se crea un documento que no está asociado a ningún esquema.

Especificar el elemento raíz de un documento XML nuevo

Si selecciona un esquema XML como esquema asociado de un documento XML y este esquema tiene varios elementos globales, todos los elementos globales se pueden usar como elemento raíz. Cuando esto ocurre, aparece el cuadro de diálogo "Seleccione un elemento raíz" (*imagen siguiente*), donde puede seleccionar qué elemento global debe servir como elemento raíz del documento XML. En el ejemplo de la imagen siguiente está seleccionado el elemento global `Document`.

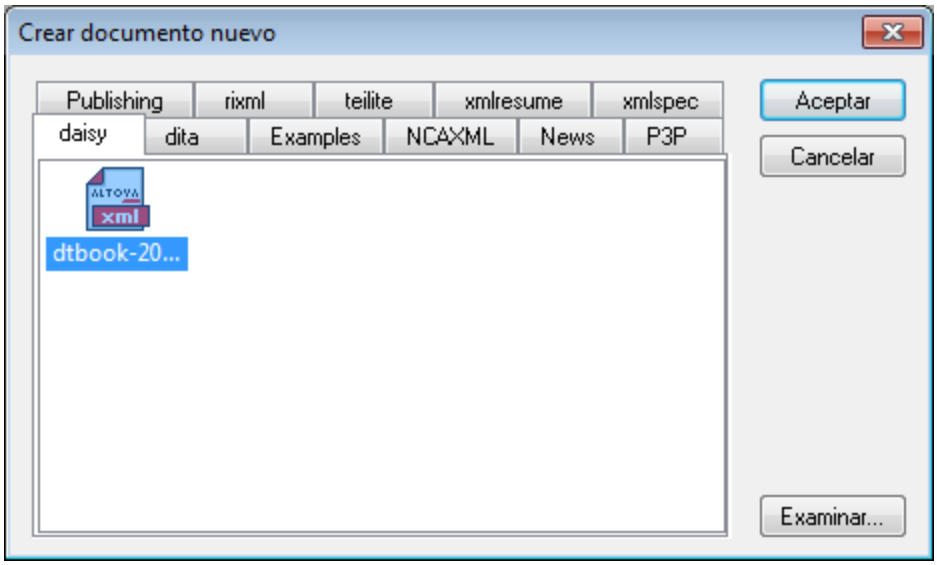


Al hacer clic en **Aceptar** se crea un documento XML nuevo cuyo elemento raíz es el elemento `Documento`.

Asignar un archivo SPS a un documento XML nuevo

Cuando cree un documento XML nuevo, puede asociarlo a un archivo SPS (StyleVision Power Stylesheet) para poder ver el documento en la vista Authentic. En el cuadro de diálogo "Crear documento nuevo" (*primera*

imagen del apartado), haga clic en el botón **Seleccione una hoja de estilos de StyleVision**. En ese momento aparece un cuadro de diálogo llamado "Crear documento nuevo" (*imagen siguiente*).



En las pestañas de este cuadro de diálogo puede buscar el archivo SPS que desea usar. También puede hacer clic en el botón **Examinar...** para buscar un archivo SPS distinto.

Crear taxonomías XBRL nuevas con el Asistente para taxonomías XBRL

Si selecciona la opción *XBRL Taxonomy Schema* (`.xsd`) en el cuadro de diálogo "Crear documento nuevo", entonces aparece el asistente para la creación de taxonomías, que le guiará en el proceso de creación de la taxonomía XBRL. Este asistente se describe en detalle en la sección XBRL[838] de la documentación.

28.1.2 Abrir

Icono y acceso rápido

Icono:	
Acceso rápido:	**Ctrl+O**

Descripción

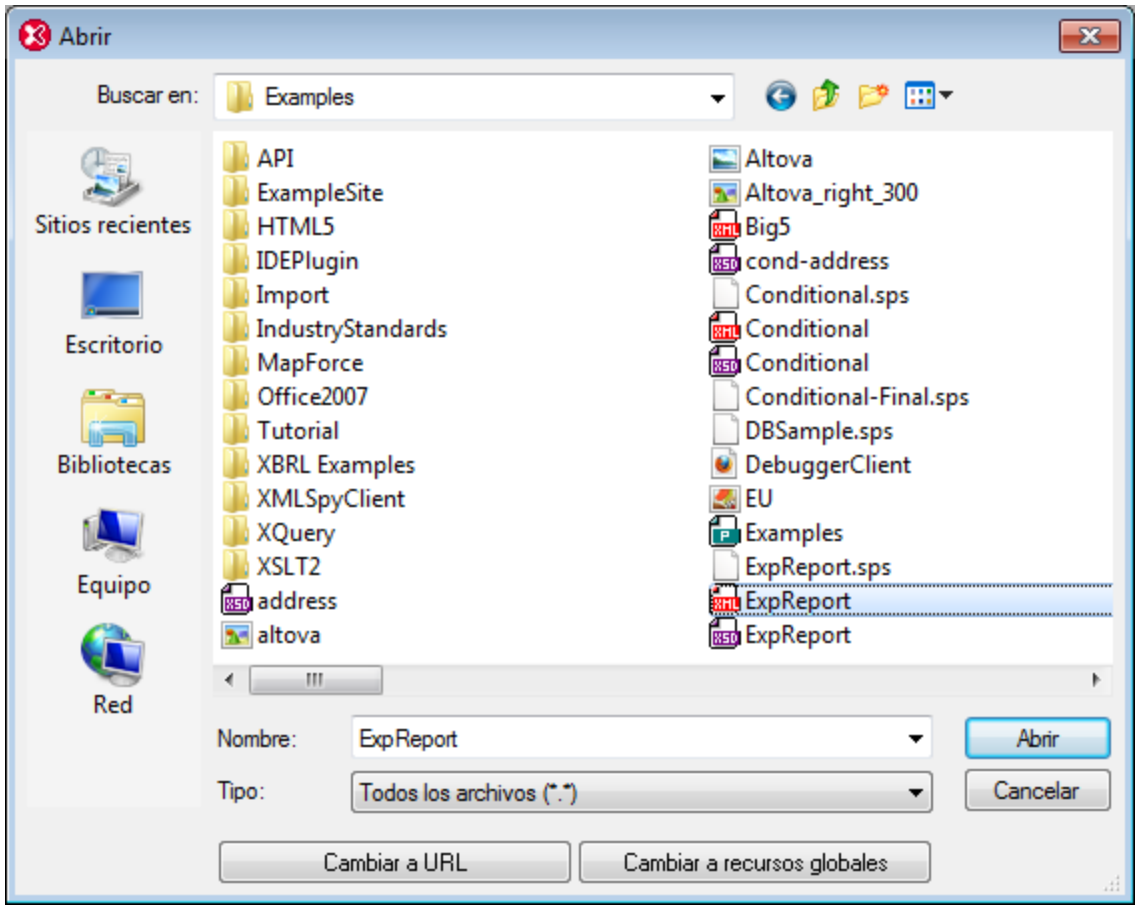
El comando **Abrir** abre el cuadro de diálogo "Abrir" de Windows, donde puede abrir cualquier documento XML o de texto. En el cuadro de diálogo "Abrir" puede seleccionar más de un archivo. Use el cuadro combinado *Tipo:* para filtrar la lista de archivos que aparecen en el cuadro de diálogo. (La lista de tipos de archivos disponibles se puede configurar en la sección *Tipos de archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**[1587])). Al abrir un archivo XML, XMLSpy comprueba inmediatamente si tiene un formato XML correcto. Si no tiene un formato XML correcto, entonces se emite un mensaje de error. Corrija el error y seleccione el comando de menú **XML | Comprobar formato XML (F7)**[1335] para volver a revisar el documento. Si seleccionó la opción de validación automática[1584] y el archivo no es válido, entonces

© 2016-2022 Altova GmbH

aparece un mensaje de error. Corrija el error y seleccione el comando **XML | Validar el documento XML (F8)** 1344 para volver a validar el archivo.

▼ Seleccionar y guardar archivos a través de direcciones URL y recursos globales

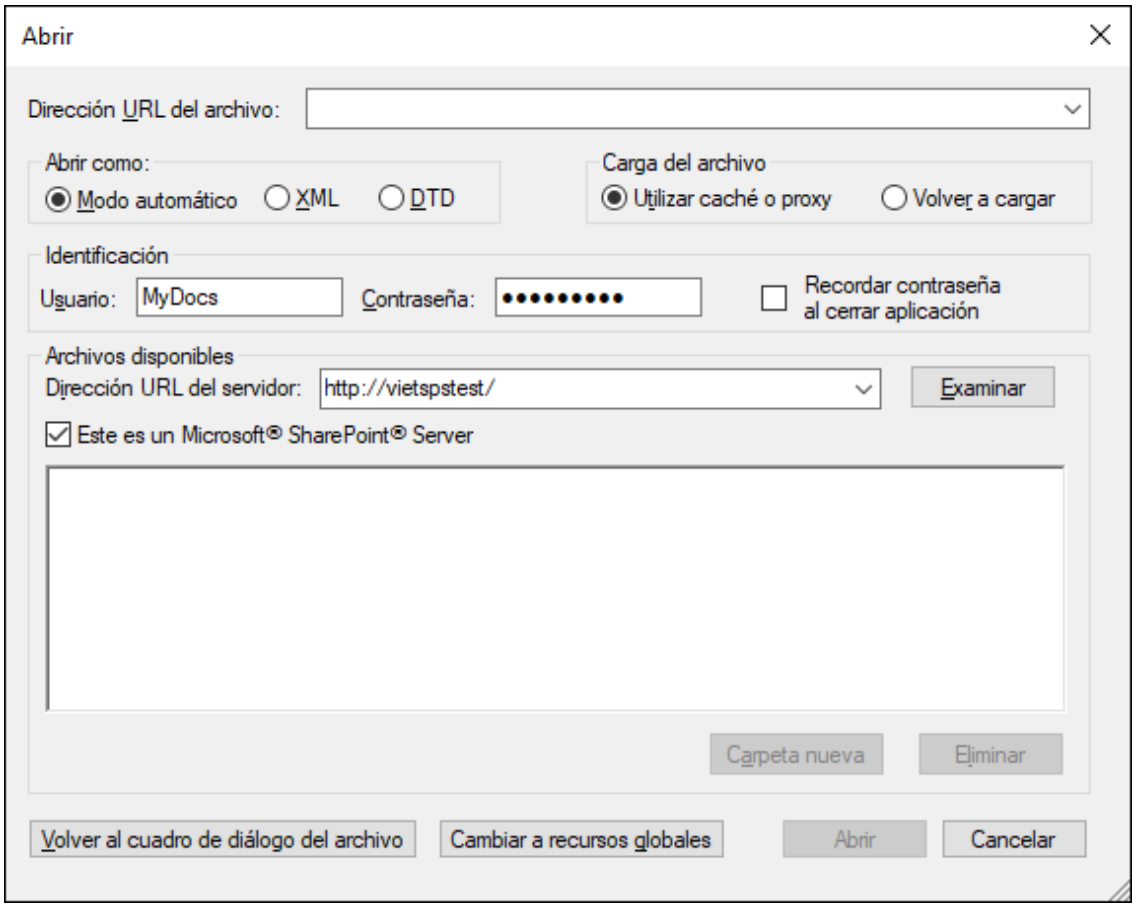
En varios cuadros de diálogo "Abrir archivo" y "Guardar archivo" de XMLSpy podrá seleccionar o guardar el archivo a través de una URL o un recurso global (*imagen siguiente*). Haga clic en el botón **Cambiar a URL** o **Cambiar a recursos globales** según corresponda.



Seleccionar archivos a través de direcciones URL

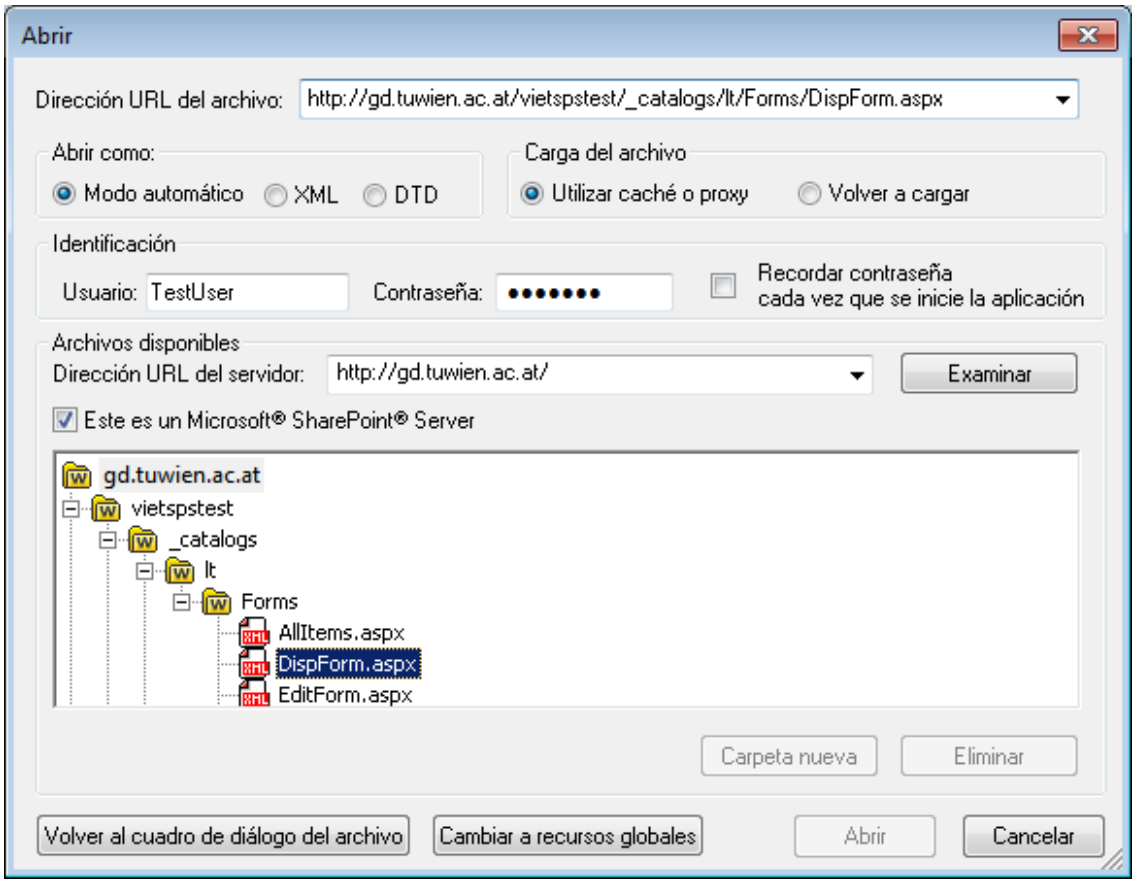
Siga estas instrucciones para seleccionar un archivo a través de una dirección URL (ya sea para abrirlo o para guardarlo):

1. Haga clic en el botón **Cambiar a URL**. El cuadro de diálogo cambia al modo URL (*imagen siguiente*).



2. Escriba la URL en el campo *Dirección URL del servidor:* (*imagen anterior*). Si el servidor es un servidor Microsoft® SharePoint® Server, marque la casilla *Microsoft® SharePoint® Server*. Consulte la nota sobre este tipo de servidores que aparece más abajo.
3. Si el servidor está protegido con contraseña, escriba el ID de usuario y la contraseña en los campos correspondientes.
4. Haga clic en **Examinar** para navegar por la estructura de directorios del servidor.
5. Busque el archivo que desea cargar y selecciónelo.

© 2016-2022 Altova GmbH



El archivo URL aparece en el campo URL del archivo (imagen anterior). En ese momento se habilita el botón Abrir/Guardar.

6. Haga clic en **Abrir** para cargar el archivo o en **Guardar** para guardarlo.

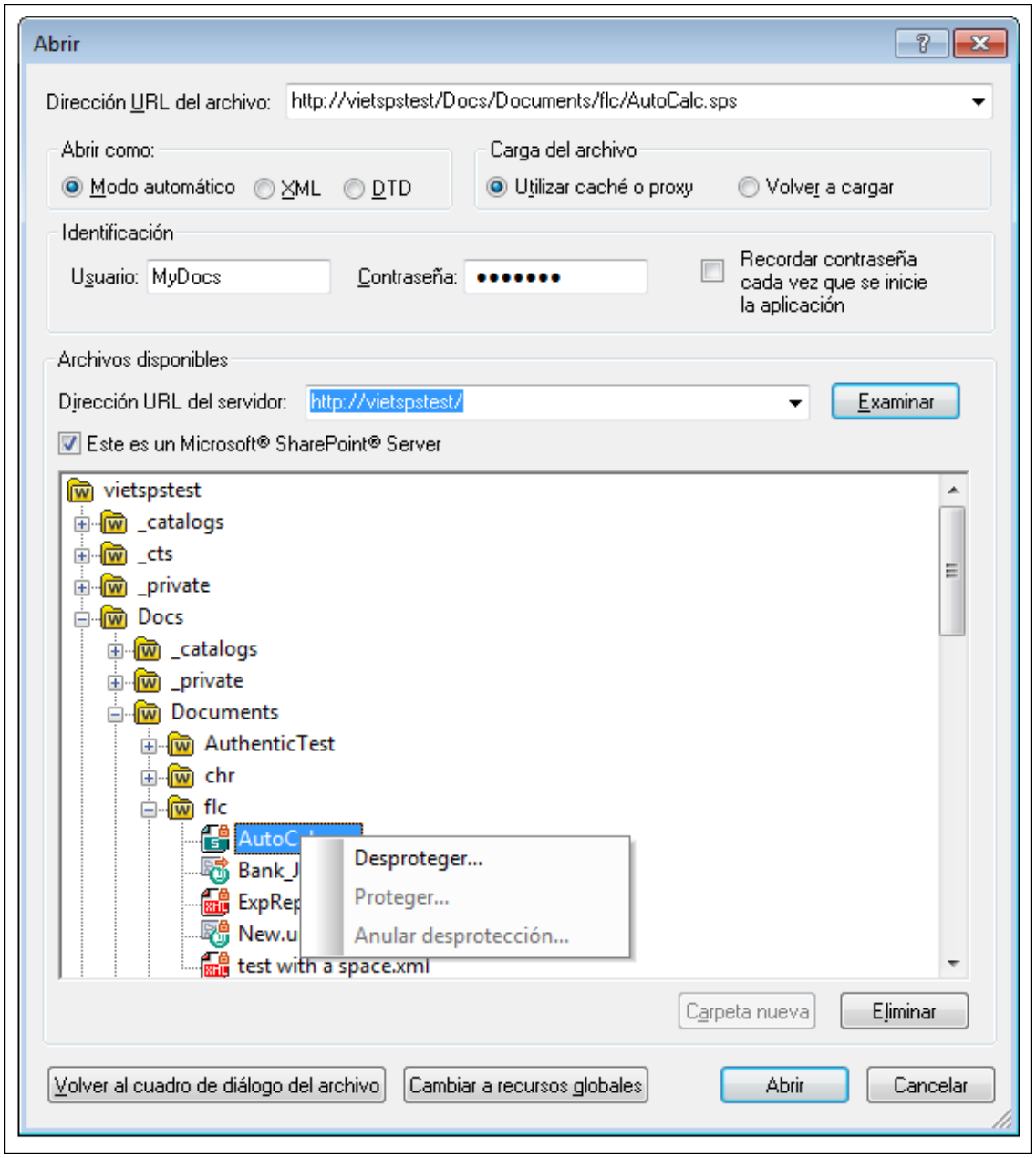
Debe tener en cuenta que:

- La función **Examinar** solamente está disponible en servidores compatibles con WebDAV y servidores Microsoft SharePoint. Los protocolos compatibles son FTP, HTTP y HTTPS.
- Si desea un mayor control durante el proceso de carga del archivo, XMLSpy ofrece la opción de cargar el archivo desde la memoria caché local o desde un servidor proxy (lo cual acelera bastante el proceso si el archivo ya se cargó previamente). Si lo prefiere, también puede volver a cargar el archivo, por ejemplo, si está trabajando con un sistema de base de datos o de publicación electrónica. Para ello está la opción *Volver a cargar*.

▼ Notas sobre Microsoft® SharePoint® Server

Es necesario tener en cuenta algunas características de los archivos residentes en servidores Microsoft® SharePoint®:

- En la estructura de directorios que aparece en el panel *Archivos disponibles* (*imagen siguiente*), los iconos de archivo tienen símbolos que indican el estado de protección de los archivos.



Al hacer clic con el botón derecho en un archivo aparece un menú contextual (*imagen anterior*).

- Estos son los iconos de archivo:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y después proteger.

- Tras desproteger el archivo podrá editarlo en la aplicación de Altova y guardarlo con el comando

 © 2016-2022 Altova GmbH

Archivo | Guardar (Ctrl+S).

- El archivo editado se puede proteger con el menú contextual del cuadro de diálogo "Abrir URL" (*imagen anterior*) o con el menú contextual que aparece al hacer clic con el botón secundario en la pestaña del archivo en la ventana principal de la aplicación (*imagen siguiente*).



- Si otro usuario desprotegió un archivo, dicho archivo no se puede desproteger.
- Si un archivo está desprotegido localmente (por usted), puede deshacer la desprotección con el comando **Deshacer desprotección** del menú contextual. Como resultado se devuelve el archivo al servidor sin ningún cambio.
- Si desprotege un archivo en una aplicación de Altova, no puede desprotegerlo en otra aplicación de Altova. En ese caso los comandos disponibles en la aplicación de Altova son **Proteger** y **Deshacer desprotección**.

▼ Abrir y guardar archivos a través de recursos globales

Para abrir o guardar un archivo a través de recursos globales, haga clic en **Recursos globales**. Aparece un cuadro de diálogo donde puede seleccionar el recurso global que desea utilizar. Estos cuadros de diálogo se describen en la sección Usar recursos globales 1054. Para obtener información sobre esta característica consulte la sección Recursos globales 1042 de este manual.

28.1.3 Volver a cargar

Icono

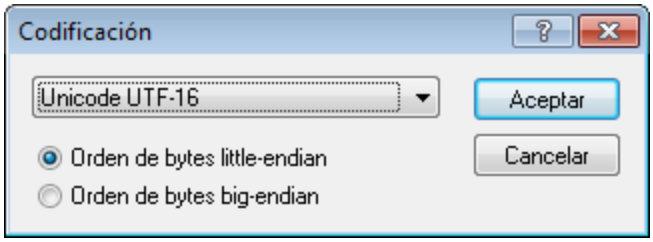
Icono:	

Descripción

Con el comando **Volver a cargar** puede volver a cargar los documentos que están abiertos. Esto puede ser muy útil si alguno de los documentos que están abiertos se modificó fuera de XMLSpy. Si se realiza algún cambio, XMLSpy le pregunta si desea volver a cargar el archivo. Si lo vuelve a cargar, los cambios realizados hasta entonces se perderán. Esta opción se puede cambiar en el cuadro de diálogo "Opciones" (**Herramientas | Opciones** 1584).

28.1.4 Codificación

Con el comando **Codificación** puede (i) ver la codificación actual del documento activo (tanto XML como no XML) y (ii) seleccionar una codificación distinta. El documento activo se guardará con esa codificación la próxima vez que se guarde.



Si cambia la codificación de un documento XML, el atributo de codificación de la declaración XML se ajusta también. Para las codificaciones de caracteres de dos y cuatro bytes (UTF-16, UCS-2 y UCS-4) también puede especificar el orden de bytes que se debe usar para el archivo. Otra manera de cambiar la codificación de un documento XML es editar el atributo `encoding` de la declaración XML del documento.

Para configurar la codificación predeterminada de sus documentos (tanto XML como no XML) utilice la sección 1589 *Codificación* 1589 del cuadro de diálogo "Opciones".

Nota: al guardar un documento, XMLSpy revisa inmediatamente la especificación de codificación y muestra el cuadro de diálogo "Codificación", donde el usuario puede seleccionar la codificación adecuada. Si al guardar el documento se detecta que contiene caracteres que no se pueden representar en la codificación seleccionada, XMLSpy emite una advertencia a tal efecto.

28.1.5 Cerrar, Cerrar todos, Cerrar documentos inactivos

Cerrar

El comando **Cerrar** cierra el documento activo. Si se realizaron cambios en el archivo (si hay un asterisco * después del nombre del archivo en la barra de título), XMLSpy le pregunta si desea guardar el archivo antes de cerrarlo.

Cerrar todos

El comando **Cerrar todos** cierra todos los documentos que están abiertos. Si se realizaron cambios en algún archivo (si hay un asterisco * después del nombre del archivo en la barra de título), XMLSpy le pregunta si desea guardar el archivo antes de cerrarlo.

Cerrar documentos inactivos

El comando **Cerrar documentos inactivos** cierra todos los documentos que están abiertos, excepto el documento que está activo en la ventana principal. Si se realizaron cambios en algún archivo (si hay un asterisco * después del nombre del archivo en la barra de título), XMLSpy le pregunta si desea guardar el archivo antes de cerrarlo.

© 2016-2022 Altova GmbH

28.1.6 Guardar, Guardar como, Guardar todos

Iconos y acceso rápido

Comando	*Icono*	*Acceso rápido*
Guardar		**Ctrl+S**
Guardar todos		

Guardar

Este comando guarda el contenido del documento activo en el archivo desde el que se abrió. Al guardar un documento, se comprueba automáticamente si el documento tiene un formato XML correcto[1335]. El archivo también se valida automáticamente si esta opción está seleccionada en la sección *Archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**[1584]). XMLSpy también revisa la declaración XML y la especificación de codificación[1589] y esta codificación se aplica al documento al guardarlo.

Guardar como

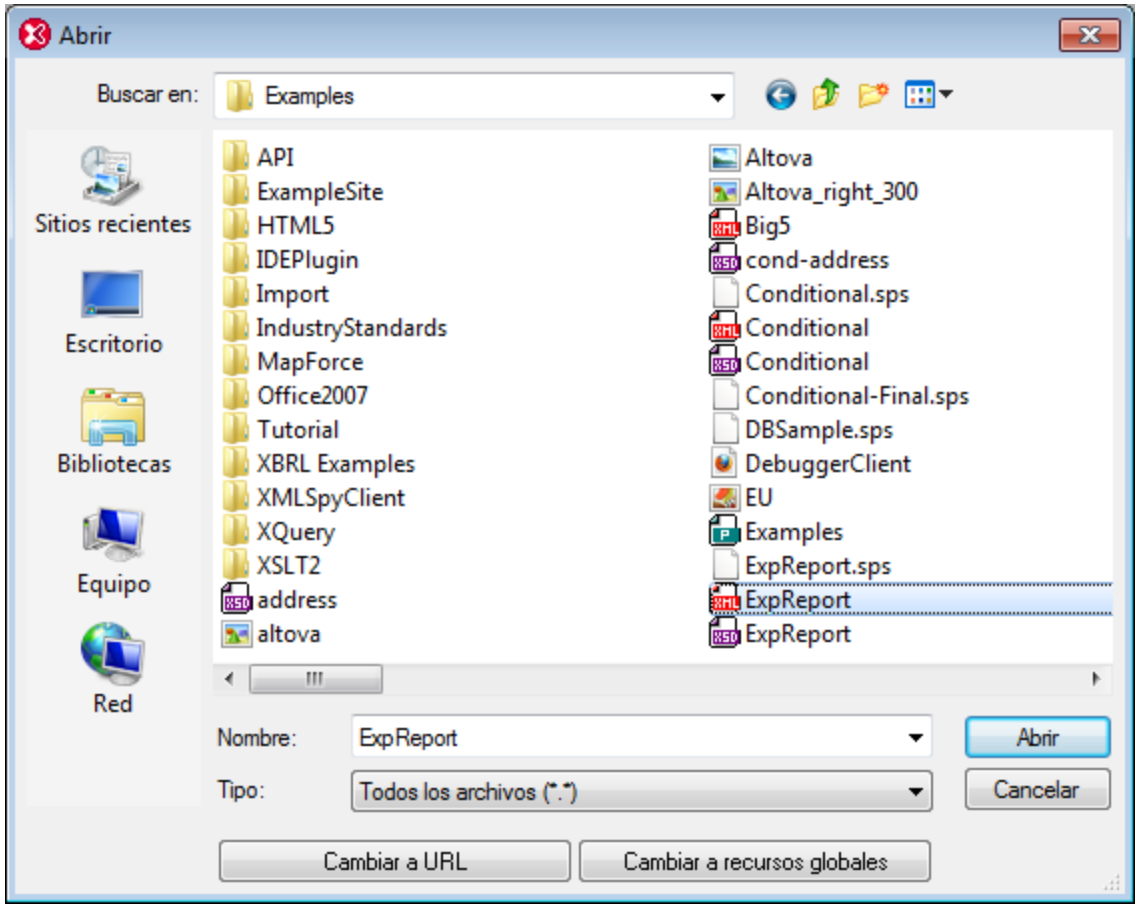
El comando **Guardar como** abre el cuadro de diálogo "Guardar como" de Windows. En este cuadro de diálogo puede introducir el nombre y la ubicación con los que desea guardar el archivo activo. Al igual que ocurre con el comando **Guardar**, la aplicación comprueba el formato XML, revisa la codificación y valida el documento.

Guardar todos

El comando **Guardar todos** guarda todos los cambios realizados en los documentos que están abiertos. Este comando es práctico si está editando varios documentos a la vez. Si alguno de los documentos no se guardó todavía (porque se acaba de crear, por ejemplo) entonces aparece el cuadro de diálogo "Guardar como".

▼ Seleccionar y guardar archivos a través de direcciones URL y recursos globales

En varios cuadros de diálogo "Abrir archivo" y "Guardar archivo" de XMLSpy podrá seleccionar o guardar el archivo a través de una URL o un recurso global (*imagen siguiente*). Haga clic en el botón **Cambiar a URL** o **Cambiar a recursos globales** según corresponda.

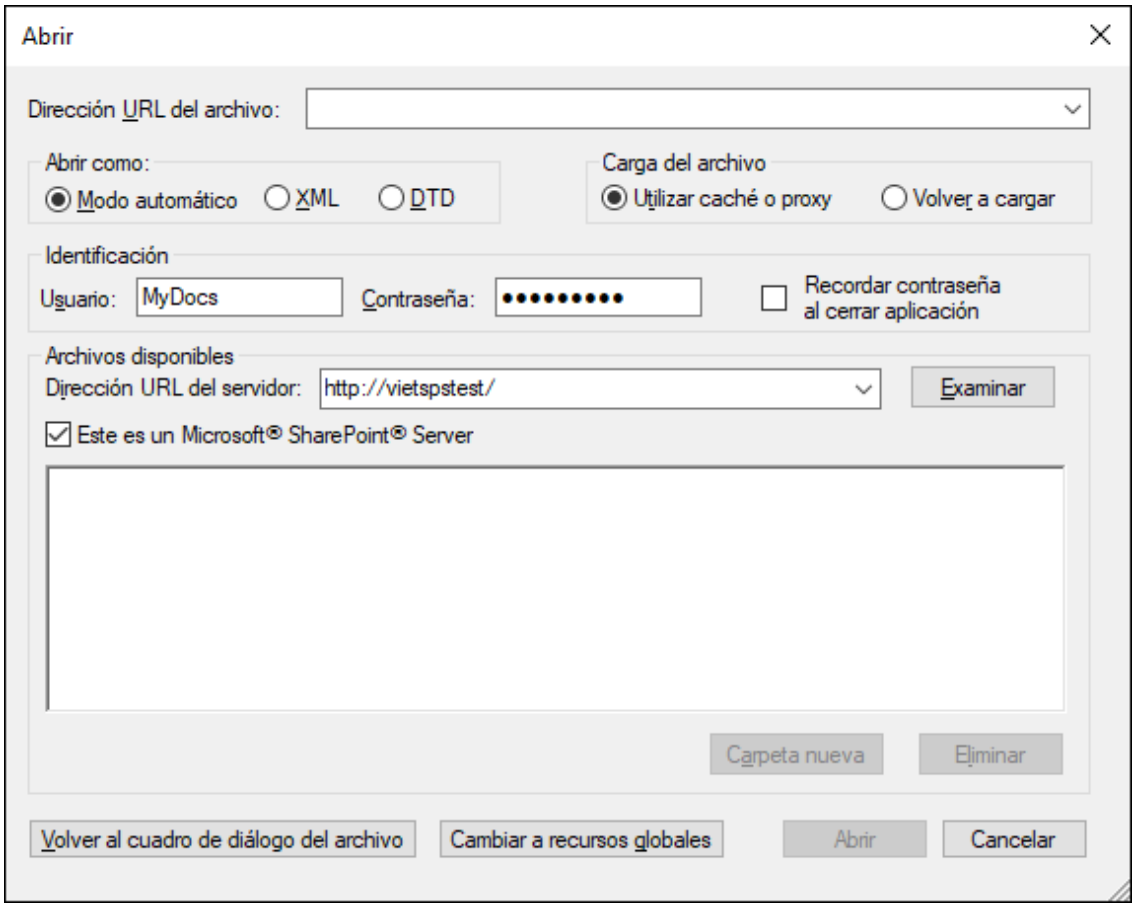


Seleccionar archivos a través de direcciones URL

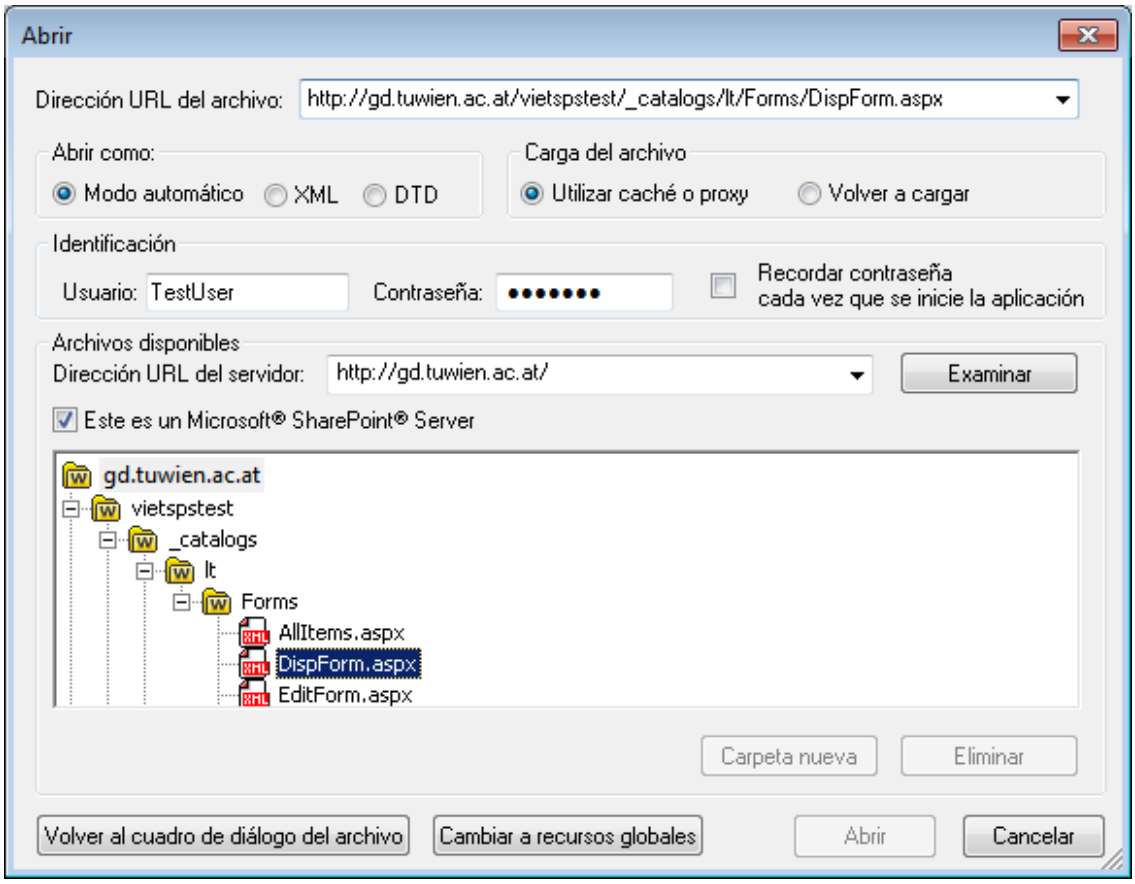
Siga estas instrucciones para seleccionar un archivo a través de una dirección URL (ya sea para abrirlo o para guardarlo):

1. Haga clic en el botón **Cambiar a URL**. El cuadro de diálogo cambia al modo URL (*imagen siguiente*).

© 2016-2022 Altova GmbH



2. Escriba la URL en el campo *Dirección URL del servidor:* (*imagen anterior*). Si el servidor es un servidor Microsoft® SharePoint® Server, marque la casilla *Microsoft® SharePoint® Server*. Consulte la nota sobre este tipo de servidores que aparece más abajo.
3. Si el servidor está protegido con contraseña, escriba el ID de usuario y la contraseña en los campos correspondientes.
4. Haga clic en **Examinar** para navegar por la estructura de directorios del servidor.
5. Busque el archivo que desea cargar y selecciónelo.



El archivo URL aparece en el campo URL del archivo (imagen anterior). En ese momento se habilita el botón Abrir/Guardar.

6. Haga clic en **Abrir** para cargar el archivo o en **Guardar** para guardarlo.

Debe tener en cuenta que:

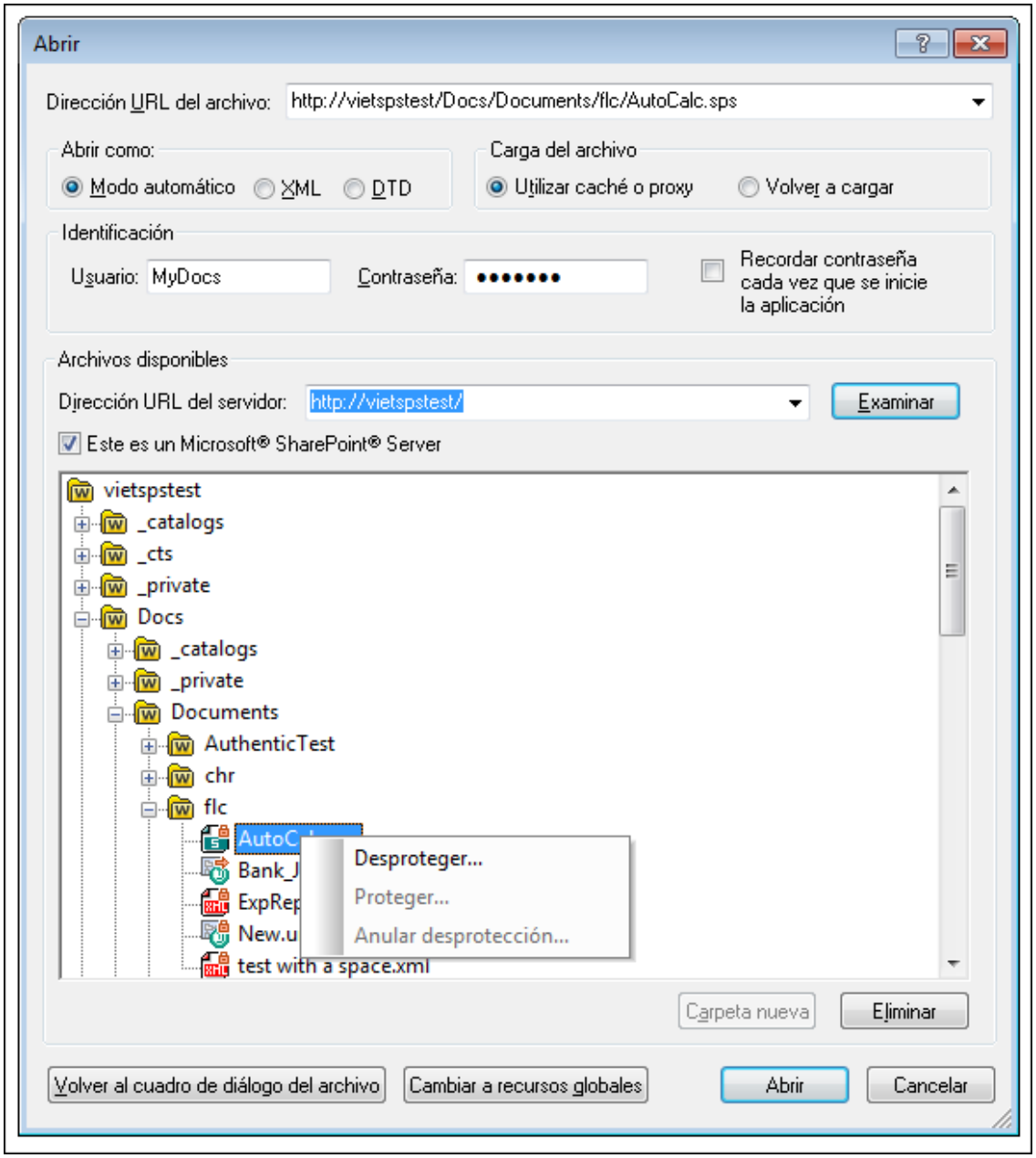
- La función **Examinar** solamente está disponible en servidores compatibles con WebDAV y servidores Microsoft SharePoint. Los protocolos compatibles son FTP, HTTP y HTTPS.
- Si desea un mayor control durante el proceso de carga del archivo, XMLSpy ofrece la opción de cargar el archivo desde la memoria caché local o desde un servidor proxy (lo cual acelera bastante el proceso si el archivo ya se cargó previamente). Si lo prefiere, también puede volver a cargar el archivo, por ejemplo, si está trabajando con un sistema de base de datos o de publicación electrónica. Para ello está la opción *Volver a cargar*.

▼ Notas sobre Microsoft® SharePoint® Server

Es necesario tener en cuenta algunas características de los archivos residentes en servidores Microsoft® SharePoint®:

- En la estructura de directorios que aparece en el panel *Archivos disponibles* (*imagen siguiente*), los iconos de archivo tienen símbolos que indican el estado de protección de los archivos.

 © 2016-2022 Altova GmbH



Al hacer clic con el botón derecho en un archivo aparece un menú contextual (*imagen anterior*).

- Estos son los iconos de archivo:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y después proteger.

- Tras desproteger el archivo podrá editarlo en la aplicación de Altova y guardarlo con el comando

Archivo | Guardar (Ctrl+S).

- El archivo editado se puede proteger con el menú contextual del cuadro de diálogo "Abrir URL" (*imagen anterior*) o con el menú contextual que aparece al hacer clic con el botón secundario en la pestaña del archivo en la ventana principal de la aplicación (*imagen siguiente*).



- Si otro usuario desprotegió un archivo, dicho archivo no se puede desproteger.
- Si un archivo está desprotegido localmente (por usted), puede deshacer la desprotección con el comando **Deshacer desprotección** del menú contextual. Como resultado se devuelve el archivo al servidor sin ningún cambio.
- Si desprotege un archivo en una aplicación de Altova, no puede desprotegerlo en otra aplicación de Altova. En ese caso los comandos disponibles en la aplicación de Altova son **Proteger** y **Deshacer desprotección**.

▼ Abrir y guardar archivos a través de recursos globales

Para abrir o guardar un archivo a través de recursos globales, haga clic en **Recursos globales**. Aparece un cuadro de diálogo donde puede seleccionar el recurso global que desea utilizar. Estos cuadros de diálogo se describen en la sección Usar recursos globales[1054]. Para obtener información sobre esta característica consulte la sección Recursos globales[1042] de este manual.

28.1.7 Enviar por correo electrónico

Icono

Icono:	

Descripción

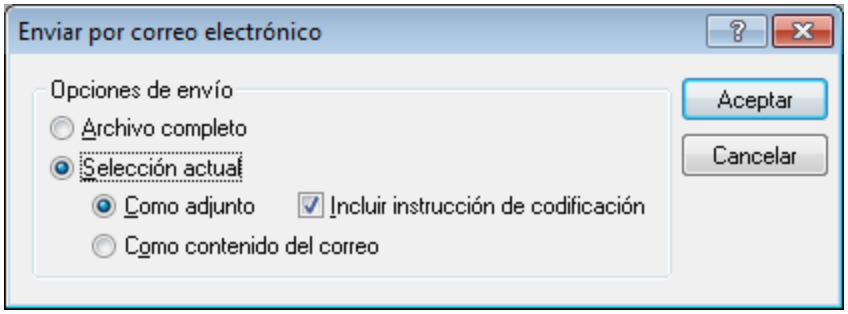
El comando **Enviar por correo electrónico...** sirve para enviar documentos XML o una selección de un documento XML por correo electrónico. El documento o la selección se puede enviar como documento adjunto, como contenido del correo o como enlace, dependiendo del tipo de documento que sea (*ver tabla siguiente*).

¿Qué puedo enviar?	¿Cómo?
Documento XML activo	Como adjunto
Selección en el documento XML activo	Como adjunto o como contenido del correo
Uno o varios archivos de la ventana Proyecto	Como adjunto

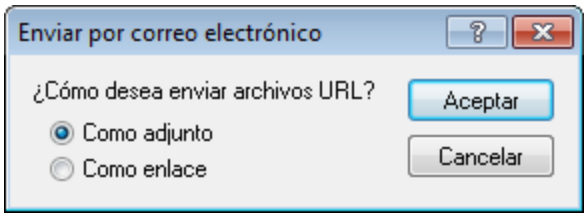
© 2016-2022 Altova GmbH

Una o varias URL de la ventana Proyecto	Como adjunto o como enlace en el correo

Cuando se invoca el comando **Enviar por correo electrónico...** para una selección del documento XML activo, aparece el cuadro de diálogo "Enviar por correo electrónico", donde puede elegir entre varias opciones (*imagen siguiente*). Si no hay texto seleccionado en el documento activo e invoca el comando **Enviar por correo electrónico...** entonces la única opción disponible en el cuadro de diálogo es la opción *Archivo completo* (las demás opciones se deshabilitan).



Puesto que los archivos enviados desde la ventana Proyecto siempre se envían como documentos adjuntos, este cuadro de diálogo no aparece y se abre directamente el correo electrónico que incluye los archivos seleccionados en forma de adjuntos. Las URL de la ventana Proyecto se pueden enviar como adjuntos o como enlaces (*imagen siguiente*). Seleccione una opción y haga clic en **Aceptar**.



28.1.8 Imprimir

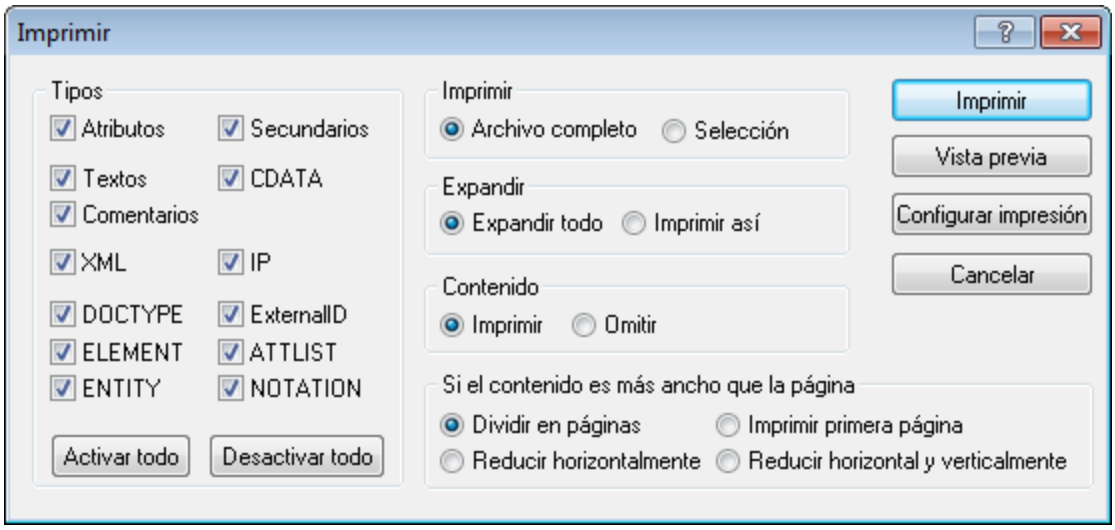
Icono y acceso rápido

Icono:	
Acceso rápido:	**Ctrl+P**

Descripción

El comando **Imprimir** abre el cuadro de diálogo "Imprimir", donde puede configurar la impresión. Después puede imprimir el documento activo tal y como aparece en la vista actual.

Si hace clic en el comando **Imprimir** en la vista Cuadrícula, aparece un cuadro de diálogo (*imagen siguiente*) donde puede definir opciones de impresión para el documento XML. Cuando termine, haga clic en el botón **Imprimir** y aparece el cuadro de diálogo "Imprimir" con opciones para la impresora.



Estas son las opciones de impresión de la vista Cuadrícula (*imagen anterior*):

- *Tipos:* aquí puede elegir qué componentes se imprimen.
- *Imprimir:* aquí puede indicar si se imprime sólo el texto seleccionado o el archivo entero.
- *Expandir:* aquí puede indicar si el documento se imprime tal y como está o si se expanden todos los elementos secundarios.
- *Contenido:* aquí puede indicar si se imprime el contenido de todos los nodos o sólo el nombre de los nodos.
- *Si el contenido es más ancho que la página:* aquí puede elegir qué se hace si el contenido es más ancho que la página. La opción *Imprimir primera página* imprime únicamente la primera página a la izquierda del área de impresión. El área que sobrepase el borde de la página no se imprime. Esta opción es muy útil si la información importante de la vista Cuadrícula está en el lado izquierdo. La opción *Reducir horizontalmente* ajusta el documento al ancho de la página y puede ser necesario imprimir varias páginas. La opción *Reducir horizontal y verticalmente* ajusta el documento a una sola página.
- El botón **Imprimir** imprime el documento en función de las opciones seleccionadas.
- El botón **Vista previa** abre una ventana de vista previa donde puede revisar el aspecto final de la impresión antes de imprimir.
- El botón **Configurar impresión** abre el cuadro de diálogo "Configurar impresión" donde puede ajustar el formato del papel, la orientación y otras opciones de impresión de este trabajo de impresión. Consulte también el apartado sobre el comando **Configurar impresión**[1277].

Nota: recuerde que puede cambiar el ancho de las columnas en la vista Cuadrícula antes de imprimir el documento.

Logotipo del programa

Si compró una licencia permanente, puede cambiar la opción de configuración para que los documentos impresos no incluyan el logotipo del programa, la nota de copyright ni los detalles de registro del software. Esta opción puede configurarse en la sección[1597] *Vista*[1597] del cuadro de diálogo "Opciones"[1597].

© 2016-2022 Altova GmbH

Diseños WSDL y XBRL

Las vistas gráficas de documentos WSDL y XBRL se pueden imprimir tal y como aparecen en pantalla. Para ello use el comando **Imprimir**.

28.1.9 Vista previa de impresión, Configurar impresión

Vista previa de impresión

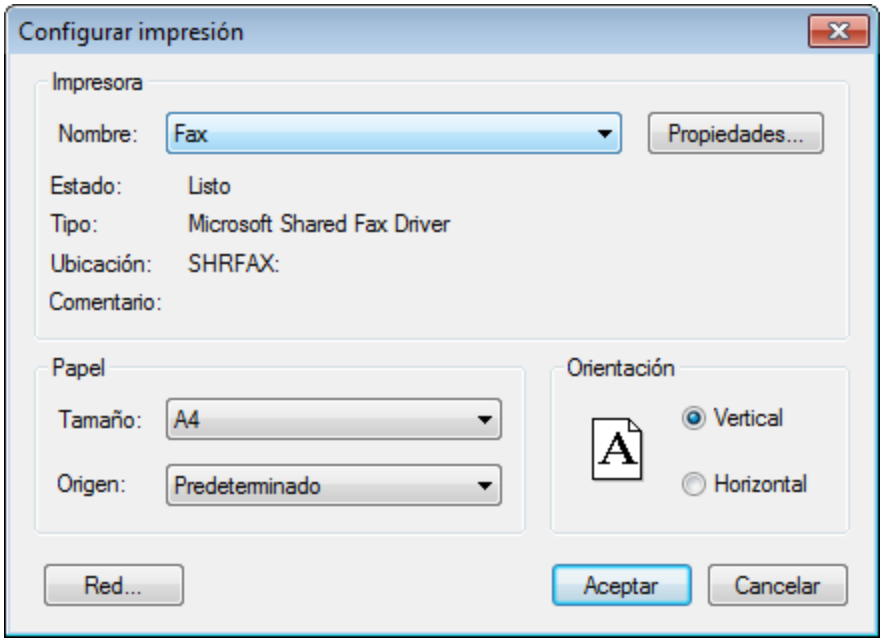
Al hacer clic en el comando **Vista previa de impresión** de la vista Texto, Authentic y Explorador se abre una vista previa de impresión del documento activo. Por el contrario, si hace clic en el comando **Vista previa de impresión** de la vista Cuadrícula, Esquema, WSDL y XBRL, se abre el cuadro de diálogo "Imprimir", donde puede seleccionar opciones de impresión y pulsar el botón **Vista previa** para ver la vista previa de impresión.

El modo *Vista previa de impresión* incluye una barra de herramientas (en la parte superior izquierda) con opciones relacionadas con la impresión y con la vista previa. Puede agrandar o reducir la vista previa con los botones **Acercarse** y **Alejarse** de la barra de herramientas. Si después de pulsar el botón **Acercarse** la vista previa cabe en una sola página, el botón **Una página / Dos páginas** se deshabilita. Si la vista previa ocupa más de una página, puede usar los botones **Siguiente** y **Anterior** para navegar por la vista previa. La barra de herramientas también incluye un botón **Imprimir** y otro para **Cerrar** la ventana de vista previa.

Nota: siga estos pasos para habilitar el uso de colores e imágenes de fondo en la vista previa de impresión: (i) en el menú **Herramientas** de Internet Explorer, haga clic en **Opciones de Internet** y después en la pestaña *Opciones avanzadas*. (ii) En el panel Configuración, marque la casilla *Imprimir colores e imágenes de fondo* de la sección Imprimir. (iii) Haga clic en **Aceptar** para terminar.

Configurar impresión

El comando **Configurar impresión** abre el cuadro de diálogo "Configurar impresión" (*imagen siguiente*), donde puede seleccionar el formato del papel y la orientación de la página, entre otras opciones. Estas opciones se aplican a todos los trabajos de impresión subsiguientes.



28.1.10 Archivos recientes, Salir

Archivos recientes

El menú **Archivo** incluye una lista con los últimos 9 archivos que se abrieron en XMLSpy. El primer archivo de esta lista es el último archivo que se abrió. Para abrir un archivo de la lista, haga clic en su nombre. Para abrir un archivo de la lista usando el teclado, pulse **Alt+A** para abrir el menú **Archivo** y después pulse el número del archivo que desea abrir.

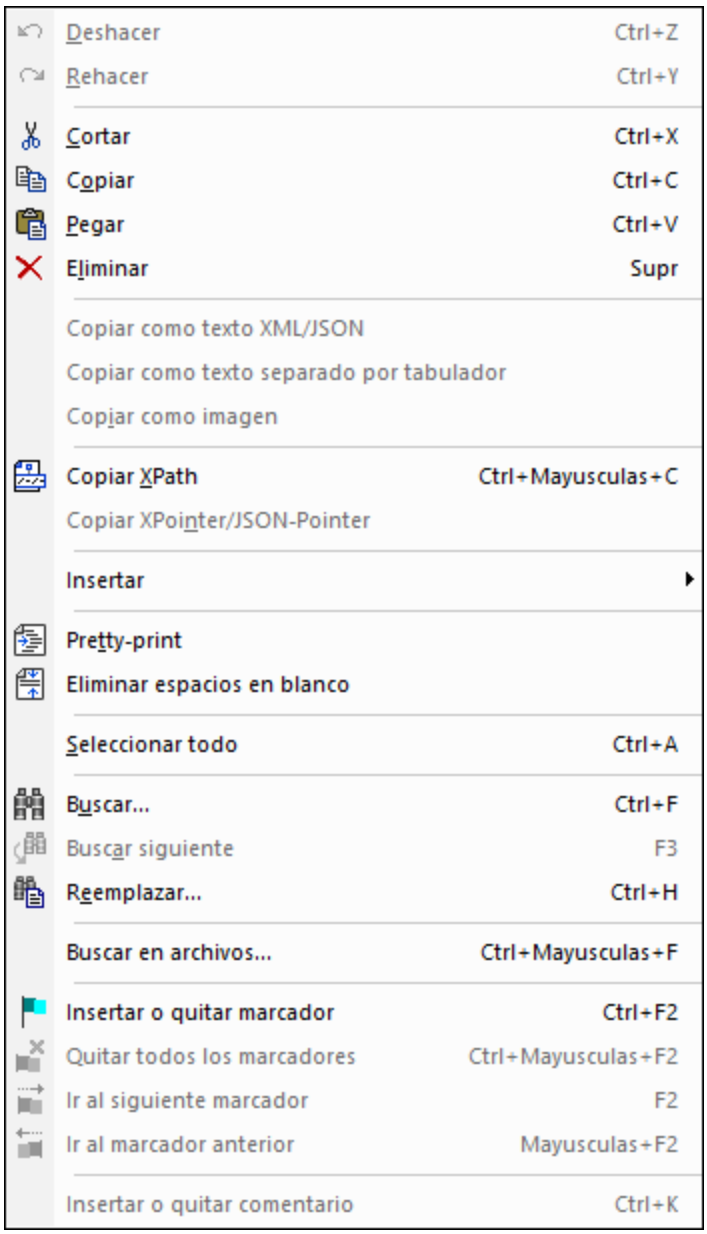
Salir

El comando **Salir** sirve para cerrar XMLSpy. Si hay algún archivo abierto que no se guardó, se le pedirá que guarde los cambios. XMLSpy también guarda los cambios realizados en las opciones de configuración del programa y la información sobre archivos recientes.

© 2016-2022 Altova GmbH

28.2 Menú Edición

Los comandos del menú **Edición** sirven para editar documentos en XMLSpy. Algunos ejemplos son los comandos **Deshacer**[1280], **Rehacer**[1280], **Cortar**[1280], **Copar**[1280], **Pegar**[1280], **Eliminar**[1280], **Seleccionar todo**[1289], **Buscar**[1289], **Buscar siguiente**[1289] y **Reemplazar**[1293].



El menú **Edición** de XMLSpy también ofrece comandos especiales para:

- copiar el texto seleccionado en el portapapeles como texto XML[1281],
- copiar como texto estructurado[1282]
- copiar un selector de XPath para el componente seleccionado[1283] en el portapapeles
- insertar y quitar marcadores y navegar hasta los diferentes marcadores.

28.2.1 Deshacer, Rehacer

Iconos y acceso rápido

Comando	*Icono*	*Acceso rápido*
Deshacer		**Ctrl+Z**
Rehacer		**Ctrl+Y**

Deshacer

El comando **Deshacer** permite eliminar todos los cambios realizados y devolver el archivo a versiones anteriores. Todos los cambios se pueden deshacer uno por uno y no hay un límite de operaciones deshacer. El historial de operaciones deshacer se mantiene tras usar el comando **Guardar**, lo cual permite devolver el archivo al estado en que estaba antes de que guardara los cambios. Es decir puede recorrer el historial de acciones con los comandos **Deshacer** y **Rehacer** (*ver siguiente*).

Rehacer

El comando **Rehacer** permite rehacer las acciones que deshizo con el comando **Deshacer**. Esto significa que puede ir adelante y atrás en el historial de acciones con los comandos **Deshacer** y **Rehacer**.

28.2.2 Cortar, Copiar, Pegar, Eliminar

Iconos y acceso rápido

Comando	*Icono*	*Acceso rápido*
Cortar		**Ctrl+X** o **Mayús+Supr**
Copiar		**Ctrl+C**
Pegar		**Ctrl+V**
Eliminar		**Supr**

Cortar

El comando **Cortar** copia el texto o los componentes seleccionados en el portapapeles y los elimina de su ubicación actual.

Copiar

El comando **Copiar** copia el texto o los componentes seleccionados en el portapapeles. Puede usar este comando para duplicar datos dentro de XMLSpy o para mover datos a otra aplicación.

© 2016-2022 Altova GmbH

Nota: hay dos maneras de copiar datos de la vista Cuadrícula: **Copiar como texto XML**[1281] y **Copiar como texto estructurado**[1282]. El primero copia la selección como texto XML y el segundo copia la selección en forma de tabla. Puede elegir cuál de estos dos métodos se usan cuando se invoca el comando **Copiar** en la vista Cuadrícula. Esto se especifica en la sección *Edición*[1590] del cuadro de diálogo "Opciones"[1590].

Pegar

El comando **Pegar** inserta el contenido del portapapeles en la posición actual del cursor.

Eliminar

El comando **Eliminar** elimina el texto o los componentes seleccionados sin colocarlos en el portapapeles.

28.2.3 Copiar como texto XML/JSON

El comando **Copiar como texto XML/JSON** copia datos XML o JSON de la vista Cuadrícula como texto XML (*imagen siguiente*). Resalte los datos XML de la vista Cuadrícula o de la vista Tabla que desea copiar y seleccione el comando **Copiar como texto XML/JSON**. Los datos se copian en el portapapeles como texto XML (*como en el fragmento de código que aparece más abajo*). Después puede pegar el texto en otro documento.

```
<row>
    <para align="left">
        <bold>Consulte las preguntas frecuentes</bold>
    </para>
    <para>
        <link mode="internal">
            <link_section>support</link_section>
            <link_subsection>faq30</link_subsection>
            <link_text>Preguntas frecuentes sobre XMLSPY</link_text>
        </link>
        <link mode="internal">
            <link_section>support</link_section>
            <link_subsection>faq25</link_subsection>
            <link_text>Preguntas frecuentes sobre XMLSPY</link_text>
        </link>
    </para>
</row>
```

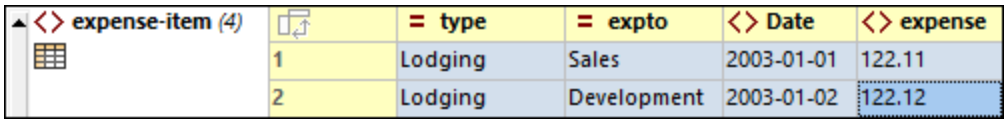
El comando **Copiar como texto XML/JSON** formatea el texto automáticamente en base a las opciones de configuración elegidas para guardar archivos. Estas opciones de configuración se pueden modificar en el grupo de opciones *Guardar archivo* de la sección *Archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).

Si cambia a la vista Texto y copia un fragmento XML con **Ctrl+C** (**Edición | Copiar**), obtendrá el mismo resultado que con el comando **Copiar como texto XML/JSON**. En la vista Cuadrícula también puede usar el comando **Copiar como texto estructurado**[1282] para copiar los datos XML tal y como se presentan en la vista Cuadrícula.

28.2.4 Copiar como texto estructurado

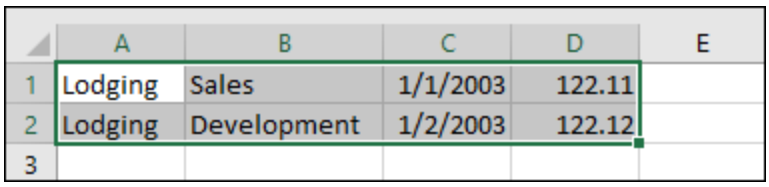
El comando **Copiar como texto estructurado** copia los elementos en el portapapeles tal y como aparecen en pantalla. Este comando es muy práctico si necesita copiar datos de tipo tabla de la vista Cuadrícula/Tabla y pegarlos en otra aplicación que admite datos en forma de tabla (como las aplicaciones de hojas de cálculo).

En la imagen siguiente se copian dos filas de la **vista Tabla** (de la vista Cuadrícula) como texto estructurado.



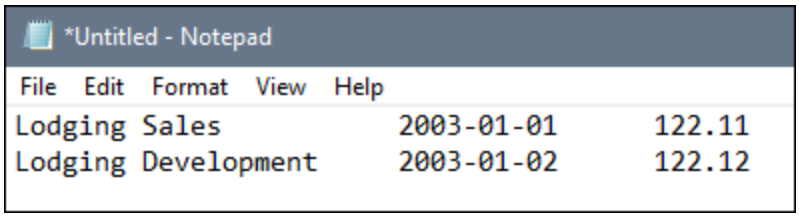
expense-item (4)		type	expto	Date	expense
	1	Lodging	Sales	2003-01-01	122.11
	2	Lodging	Development	2003-01-02	122.12

Las dos imágenes siguientes muestran un documento de Excel y un documento de Notepad donde se pegaron los datos copiados como texto estructurado en el paso anterior.



	A	B	C	D	E
1	Lodging	Sales	1/1/2003	122.11	
2	Lodging	Development	1/2/2003	122.12	
3					

Observe que Excel aplica formato a cada celda según el formato léxico del texto (*imagen anterior*). Por el contrario, Notepad pega todo el texto como una cadena (*imagen siguiente*).



Para más información consulte las secciones Vista Diseño de tabla (XML)[309] and Vista Diseño de tabla (JSON)[692].

28.2.5 Copiar como imagen

El comando Copiar como imagen se habilita solamente si la celda seleccionada en la vista Tabla de la vista Cuadrícula XML[309] o la vista Cuadrícula JSON[692] contienen una imagen. El comando copia la cadena cifrada en base 64 de la imagen seleccionada. Si esta cadena de texto se pega en un documento en el que pueda mostrarse como imagen (por ejemplo a otra celda en la vista Tabla), entonces se verá como tal. De lo contrario se quedará como cadena de texto.

Para más información consulte las secciones Menú contextual de la vista Cuadrícula[323] y Menú contextual de la vista Cuadrícula[706].

© 2016-2022 Altova GmbH

28.2.6 Copiar XPath

El comando **Copiar XPath** está disponible en las vistas Texto y Cuadrícula. Este comando crea una expresión XPath que encuentra el nodo actual en el documento y copia la expresión XPath en el portapapeles. Después puede pegar la expresión XPath en otro archivo (por ejemplo, en un documento XSLT). Todas las expresiones empiezan desde la raíz del documento.

Por ejemplo, si seleccionamos un elemento llamado `LastName` del tercer elemento `Person` del segundo elemento `Company` la expresión XPath que se copia es esta:

```
/Companies/Company[2]/Person[3]/LastName
```

Nota: en la vista Cuadrícula también puede acceder al comando **Copiar XPath** desde el menú contextual.

28.2.7 Copiar XPointer/JSON-Pointer

El comando **Copiar XPointer** está disponible en las vistas Texto y Cuadrícula de los documentos XML y JSON. Este comando crea un XPointer de tipo `element()` para los nodos actuales y lo copia en el portapapeles. Después puede pegar el XPointer en otro archivo (por ejemplo, en el atributo `xpointer` de un elemento `XInclude` de un documento XML).

- *Documentos XML:* este comando crea un XPointer de tipo `element()` para los nodos actuales y lo copia en el portapapeles. Después puede pegar el XPointer en otro archivo. Por ejemplo, el esquema `element()` de XPointer devuelve resultados de tipo `element(/1/3)`. Este XPointer selecciona el tercer secundario del elemento de documento (o elemento raíz).
- *Documentos JSON:* este comando crea un JSON-Pointer para los nodos actuales y lo copia en el portapapeles. Después puede pegar el XPointer en otro archivo. Por ejemplo, para seleccionar un nodo JSON el Pointer JSON **`/Artists/1/Albums/2/Tracks/3/Title`** busca el primer objeto de la matriz de nivel superior **`Artists`**; después busca dentro de ese objeto el segundo objeto de la matriz **`Albums`**; después, dentro de ese objeto busca el tercer objeto de la matriz **`Tracks`**, dentro del cual busca el objeto **`Title`**.

El esquema `element()` de XPointer devuelve resultados de tipo `element(/1/3)`. Este XPointer selecciona el tercer secundario del elemento de documento (o elemento raíz). Tenga en cuenta estos aspectos:

- Los atributos no se pueden representar usando la combinación `element()`. Si selecciona un atributo en la vista Cuadrícula, el comando **Copiar XPointer** se deshabilita. Si selecciona un atributo en la vista Texto, se genera el XPointer del elemento primario de ese atributo.
- No se pueden generar XPointers para varios elementos. Si selecciona varios elementos en la vista Cuadrícula, el comando **Copiar XPointer** se deshabilita. Si selecciona varios elementos en la vista Texto, se genera el XPointer del elemento primario de los elementos seleccionados.

Nota: en la vista Cuadrícula también puede acceder al comando **Copiar XPointer** desde el menú contextual.

Recuerde que:

- Los atributos XML no se pueden representar con el esquema **`element()`**. Si se selecciona un atributo, se genera el XPointer del elemento superior del atributo.

- Si se seleccionan varios elementos XML, se genera el XPointer del primero.
- Si se selecciona un valor JSON, se genera el JSON-Pointer de la clave del valor.
- Si se seleccionan varis nodos JSON, se genera el JSON-Pointer del primero.

Nota: en la vista Cuadrícula también puede acceder al comando **Copiar XPointer/JSON-Pointer** desde el menú contextual.

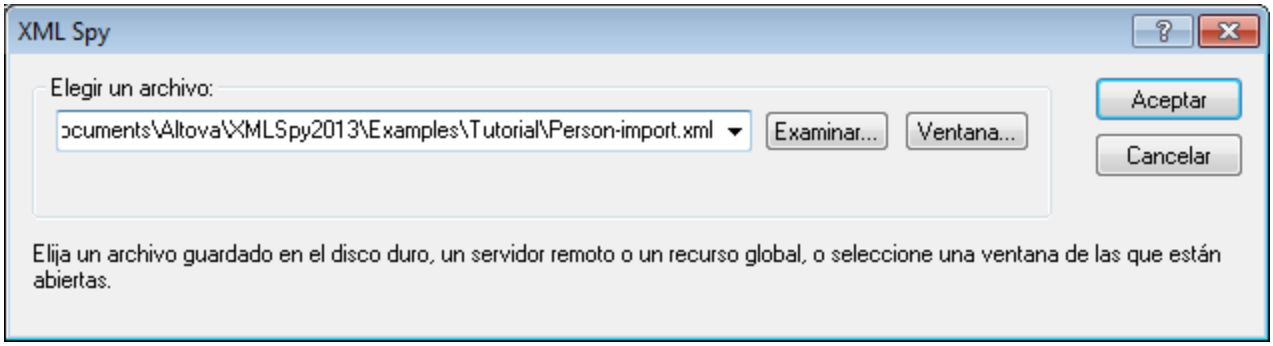
28.2.8 Insertar

Al seleccionar el comando **Insertar** (o al pasar el cursor por este comando) aparece un submenú con estos tres comandos:

- Insertar ruta de acceso del archivo 1284
- Insertar XInclude 1284
- Insertar archivo externo codificado 1287

Insertar ruta de acceso del archivo

El comando **Ruta de acceso del archivo** está disponible en las vistas Texto y Cuadrícula para todos los tipos de documento. Este comando inserta la ruta de acceso de un archivo en la posición del cursor. Al hacer clic en este comando aparece un cuadro de diálogo (*imagen siguiente*) donde puede seleccionar el archivo pertinente.

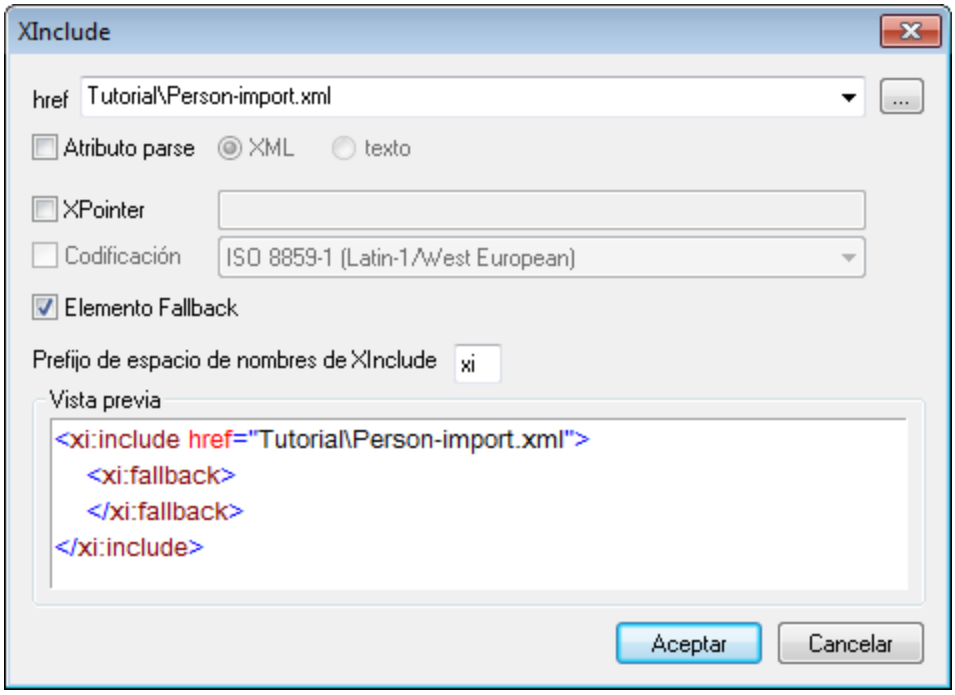


Hay dos maneras de seleccionar el archivo: (i) buscando el archivo, su URL o un recurso global (con el botón **Examinar** 1262) o (ii) seleccionando la ventana en la que está abierto el archivo (con el botón **Ventana**). Seleccione el archivo y haga clic en **Aceptar**. La ruta de acceso del archivo seleccionado se inserta en el documento activo en la posición del cursor.

Insertar XInclude

El comando **XInclude** está disponible en las vistas Texto y Cuadrícula. Este comando sirve para insertar un elemento XInclude nuevo en la posición del cursor (en la vista Texto) o antes del elemento seleccionado (en la vista Texto y Cuadrícula). Si en la vista Cuadrícula selecciona un atributo, el elemento XInclude se inserta después del atributo y antes del primer elemento secundario del elemento primario del atributo. Al seleccionar este comando aparece el cuadro de diálogo XInclude (*imagen siguiente*).

© 2016-2022 Altova GmbH



En el cuadro de texto `href` introduzca el archivo XML que se debe incluir (también puede buscar el archivo haciendo clic en el botón **Examinar** situado a la derecha del cuadro de texto). El nombre del archivo se insertará en el documento XML como valor del atributo `href`. En este diálogo también puede definir los atributos `parse`, `xpointer` y `encoding` del elemento XInclude (`xi:include`), así como el elemento secundario `fallback`. Para ello, seleccione la casilla correspondiente y seleccione/escriba un valor. En el caso del elemento `fallback`, al marcar la casilla se inserta un elemento fallback vacío. El contenido del elemento `fallback` se debe insertar después en la vista de edición.

El atributo `parse` determina si el documento incluido se debe analizar como XML o como texto (`XML` es el valor predeterminado). El atributo `xpointer` identifica a un fragmento concreto del documento que se localiza por medio del atributo `href`. Este es el fragmento que se incluirá. El atributo `encoding` especifica la codificación del documento incluido para que XMLSpy pueda transcodificar este documento (o la parte del documento que se debe incluir) a la codificación del documento de destino. El contenido del elemento secundario `fallback` sustituye al elemento `xi:include` si no se encuentra el documento que se debe incluir.

Por ejemplo, este documento XML usa XInclude para incluir dos documentos XML:

```
<?xml version="1.0" encoding="UTF-16"?>
<AddressBook xsi:schemaLocation="http://www.altova.com/sv/myaddresses AddressBook.xsd"
       xmlns="http://www.altova.com/stylevision/tutorials/myaddresses"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:xi="http://www.w3.org/2001/XInclude">
       <xi:include href="BusinessAddressBook.xml"/>
       <xi:include href="PersonalAddressBook.xml"/>
</AddressBook>
```

Cuando se analice este documento XML, se sustituirán los dos elementos XInclude por los archivos indicados en los atributos `href`.

xml:base
Cuando el validador XML de XMLSpy lee un documento XML y encuentra el elemento `include` en el espacio de nombres XInclude (en adelante `xi:include`), el validador sustituye este elemento (`xi:include`) por el documento XML indicado en el atributo `href` del elemento `xi:include`. El elemento de documento (elemento raíz) del documento XML incluido (o el elemento identificado por un XPointer) se incluye con un atributo `xml:base` a fin de conservar los URI base del elemento incluido. Si el documento XML resultante (que contiene el documento XML o los fragmentos XML incluidos) debe ser válido con respecto a un esquema, entonces el elemento de documento del documento incluido (o el elemento de nivel superior del fragmento XML) debe crearse con un modelo de contenido que admita un atributo `xml:base`. Si, según el esquema, el atributo `xml:base` no está permitido en dicho elemento, entonces el documento resultante no será válido. Para más información sobre cómo definir un atributo `xml:base` en el modelo de contenido de un elemento usando la vista Esquema de XMLSpy, consulte el apartado Prefijos xml: 248 de la sección Vista Esquema del *Manual del usuario*.

XPointers

XMLSpy admite el uso de XPointers en XInclude. Las recomendaciones del W3C relacionadas son XPointer Framework y XPointer element() Scheme. El uso de un XPointer en un elemento XInclude permite incluir una parte concreta del documento XML, en lugar del documento XML entero. Así se usan los XPointer dentro de un elemento XInclude:

```
<xi:include href="PersonalAddressBook.xml" xpointer="element(usa)"/>
<xi:include href="BusinessAddressBook.xml" xpointer="element(/1/1)"/>
<xi:include href="BobsAddressBook.xml" xpointer="element(usa/3/1)"/>
<xi:include href="PatsAddressBook.xml" xpointer="element(usa)element(/1/1)"/>
```

En el modelo element() de XPointer se puede usar un NCName o una secuencia de secundarios dirigida por íntegros.

- En el primer elemento `xi:include` del ejemplo anterior, el atributo `xpointer` usa el modelo element() con un NCName `usa`. Según la recomendación *XPointer Framework*, este NCName identifica al elemento cuyo ID sea `usa`.
- En el segundo elemento `xi:include` del ejemplo anterior, el atributo `xpointer` con el valor `element(/1/1)` identifica en el primer paso al primer elemento secundario de la raíz del documento (si el documento tiene un formato XML correcto, este elemento será el elemento de documento o elemento raíz). En el segundo paso, el atributo `xpointer` identifica al primer elemento secundario del elemento identificado en el paso anterior (en este caso el primer elemento secundario del elemento de documento).
- El atributo `xpointer` del tercer `xi:include` del ejemplo combina un NCName con una secuencia de secundarios. Este XPointer localiza el primer elemento secundario del tercer elemento secundario del elemento cuyo ID sea `usa`.
- Si no está seguro del efecto que tendrá su primer XPointer, puede reforzarlo con otro XPointer tal y como muestra el cuarto `xi:include` del ejemplo: `xpointer="element(usa)element(/1/1)"`. El XPointer de apoyo indica que, en caso de que no exista ningún elemento de ID `usa`, se debe seleccionar el primer elemento secundario del elemento de documento. Puede insertar más XPointers de apoyo si lo desea. Los XPointers pueden estar separados por un espacio en blanco, por ejemplo `xpointer="element(usa)element(addresses/1) element(/1/1)"`.

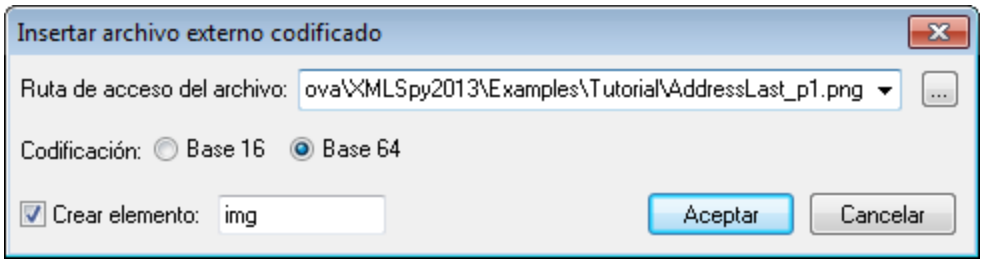
Nota: el contexto del enlace de espacio de nombres no se usa en el modelo element() porque el modelo element() no es compatible con nombres completos (QName).

© 2016-2022 Altova GmbH

Insertar archivo externo codificado

El comando **Archivo externo codificado** está disponible en las vistas Texto y Cuadrícula. Este comando sirve para incluir un archivo externo como texto Base-16 o Base-64 codificado en cualquier posición del documento XML. Es decir, permite incrustar archivos externos en el documento XML.

Al hacer clic en **Insertar | Archivo externo codificado** aparece este cuadro de diálogo:



Puede buscar el archivo externo o escribir su nombre en el recuadro. Es necesario especificar la codificación: *Base-16* o *Base-64*. Si desea incluir el texto codificado dentro de un elemento, marque la casilla *Crear elemento* y escriba el nombre del elemento nuevo en el recuadro. Si no marca la casilla *Crear elemento*, entonces el texto codificado se insertará en la posición del cursor.

Para terminar haga clic en **Aceptar**. El texto codificado del archivo seleccionado se inserta en la posición del cursor y dentro del elemento especificado (si marcó la casilla *Crear elemento*).

Por ejemplo, el fragmento XML que aparece a continuación muestra el texto codificado de un archivo de imagen PNG. En este caso se creó un elemento `img` y en él se insertó el texto codificado.

```
<img ext="png" encoding="xs:base64Binary">
  iVBORw0KGgoAAAANSUhEUgAAABAAAAAQAQMAAAAlPW0iAAAABlBMVEUAAAD/
  //+l2Z/dAAAAM0lEQVR4nGP4/5/h/1+G/58ZDrAz3D/McH8yw83NDDeNGe4U
  g9C9zwz3gVLMDA/A6P9/AFGGFyjOXZtQAAAAAElFTkSuQmCC
</img>
```

28.2.9 Guardar como imagen

Para guardar una cadena cifrada en base 64 en formato imagen haga clic con el botón derecho del ratón en la imagen o su celda y seleccione el comando **Guardar como imagen**. (Tenga en cuenta que la imagen está cifrada como una cadena en base 64 en el archivo activo pero aparece en la vista Cuadrícula como una imagen.) En el cuadro de diálogo que aparece, seleccione la ubicación donde quiere guardar la imagen y asigne un nombre al archivo de imagen. La extensión del archivo (`.png`, `.gif`, `.svg`, etc.) se detecta automáticamente en el cifrado en base 64 y aparece en el cuadro de diálogo **Guardar**.

Para más información consulte las descripciones de la vista Texto y la vista Cuadrícula en las secciones sobre documentos XML 285 y JSON 1351.

28.2.10 Pretty-Print

Icono

Icono:	

Descripción

El comando **Pretty-Print** cambia el formato del documento XML en la vista Texto. Hay dos opciones de formato, dependiendo de si está marcada la casilla *Aplicar sangría* en la sección *Vista* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**):

- Si **está marcada** la casilla *Aplicar sangría,* se cambia el formato del documento para darle un aspecto estructurado, aplicando sangría a los nodos dependiendo de su nivel y de la cantidad de sangría definida por el usuario en el cuadro de diálogo "Configurar la vista Texto" (**Vista | Configurar la vista Texto**). El resultado es un documento más claro y fácil de leer.
- Si **no está marcada** la casilla *Aplicar sangría,* entonces se cambia el formato del documento y cada línea del documento se alinea a la izquierda.

Procedimiento para usar el comando Pretty-Print

Para configurar la vista estructurada con sangría del documento XML:

1. Marque la casilla *Aplicar sangría* en la sección *Vista* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).
2. Defina el tamaño de la sangría en el cuadro de diálogo Configurar la vista Texto (**Vista | Configurar la vista Texto**).
3. Indique a qué elementos no se les debe dar formato de salida (sangría) en la sección *Archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).
4. Haga clic en el comando **Pretty-Print** (el comando que nos ocupa).

Para cambiar el formato del documento de modo que todas las líneas estén alineadas a la izquierda, desactive la casilla *Aplicar sangría*.

Es importante tener en cuenta que:

- Para que funcione este comando es necesario que el documento XML tenga un formato XML correcto.
- El formato pretty-print añade espacios o tabulaciones al documento al guardarlo.
- Si está marcada la casilla *Aplicar sangría* (**Herramientas | Opciones | Vista | Aplicar sangría**) y cambia de la vista Texto a la vista Cuadrícula y después vuelve a la vista Texto, el documento tendrá formato pretty-print automáticamente. No hace falta seleccionar otra vez el comando **Pretty-Print**.
- Para quitar todos los espacios en blancos (líneas nuevas y sangría) creados con el comando **Pretty-Print**, use el comando **Eliminar espacios en blanco**.

© 2016-2022 Altova GmbH

28.2.11 Eliminar espacios en blanco

Icono

Icono:	

Descripción

El comando **Eliminar espacios en blanco** quita del documento todos los espacios en blanco. Esto puede ayudarle a reducir el tamaño del documento y puede ser muy práctico para eliminar los espacios en blanco que genera el comando **Pretty-Print**[1288] (para añadir líneas nuevas y aplicar sangría al documento).

28.2.12 Seleccionar todo

El comando **Seleccionar todo** (**Ctrl+A**) sirve para seleccionar el contenido de todo el documento.

28.2.13 Buscar, Buscar siguiente

Iconos y acceso rápido

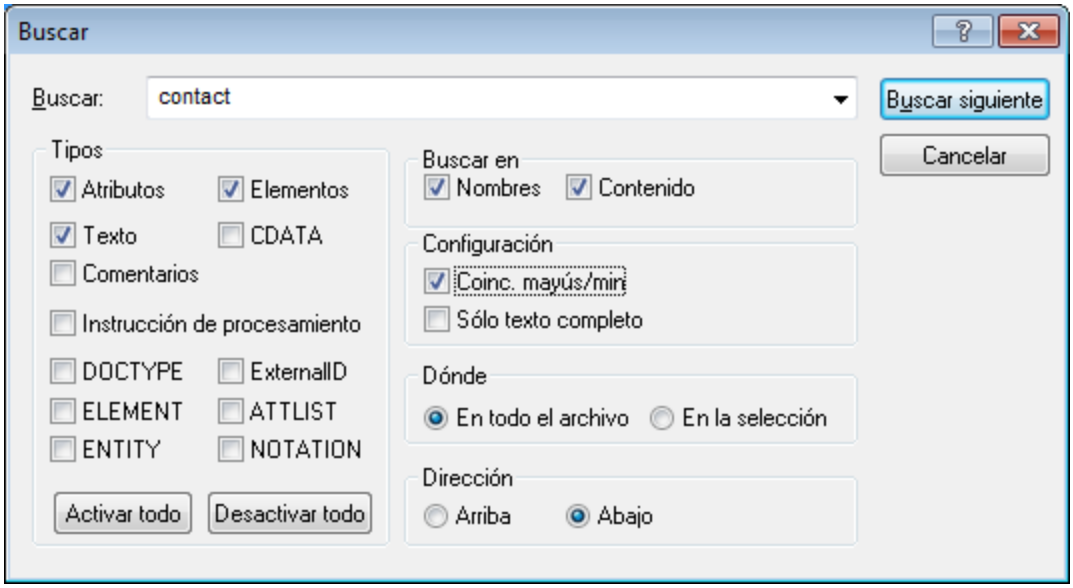
Comando	*Icono*	*Acceso rápido*
Buscar		**Ctrl+F**
Buscar siguiente		**F3**

Buscar

El comando **Buscar** abre el cuadro de diálogo "Buscar", donde puede indicar qué cadena de texto desea buscar y configurar la búsqueda. Dependiendo de la vista que esté usando, el cuadro de diálogo "Buscar" muestra unas opciones u otras. Escriba el término de búsqueda en el campo *Buscar* o seleccione una opción del cuadro combinado, que recuerda los 10 términos de búsqueda más recientes. Después configure las opciones de búsqueda.

Los comandos **Buscar** y **Buscar siguiente** también sirven para buscar nombres de archivo y carpeta cuando en la ventana Proyecto está seleccionado un proyecto.

Vista Cuadrícula
Este es el cuadro de diálogo "Buscar" de la vista Cuadrícula. Las opciones de este cuadro de diálogo se describen a continuación.



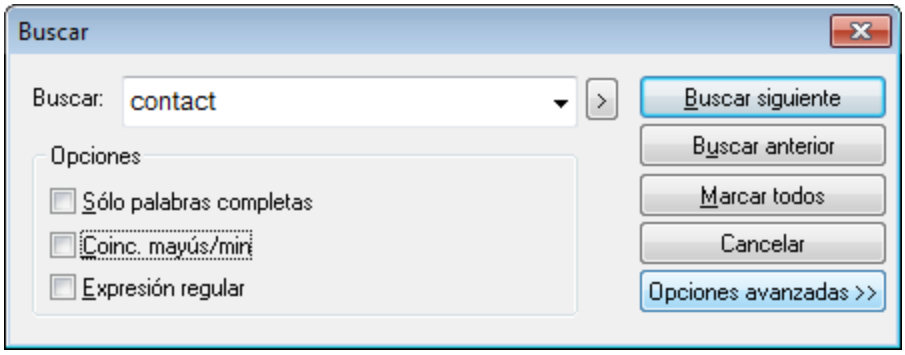
Marque las casillas o seleccione los botones de opción necesarios para su búsqueda.

- *Tipos*: aquí puede seleccionar los nodos o componentes del documento XML que desea incluir en la búsqueda. Así puede omitir ciertos tipos de nodos si lo desea para agilizar la búsqueda. Con el botón **Activar todo** se marcan todos los tipos, mientras que el botón **Desactivar todo** desactiva todos los tipos.
- *Buscar en*: aquí puede definir si la cadena de texto se debe buscar en el contenido de los nodos, en el nombre de los nodos o en ambos.
- *Configuración*: aquí puede definir si los resultados deben coincidir palabra por palabra con todo el término de búsqueda y si deben coincidir en el uso de las mayúsculas y minúsculas.
- *Dónde*: aquí puede definir el ámbito de la búsqueda (el archivo entero o el texto seleccionado solamente).
- *Dirección*: aquí puede especificar la dirección de la búsqueda.

Vista Texto

Para aprender a usar la función de búsqueda en vistas de texto editables, consulte la sección Vista Texto[157]. A continuación describimos el cuadro de diálogo "Buscar" de las vistas de texto no editables (p. ej. de la vista de la ventana XPath/XQuery[127]).

© 2016-2022 Altova GmbH



Estas son las opciones que puede seleccionar en este cuadro de diálogo:

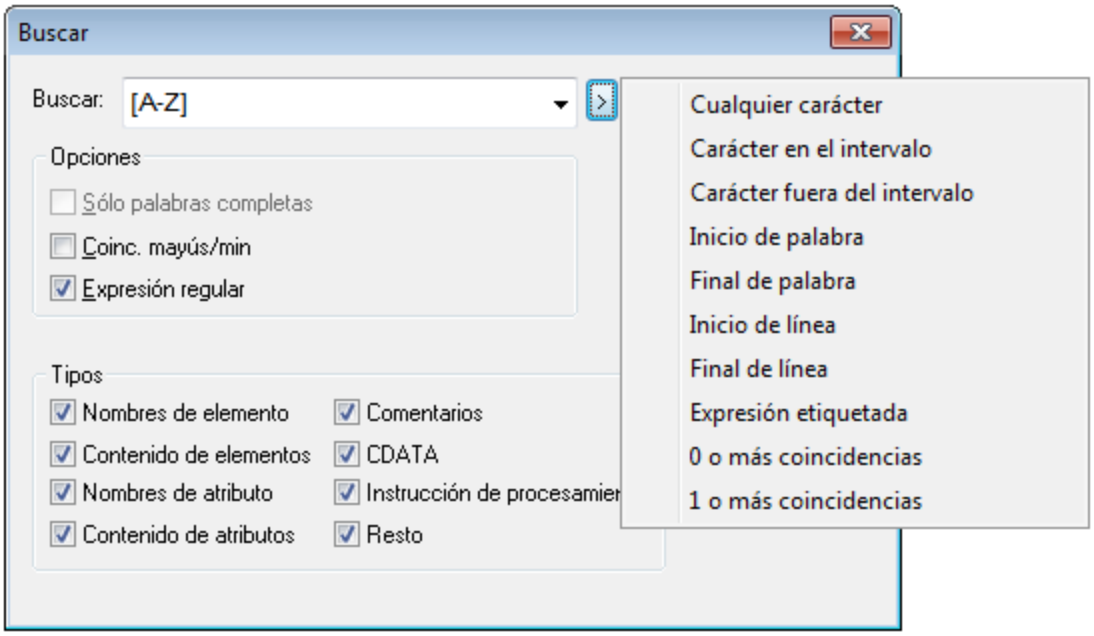
- *Sólo palabras completas*. Sólo se buscan palabras que coincidan con todo el término de búsqueda. Por ejemplo, para el término de búsqueda `par`, sólo se encontraría la palabra `par`. Sin embargo, si no se marca esta casilla, el término de búsqueda `par` encontraría las palabras `par`, `pareja` y `compartir`, por ejemplo.
- *Coincidencia de mayúsculas/minúsculas*. Los resultados tienen que coincidir con el término de búsqueda en el uso de mayúsculas y minúsculas (`Carta` no es lo mismo que `carta`).
- *Expresión regular*. Busca el texto especificado mediante la expresión regular. Para más información consulte el apartado *Expresiones regulares*[1291].

Tenga en cuenta que:

- El cuadro de diálogo "Buscar" es un cuadro de diálogo *no modal*. Esto significa que puede seguir abierto mientras usa la vista Texto. Si pulsa la tecla **Entrar** con el diálogo abierto, el diálogo se cierra. Si selecciona texto antes de abrir el cuadro de diálogo, el texto seleccionado se inserta automáticamente el cuadro de texto *Buscar*.
- Después de cerrar el cuadro de diálogo Buscar, pulse **F3** para repetir la búsqueda actual hacia adelante o **Mayús+F3** para repetir la búsqueda hacia atrás.
- El botón de expansión situado a la derecha del cuadro combinado *Buscar* (el botón **>**) abre un menú con opciones para crear expresiones regulares[1291].

Expresiones regulares

Puede usar expresiones regulares para limitar sus criterios de búsqueda. El cuadro de diálogo incluye una lista emergente con opciones para crear expresiones regulares. Para abrir la lista, haga clic en el botón de expansión **>** situado a la derecha del campo de entrada *Buscar*.



Haga clic en la descripción de la expresión que desea usar y esta se insertará en el campo de entrada. Los caracteres utilizados en la sintaxis de expresiones regulares son:

.	Cualquier carácter. Es un comodín para un solo carácter.
(	El inicio de una región en la que se realizará la búsqueda.
)	El final de la región en la que se realizará la búsqueda.
\n	En una expresión de búsqueda o reemplazo, indica el texto coincidente con la n-ésima expresión etiquetada, siendo n un número del 1 al 9. Por ejemplo, si el término de búsqueda es `Fred([1-9])XXX` y el término de reemplazo es `Sam\1YYY`, el texto Fred2XXX se reemplazará con Sam2YYY.
\<	Inicio de palabra.
\>	Final de palabra.
\x	Encuentra el carácter `x` que sigue a la barra inversa, cuando el carácter `x` es un carácter que se utiliza en la sintaxis de expresiones regulares. Por ejemplo, `\[` encuentra el carácter `[` y no el inicio de un conjunto de caracteres.
[...]	Conjunto de caracteres. Por ejemplo, `[abc]` encuentra los caracteres `a`, `b` o `c`. También puede usar intervalos como `[a-z]` para buscar cualquier carácter en minúsculas.
[^...]	Un carácter cualquiera no perteneciente al conjunto. Por ejemplo, `[^A-Za-z]` encuentra cualquier carácter excepto caracteres alfabéticos en mayúsculas o minúsculas.
^	Inicio de línea (a no ser que se use dentro de un conjunto de caracteres, ver fila anterior).
$	Final de línea. Por ejemplo, `A+$` encuentra una A o más de una A que estén al final de una línea.
*	0 veces o más. Por ejemplo, `Sa*m` encuentra `Sm`, `Sam`, `Saam`, `Saaam`, etc.
+	1 o más veces. Por ejemplo `Sa+m` encuentra `Sam`, `Saam`, `Saaam`, etc

Nota: en el campo *Reemplazar por*[1293] no se pueden usar expresiones regulares.

 © 2016-2022 Altova GmbH

Buscar siguiente

El comando **Buscar siguiente** repite la última búsqueda realizada con el comando **Buscar** y busca la siguiente instancia del término de búsqueda.

28.2.14 Reemplazar

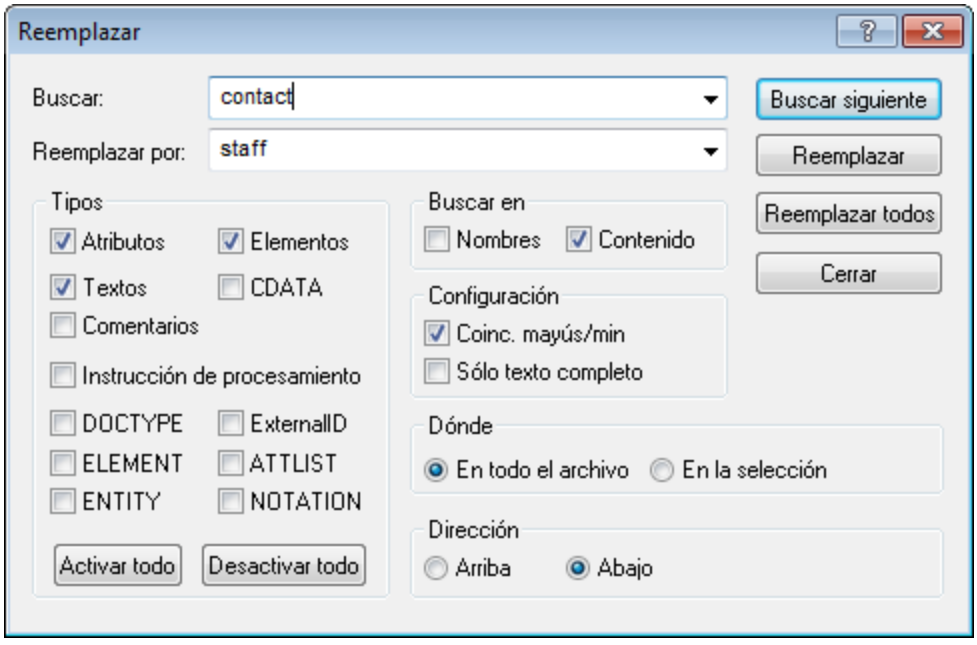
Icono y acceso rápido

Comando	*Icono*	*Acceso rápido*
Reemplazar		**Ctrl+H**

Descripción

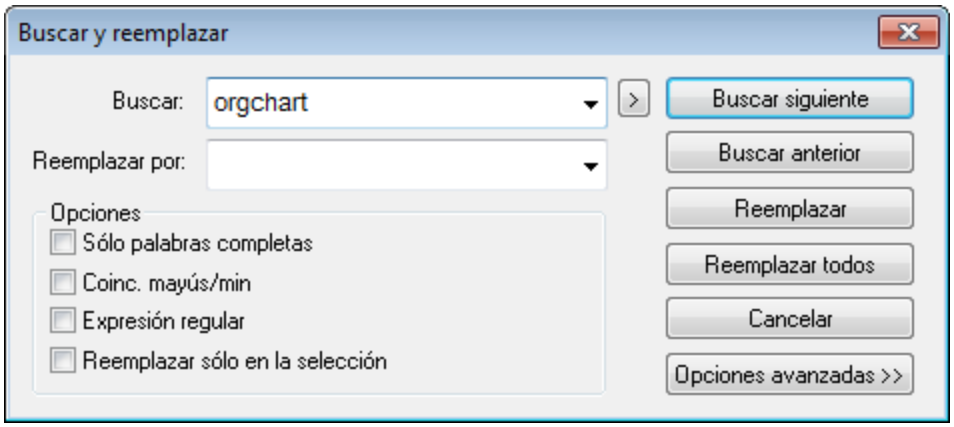
Con el comando **Reemplazar** puede buscar y reemplazar una cadena de texto con otra. Este comando tiene las mismas características que el comando **Buscar**[1289]. Dependiendo de la vista que esté usando, el cuadro de diálogo Reemplazar muestra unas opciones u otras. Puede reemplazar los términos encontrados uno por uno o usar el botón **Reemplazar todos** para realizar una operación global de búsqueda y reemplazo.

Vista Cuadrícula
En la vista Cuadrícula, el comando **Reemplazar** abre este cuadro de diálogo:



Las opciones de este cuadro de diálogo se describen en el apartado dedicado al comando **Buscar**[1289].

Vista Texto
Para aprender a usar la función de reemplazo en vistas de texto editables, consulte la sección Vista Texto[157]. La función de búsqueda en vistas de texto no editables (p. ej. en la vista de la ventana XPath/XQuery[127]) se describe a continuación. Al seleccionar el comando **Reemplazar** en la vista Texto aparece el cuadro de diálogo "Buscar y reemplazar". Este diálogo incluye las mismas opciones que el diálogo Buscar[1289]. La opción *Reemplazar sólo en la selección* realiza la operación de búsqueda y reemplazo únicamente en la parte del texto que está seleccionada en la vista.

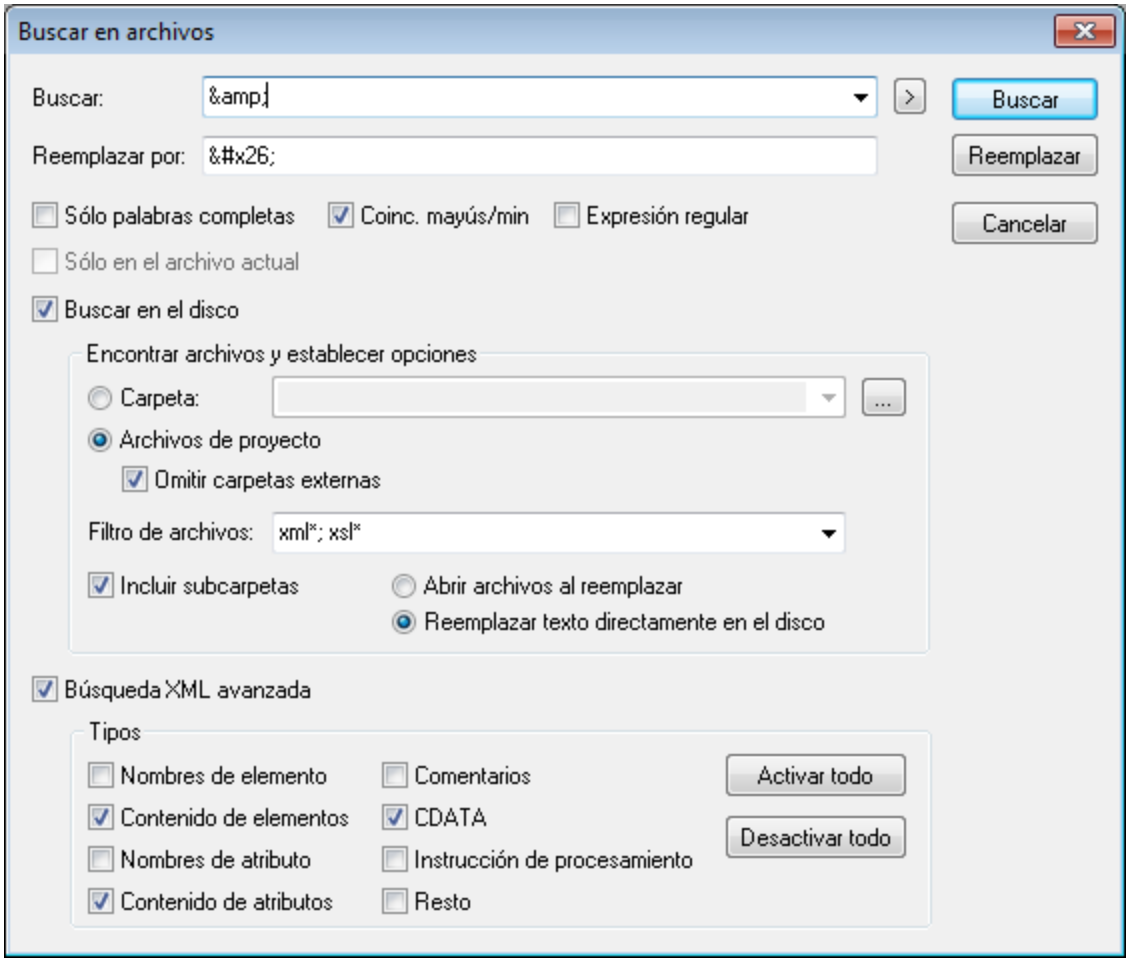


Nota: cuando se usa el comando **Reemplazar todos**, cada reemplazo se registra como una sola operación. De este modo, se pueden deshacer cada uno de los reemplazos efectuados por el comando **Reemplazar todos**.

28.2.15 Buscar en archivos

Con el comando **Buscar en archivos** puede buscar y reemplazar texto en un gran número de archivos con gran rapidez. Al hacer clic en este comando aparece el cuadro de diálogo "Buscar en archivos" (*ver imagen*). El comando **Buscar en archivos** difiere del comando **Buscar** porque busca la cadena de texto en todas las ubicaciones indicadas por el usuario de una sola vez y reemplaza el texto en todas las ubicaciones inmediatamente y de una sola vez. La aplicación informa de los resultados en la ventana Buscar en archivos[129]. Por el contrario, con el comando **Buscar** el usuario introduce la cadena de búsqueda y recorre el documento activo deteniéndose en cada resultado.

© 2016-2022 Altova GmbH



Criterios de búsqueda

Hay dos criterios de búsqueda principales: (i) qué se busca y (ii) dónde se busca. Para más información sobre cómo crear cadenas de búsqueda, consulte la descripción del comando **Buscar**[1289]. Si el texto introducido en el cuadro *Buscar:* es una expresión regular, entonces debe marcarse la casilla *Expresión regular*. Si hace clic en el botón de la flecha (situado a la derecha del campo de texto *Buscar*) se abre el ayudante de entrada de expresiones regulares. Para más información sobre el uso de expresiones regulares en las búsquedas, consulte la descripción del comando **Buscar**[1289].
Para indicar en qué tipos de nodo y en qué partes del documento XML se debe realizar la búsqueda, marque la casilla *Búsqueda XML avanzada* y después seleccione los tipos de nodo en los que desea buscar.

También puede indicar en qué archivos se debe buscar: puede marcar la casilla *Sólo en el archivo actual* o la casilla *Buscar en el disco*. Si decide buscar en el disco, puede realizar la búsqueda en una carpeta o en un proyecto[1298] (primero debe marcar la casilla *Buscar en el disco*). Si decide buscar en los archivos de proyecto, puede omitir las carpetas externas añadidas a los proyectos. Puede limitar los tipos de archivo en los que se realiza la búsqueda introduciendo la extensión del archivo y un asterisco (por ejemplo `xml*` o `xsl*`). El separador entre las extensiones de archivo puede ser una coma o un punto y coma (por ejemplo `xml*;xsl*`). El asterisco también se puede usar como comodín.

Los resultados de la búsqueda realizada en todas las ubicaciones aparecen en la ventana de resultados Buscar en archivos 129. Al hacer clic en uno de los resultados, el archivo correspondiente se abre en la vista Texto con el nodo de búsqueda resaltado.

Reemplazar

Lo más importante que debe tener en cuenta es que al hacer clic en el botón **Reemplazar** se reemplazan todas las instancias de la cadena de búsqueda con la cadena de reemplazo. Si se marca la opción *Abrir archivos al reemplazar* en el cuadro de diálogo "Buscar en archivos", entonces los archivos se abren en la vista Texto. De lo contrario, la operación de reemplazo se realiza en total silencio. Todas las instancias reemplazadas aparecen en una lista en la ventana de resultados Buscar en archivos 129. Al hacer clic en uno de los resultados de la lista, el archivo correspondiente se abre en la vista Texto con el nodo de reemplazo resaltado.

Nota: en el campo *Reemplazar por:* del cuadro de diálogo no se pueden usar expresiones regulares.

28.2.16 Marcadores

Iconos y acceso rápido

Comando	*Icono*	*Acceso rápido*
Insertar o quitar marcador		**Ctrl+F2**
Quitar todos los marcadores		**Ctrl+Mayús+F2**
Ir al siguiente marcador		**F2**
Ir al marcador anterior		**Mayús+F2**

Insertar o quitar marcador

El comando **Insertar o quitar marcador** inserta un marcador en la posición del cursor o quita el marcador si el cursor se encuentra en una línea en la que se insertó un marcador previamente. Este comando está disponible solamente en la vista Texto.

Hay dos maneras de ver en qué líneas se insertó un marcador:

- Si el margen de marcadores está habilitado, a la izquierda de la línea que contiene el marcador aparece una esfera de color aguamarina.
- Si el margen de marcadores no está habilitado, entonces se resalta toda la línea que contiene el marcador.

Con la tecla **F2** puede recorrer el documento de marcador en marcador.

© 2016-2022 Altova GmbH

Quitar todos los marcadores

El comando **Quitar todos los marcadores** elimina todos los marcadores del documento. Este comando está disponible solamente en la vista Texto. Tenga en cuenta que el comando **Deshacer** no deshace el efecto de este comando.

Ir al siguiente marcador

El comando **Ir al siguiente marcador** coloca el cursor de texto al principio de la línea en la que está el siguiente marcador. Este comando está disponible en la vista Texto solamente.

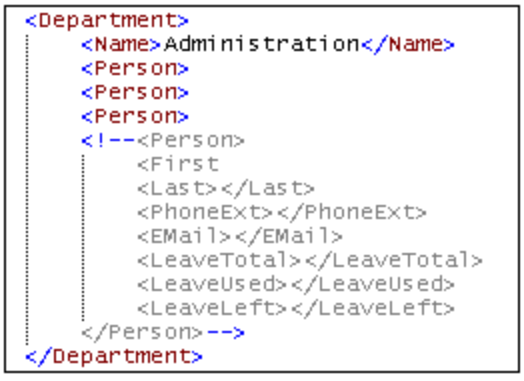
Ir al marcador anterior

El comando **Ir al marcador anterior** coloca el cursor de texto al principio de la línea en la que está el marcador anterior. Este comando está disponible en la vista Texto solamente.

28.2.17 Insertar o quitar comentario

El comando **Insertar o quitar comentario** está disponible en la vista Texto y se usa para convertir fragmentos de texto XML en comentarios o quitar la marca de comentario de un fragmento de texto XML. Para convertir un fragmento de texto XML en comentario basta con insertar los delimitadores de comentario XML `<!--` y `-->` al principio y al final del texto respectivamente. En XMLSpy estos delimitadores de comentario se pueden insertar alrededor del texto seleccionado con el comando de menú **Insertar o quitar comentario**.

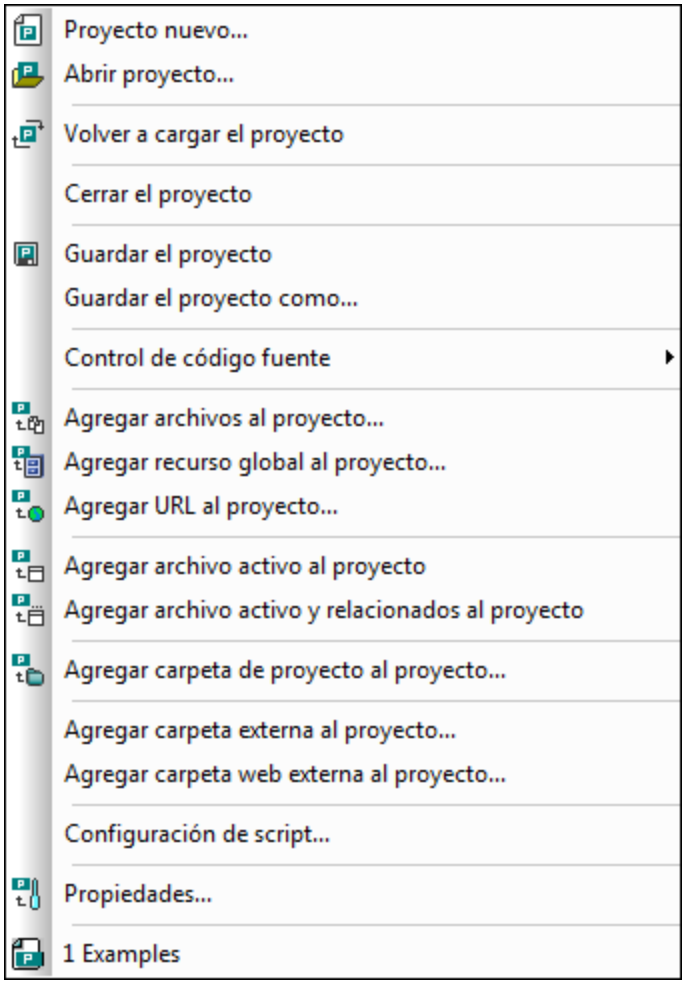
Para convertir un bloque de texto en comentario, seleccione el texto y después haga clic en el comando **Insertar o quitar comentario** (bien en el menú **Edición** o bien en el menú contextual que aparece al hacer clic con el botón derecho en el texto seleccionado). El texto que lleva la marca de comentario ahora aparece atenuado (*ver imagen siguiente*).



Para quitar la marca de comentario de un bloque de texto, ponga el cursor en el bloque comentado y seleccione el comando **Insertar o quitar comentario** (bien en el menú **Edición** o bien en el menú contextual que aparece al hacer clic con el botón derecho en el texto que forma parte del comentario). Los delimitadores de comentario se eliminan y el texto deja de estar atenuado.

28.3 Menú Proyecto

XMLSpy usa una estructura en forma de árbol para gestionar y organizar varios archivos o direcciones URL en proyectos XML. Los archivos[1317] y direcciones URL[1317] se pueden organizar por carpetas[1318] según su extensión de archivo o cualquier otro criterio, para que pueda estructurar su proyecto y realizar operaciones por lotes con total facilidad.



Nota importante: la mayoría de los comandos para trabajar con proyectos también están disponibles en el menú contextual que aparece al hacer clic con el botón derecho en cualquier elemento de la ventana Proyecto.

Rutas de archivo relativas y absolutas

Los proyectos se guardan como archivo de proyecto y tienen la extensión `.spp`. Se trata, de hecho, de documentos XML que se pueden editar como si fueran archivos XML normales y corrientes. En el archivo de proyecto se usan rutas de archivo absolutas para archivos y carpetas que están al mismo nivel o en un nivel superior, mientras que para archivos y carpetas de la carpeta actual o subcarpetas se usan rutas de archivo relativas. Por ejemplo, si la estructura de directorios de su proyecto es como esta:

```
|-Carpeta1
```


© 2016-2022 Altova GmbH

```
|      |
|      |-Carpeta2
|            |
|            |-Carpeta3
|                  |
|                  |-Carpeta4
```

Y su archivo `.spp` está guardado en la `Carpeta3`, entonces las referencias a los archivos de la `Carpeta1` y `Carpeta2` serán así::

```
c:\Carpeta1\NombreDeArchivo.ext
c:\Carpeta1\Carpeta2\NombreDeArchivo.ext
```

Por su parte, las referencias a los archivos de la `Carpeta3` y `Carpeta4` serán así:

```
.\NombreDeArchivo.ext
.\Carpeta4\NombreDeArchivo.ext
```

Si desea que todos las rutas de archivo sean relativas, guarde los archivos `.spp` en el directorio raíz de su disco duro.

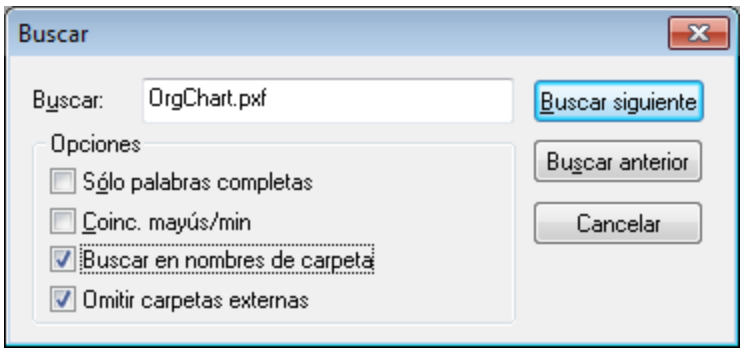
Arrastrar y colocar

En la ventana Proyecto puede arrastrar carpetas hasta otra carpeta o hasta otra posición de la misma carpeta. Los archivos se pueden arrastrar hasta otra carpeta, pero no se pueden cambiar de posición dentro de la misma carpeta (en las carpetas los archivos se organizan alfabéticamente). Además, puede arrastrar archivos y carpetas desde el explorador de Windows hasta la ventana Proyecto de XMLSpy.

Cómo realizar búsquedas en el proyecto

Puede buscar archivos y carpetas del proyecto por su nombre o sólo parte del nombre. Si se encuentra un archivo o una carpeta que coincida con la búsqueda, estos se resaltan uno por uno en la ventana Proyecto.

Para iniciar una búsqueda, en la ventana Proyecto seleccione la carpeta de proyecto en la que desea buscar y después seleccione el comando **Edición | Buscar** (**Ctrl+F**). Aparece el cuadro de diálogo Buscar (*imagen siguiente*). Escriba el texto que desea buscar y seleccione las opciones de búsqueda que correspondan (*véase más adelante*).



Este cuadro de diálogo ofrece estas opciones de búsqueda:

- Puede buscar *palabras completas* solamente. Esto limita los resultados porque el nombre de carpeta o de archivo debe incluir la palabra completa. En los nombres de archivo, la parte situada delante y detrás del punto (sin el punto) se considera una palabra.
- Puede buscar sólo los archivos o carpetas cuyo nombre coincida con el texto de búsqueda en el uso de *mayúsculas y minúsculas*.
- Puede buscar en los *nombres de carpeta* o sólo en los nombres de archivo.
- Puede incluir o excluir las *carpetas externas* 1319 en la búsqueda. Las carpetas externas son carpetas reales del sistema de archivos o de la red. Por el contrario, las carpetas de proyecto se crean dentro del proyecto pero no en el sistema.

Si la búsqueda encuentra resultados, se resalta el primer resultado en la ventana Proyecto. Para navegar por el resto de resultados de la búsqueda, haga clic en **Buscar siguiente** y **Buscar anterior**.

Actualizar los proyectos

Si se realizan cambios en una carpeta externa, el cambio no se refleja en la ventana Proyecto a no ser que se actualice el proyecto.

Recursos globales en el menú contextual

Al hacer clic con el botón derecho en una carpeta en la ventana Proyecto, el menú contextual que aparece incluye la opción **Agregar recurso global**, que sirve para añadir recursos globales 1042. Al hacer clic en este comando aparece el cuadro de diálogo "Agregar recurso global". Este diálogo enumera todos los recursos globales de tipo archivo o carpeta disponibles en el archivo XML de recursos globales activo. Seleccione un recurso global y este se añade a la carpeta de proyecto seleccionada.

Proyectos y proveedores de control de versiones

Si desea añadir un proyecto de XMLSpy a un repositorio de control de versiones, antes es necesario comprobar si la posición de los archivos del proyecto en la estructura del sistema de archivos permite añadir archivos desde un nivel inferior (entendiendo el directorio raíz como nivel superior de la estructura de directorios).

En otras palabras, el directorio donde está ubicado el **archivo de proyecto** básicamente representa el **directorio raíz** del proyecto en el repositorio de control de versiones. Los archivos que se añadan desde un nivel superior (al directorio raíz) se añaden al proyecto de XMLSpy, pero su posición en el repositorio puede ser diferente a la esperada, si es que se pueden colocar en esa posición.

Por ejemplo, en la estructura de carpetas que aparece más arriba, si un archivo de proyecto se guarda en la `Carpeta3` y se pone bajo control de versiones:

- Los archivos que se añadan en la `Carpeta1` no se pueden poner bajo control de versiones.
- Los archivos que se añadan a la `Carpeta2` se ponen bajo control de versiones en el directorio raíz del repositorio y no dentro de la carpeta de proyecto.
- Los archivos que se añadan a la `Carpeta3` y `Carpeta4` se añaden en la posición esperada y se ponen bajo control de versiones.

© 2016-2022 Altova GmbH

28.3.1 Proyecto nuevo



El comando **Proyecto nuevo** crea un proyecto **nuevo** en XMLSpy. Si al hacer clic en este comando está trabajando en otro proyecto, aparece un mensaje preguntando si desea cerrar todos los documentos que pertenecen al proyecto actual. El nombre del proyecto se le asigna al guardarlo en un archivo **.spp**.

28.3.2 Abrir proyecto



El comando **Abrir proyecto...** abre un proyecto de XMLSpy. Si al hacer clic en este comando está trabajando en otro proyecto, primero se cierra el proyecto actual.

28.3.3 Volver a cargar el proyecto



El comando **Volver a cargar el proyecto** vuelve a cargar el proyecto desde el disco. Si trabaja en un entorno de varios usuarios, a veces puede ser necesario volver a cargar el proyecto si otro usuario realizó cambios en el proyecto.

Nota importante: los archivos de proyecto (archivos `.spp`) son de hecho documentos XML que se pueden editar como cualquier otro archivo XML.

28.3.4 Cerrar el proyecto

El comando **Cerrar el proyecto** sirve para **cerrar** el proyecto activo. Si se realizaron cambios en el proyecto, la aplicación le pregunta si desea guardar el proyecto antes de cerrarlo. Si se realizaron cambios en el proyecto, esto se indica en la ventana Proyecto con un asterisco junto al nombre del proyecto.

28.3.5 Guardar el proyecto, Guardar el proyecto como



El comando **Guardar el proyecto** sirve para **guardar** el proyecto actual. También puede hacer clic en la ventana Proyecto y pulsar el icono .
El comando **Guardar el proyecto como** sirve para **guardar** el proyecto actual con un nombre nuevo.

28.3.6 Control de código fuente

Su aplicación de Altova es compatible con Microsoft SourceSafe y otros repositorios de archivos. El apartado Sistemas de control de código fuente compatibles[1100] incluye una lista con todos los sistemas compatibles con su aplicación de Altova. En este apartado describimos los comandos del submenú **Proyecto | Control de código fuente**, que sirven para trabajar con el sistema de control de código fuente desde su aplicación de Altova.

Resumen de la función de control de código fuente

Este es el mecanismo para colocar archivos de un proyecto bajo control de código fuente:

1. En XMLSpy se crea una carpeta de proyecto que contiene los archivos que se deben colocar bajo control de código fuente. Esta carpeta se corresponde normalmente con una carpeta local donde están guardados los archivos del proyecto. La ruta de acceso de la carpeta local se denomina ruta de acceso local.
2. En la base de datos del sistema de control de código fuente (también llamada repositorio) se crea una carpeta que contendrá los archivos que se deben poner bajo control de código fuente.
3. Los archivos del proyecto se añaden al control de código fuente con el comando **Proyecto | Control de código fuente | Agregar al control de código fuente**[1309].
4. Las acciones de control de código fuente (como proteger, desproteger y quitar archivos del control de código fuente) se pueden llevar a cabo con los comandos del submenú **Proyecto | Control de código fuente**[1302]. Los comandos de este submenú se describen en los siguientes subapartados.

Nota: si desea cambiar de proveedor de control de código fuente, tiene dos opciones: (i) cambiar las opciones de control de código fuente (**Herramientas | Opciones | Control de código fuente**[1625]) o (ii) cambiar de proveedor en el diálogo "Cambiar control de código fuente" (**Proyecto | Control de código fuente | Cambiar control de código fuente**[1316]).

Nota: un proyecto de control de código fuente no es lo mismo que un proyecto de la aplicación. Los proyectos de control de código fuente dependen de la estructura del directorio, mientras que los proyectos de XMLSpy son construcciones lógicas que no dependen directamente de la estructura del directorio.

Para más información, consulte la sección Control de código fuente[1097].

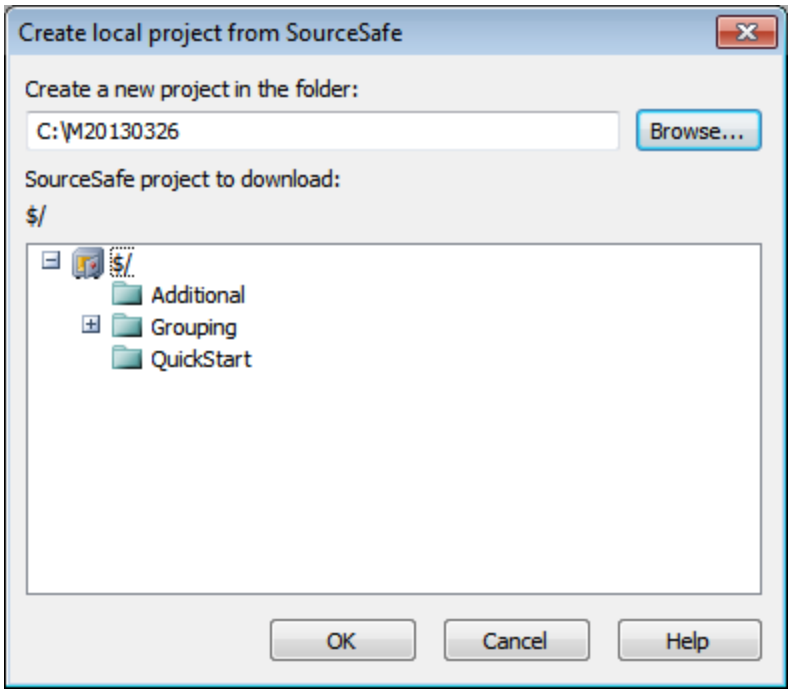
© 2016-2022 Altova GmbH

28.3.6.1 Abrir desde el control de código fuente

El comando **Abrir desde el control de código fuente** crea un proyecto de aplicación nuevo a partir de un proyecto bajo control de código fuente.

Estos son los pasos para crear el proyecto:

1. Dependiendo del sistema de control de código fuente que utilice, quizás sea necesario comprobar que ningún archivo del proyecto está desprotegido antes de crear un proyecto nuevo.
2. No es necesario tener abierto un proyecto en la aplicación.
3. Seleccione el comando **Proyecto | Control de código fuente | Abrir desde el control de código fuente**.
4. El sistema de control de código fuente que tiene instalado muestra sus cuadros de diálogo de verificación y conexión. Conéctese al repositorio pertinente (es decir, a la carpeta del repositorio que se corresponde con la carpeta local).
5. En el cuadro de diálogo que aparece (*imagen siguiente*) busque la carpeta local en la que desea copiar el contenido de la carpeta del repositorio (a la que se acaba de conectar). En la imagen siguiente, por ejemplo, la carpeta asociada del repositorio se llama `MiProyecto` y se representa con el signo `$`. La carpeta local es `C:\M20130326`.



6. Haga clic en **OK**. El contenido de la carpeta asociada (`MyProject`) se copiará a la carpeta local `C:\M20130326`. y aparece un cuadro de diálogo pidiendo que seleccione el archivo de proyecto (archivo `.spp`) que desea crear como proyecto nuevo.
7. Seleccione el archivo `.spp` que se copió a la carpeta local. En nuestro ejemplo, se trata del archivo `MiProyecto.spp` situado en la carpeta `C:\M20130326`. En la aplicación se crea un proyecto nuevo llamado `MiProyecto` y este proyecto aparece en la ventana Proyecto. Los archivos del proyecto están en la carpeta `C:\M20130326`.

Símbolos de control de código fuente

Las carpetas y los archivos aparecen con un icono diferente dependiendo de su estado:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

28.3.6.2 Habilitar control de código fuente

El comando **Habilitar control de código fuente** sirve para habilitar/deshabilitar el control de código fuente para un proyecto de la aplicación. Este comando habilita/deshabilita el control de código fuente para todo el proyecto, aunque se seleccione desde el menú contextual de un archivo o de una carpeta. Una vez habilitado el control de código fuente, el estado de los archivos se recupera y se muestra en la ventana proyecto.

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

28.3.6.3 Obtener la versión más reciente

El comando **Obtener la versión más reciente** (del menú **Proyecto | Control de código fuente**) recupera y coloca en el directorio de trabajo la versión más reciente de los archivos seleccionados del control de código fuente. Los archivos se recuperan como archivos de solo lectura y no están desprotegidos. Este comando es igual que el comando **Obtener**, la única diferencia es que no aparece el cuadro de diálogo "Obtener".

Si los archivos seleccionados están desprotegidos, la acción que se toma dependerá de lo que haga su sistema de control en situaciones así. Por lo general, el sistema de control de código fuente le preguntará si desea reemplazar el archivo desprotegido, combinarlo o dejarlo tal y como está.

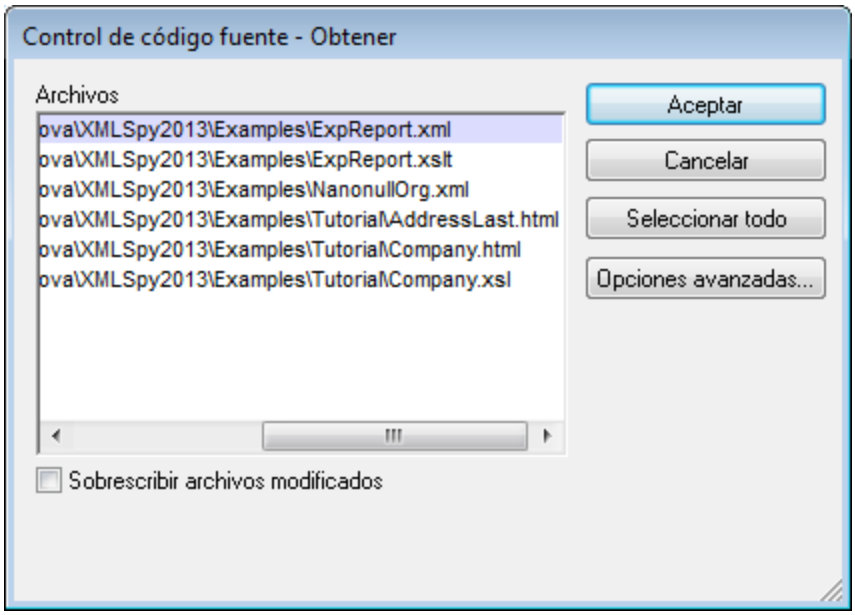
Nota: cuando seleccione este comando para una carpeta, el comando se aplica de forma recursiva. Esto significa que se aplica a todos los archivos situados dentro de la carpeta actual.

28.3.6.4 Obtener, Obtener carpetas

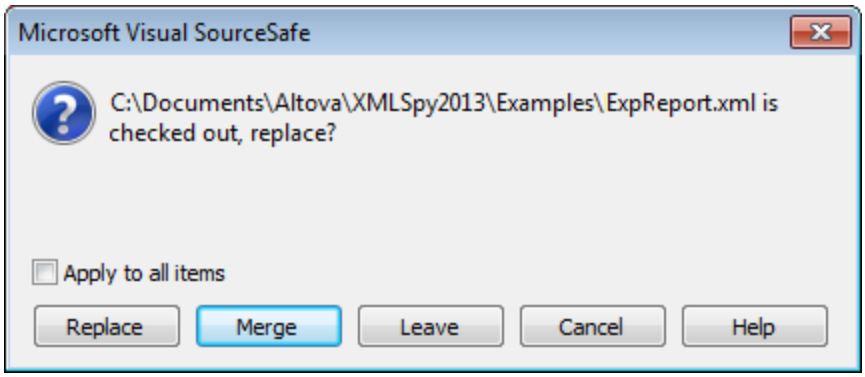
El comando **Obtener** (del menú **Proyecto | Control de código fuente**) recupera los archivos del repositorio como archivos de solo lectura. (Para poder editar un archivo, antes es necesario desprotegerlo.) El cuadro de diálogo "Obtener" enumera los archivos del objeto (proyecto o carpeta) sobre el que se ejecutó el comando **Obtener** (*imagen siguiente*). En el cuadro de diálogo puede marcar las casillas de los archivos que desea recuperar.

© 2016-2022 Altova GmbH

Nota: el comando **Obtener carpetas** sirve para seleccionar subcarpetas del repositorio (si su sistema de control de código fuente permite esta opción).

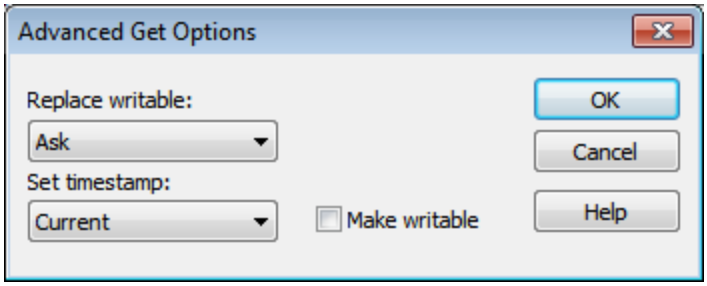


Si quiere, puede sobrescribir los archivos desprotegidos modificados marcando la opción *Sobrescribir archivos modificados* situada en la parte inferior del cuadro de diálogo. Al hacer clic en **Aceptar** los archivos se sobrescriben. Si alguno de los archivos sobrescritos está abierto, aparece un aviso preguntando si desea volver a cargar los archivos (botón **Volver a cargar**), cerrar los archivos (botón **Cerrar**) o conservar la vista actual del archivo (botón **Cancelar**).



Opciones avanzadas

El cuadro de diálogo "Opciones avanzadas" (*imagen siguiente*) se abre con el botón **Opciones avanzadas** del cuadro de diálogo "Obtener" (*primera imagen de este apartado*).



Aquí puede seleccionar (i) si reemplaza los archivos que se pueden escribir y que están desprotegidos, (ii) la marca de tiempo y (iii) si la propiedad de solo lectura del archivo recuperado se cambia para que el archivo se pueda escribir.

28.3.6.5 Desproteger, Proteger

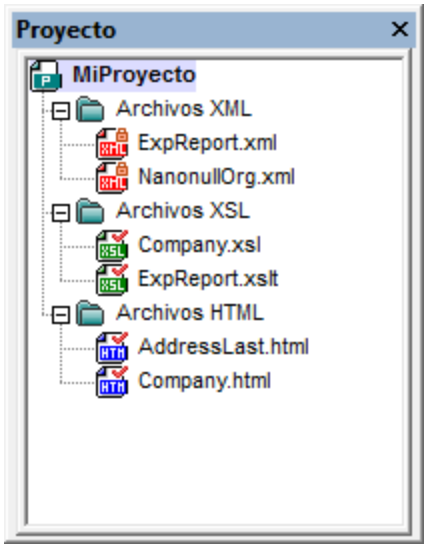
Tras poner un archivo de proyecto bajo control de código fuente, puede desprotegerlo o protegerlo. Para ello seleccione el archivo (en la ventana Proyecto) y seleccione el comando pertinente en el menú **Proyecto | Control de código fuente**, es decir **Desproteger** o **Proteger**.

Cuando se desprotege un archivo, se pone una copia del archivo del repositorio en la carpeta local. Los archivos desprotegidos se pueden editar. Si un archivo que está bajo control de código fuente no se desprotege, no se puede editar. Después de editar el archivo, los cambios se pueden guardar en el repositorio protegiendo el archivo. Incluso si no guarda el archivo, al protegerlo se guardan los cambios en el repositorio. El icono del archivo indica si está desprotegido o protegido.

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

Si selecciona el proyecto o una carpeta del proyecto, se seleccionan todos los archivos del objeto seleccionado. Para seleccionar varios objetos (archivos y carpetas) pulse la tecla **Ctrl** mientras hace clic en los objetos. La imagen siguiente muestra un proyecto que se desprotegió. Después se protegió el archivo `ExpReport.xml`.

© 2016-2022 Altova GmbH



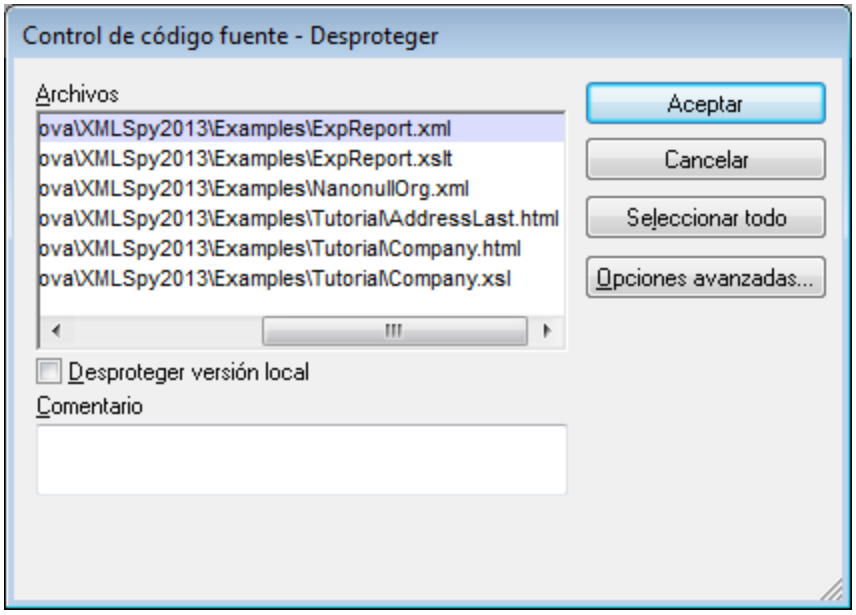
Guardar cambios y rechazar cambios

Recuerde que al proteger un archivo puede elegir si se protegen los cambios o no. Es decir, puede guardar los cambios en el repositorio y seguir con el archivo desprotegido. Esto es muy útil si desea guardar cambios cada cierto tiempo y después seguir editando el archivo desprotegido.

Si desprotegió un archivo y realizó cambios pero quiere rechazarlos por cualquier motivo, puede volver a la versión del documento guardada en el repositorio con solo hacer clic en el comando **Proyecto | Control de código fuente | Deshacer desprotección**.

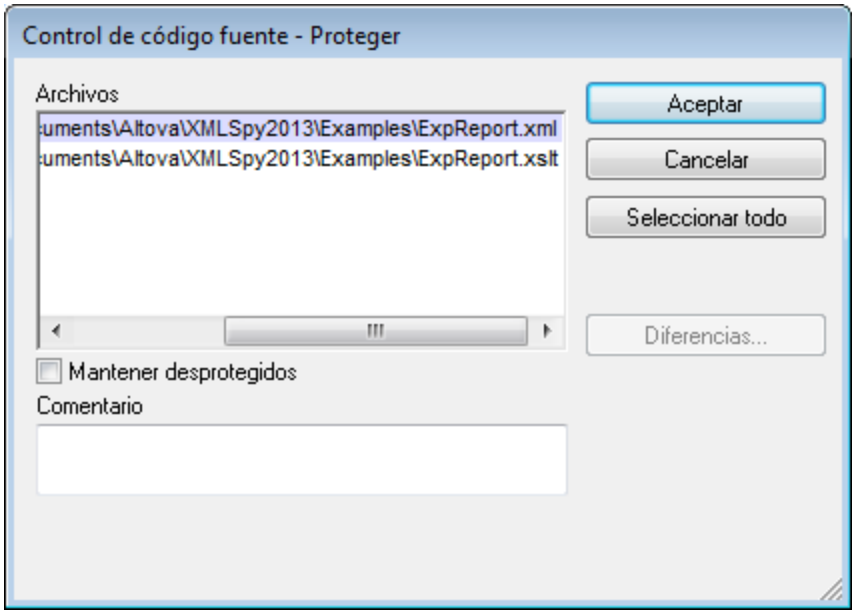
Desproteger

El cuadro de diálogo "Desproteger" (*imagen siguiente*) sirve para: (i) seleccionar los archivos que desea desproteger y (ii) seleccionar si se desprotege la versión del repositorio o la versión local.



Proteger

El cuadro de diálogo "Proteger" (*imagen siguiente*) sirve para: (i) seleccionar los archivos que desea proteger y (ii) mantener el archivo desprotegido.



Nota: en estos dos cuadros de diálogo aparecen varios archivos si el objeto seleccionado (el proyecto o las carpetas de proyecto) contienen varios archivos.

© 2016-2022 Altova GmbH

28.3.6.6 Anular desprotección

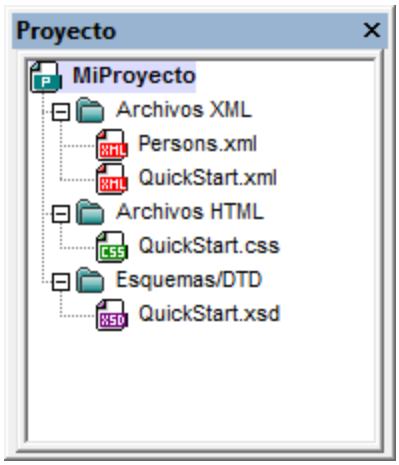
Si desprotegió un archivo y realizó cambios pero ahora prefiere rechazar esos cambios por cualquier motivo, puede volver a la versión del documento guardada en el repositorio. Para ello seleccione el comando **Proyecto | Control de código fuente | Anular desprotección**.

Las carpetas y los archivos aparecen con un icono diferente dependiendo de su estado:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

28.3.6.7 Agregar al control de código fuente

Después de añadir un proyecto al control de código fuente, puede añadir archivos o grupos de archivos al sistema de control. Seleccione el archivo en la ventana Proyecto y haga clic en el comando **Proyecto | Control de código fuente | Agregar al control de código fuente**. Para seleccionar varios archivos mantenga pulsada la tecla **Ctrl** mientras hace clic en los archivos que desea añadir. Si ejecuta el comando en una carpeta de proyecto (de color verde), entonces se agregan al control de código fuente todos los archivos y subcarpetas de la carpeta.



Cuando se agregan archivos al control de código fuente, la jerarquía de la carpeta local se copia al repositorio (no la jerarquía de la carpeta de proyecto). Es decir, si un archivo está en una subcarpeta de la carpeta local, entonces se crean en el repositorio la carpeta primaria y las demás carpetas antecesoras del archivo.

Cuando se añade el primer archivo de un proyecto al control de código fuente, se crean los enlaces correspondientes en el repositorio y se añade automáticamente el archivo de proyecto (archivo `.spp`). Para más información consulte el apartado Agregar al control de código fuente 1105.

Símbolos de control de código fuente

Las carpetas y los archivos aparecen con un icono diferente dependiendo de su estado:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

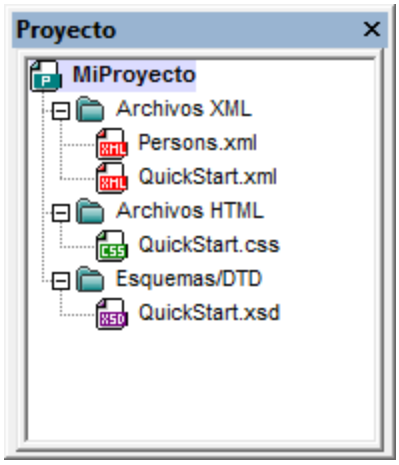
28.3.6.8 Quitar del control de código fuente

Para quitar un archivo del control de código fuente, selecciónelo y haga clic en el comando **Proyecto | Control de código fuente | Quitar del control de código fuente**. También puede quitar: (i) archivos de una carpeta de proyecto ejecutando el comando sobre la carpeta, (ii) varios archivos, si mantiene pulsada la tecla **Ctrl** mientras los selecciona y (iii) todo el proyecto, ejecutando el comando sobre el proyecto.

28.3.6.9 Compartir desde el control de código fuente

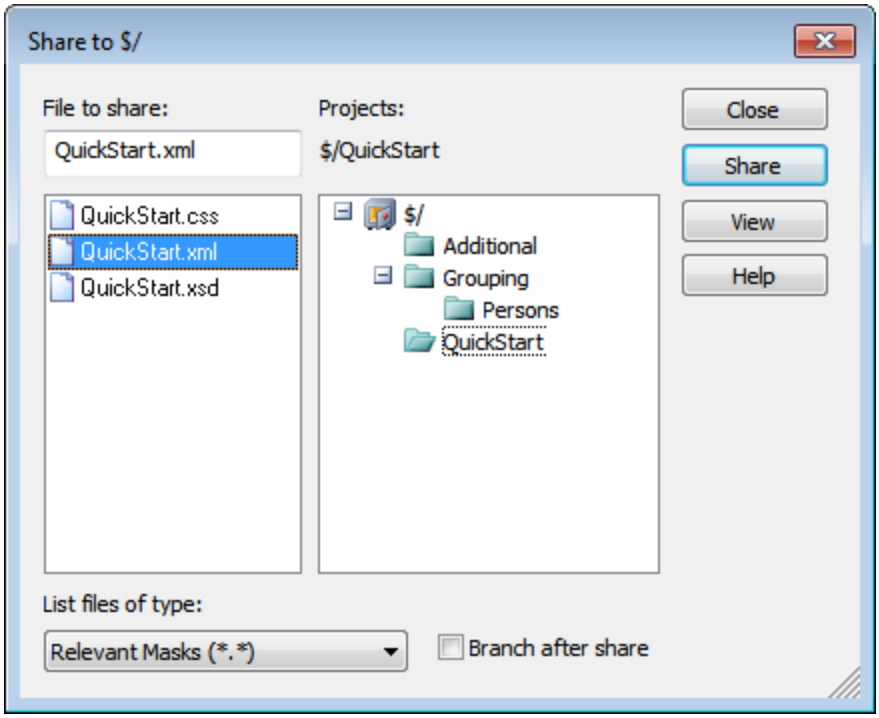
El comando **Compartir desde el control de código fuente** está disponible si su sistema de control de código fuente permite compartir contenido. Por ejemplo, puede compartir un archivo para que esté disponible en varias ubicaciones locales. Si se realizan cambios en uno de estos archivos locales, los cambios se reflejan en todas las versiones "compartidas" del archivo.

En la ventana Proyecto seleccione el proyecto (*imagen siguiente*) y después haga clic en **Compartir desde el control de código fuente**.



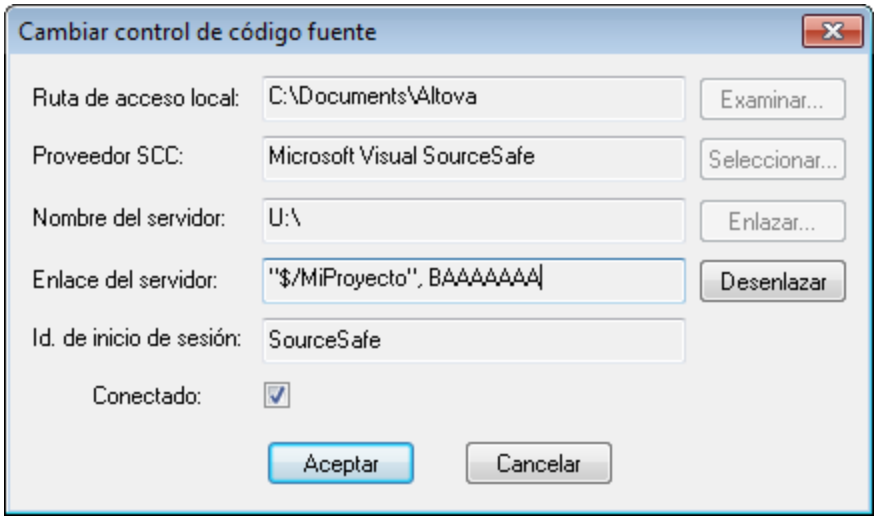
Aparece el cuadro de diálogo "Share To [carpeta]" (*imagen siguiente*).

© 2016-2022 Altova GmbH



Ahora puede seleccionar los archivos que desea compartir. Primero seleccione en el panel derecho la carpeta donde están los archivos. Los archivos de la carpeta elegida aparecen en el panel izquierdo. Seleccione el archivo que desea compartir (para seleccionar varios pulse la tecla **Ctrl** mientras hace clic en los archivos). Los archivos seleccionados aparecen en el cuadro situado en la parte superior izquierda del diálogo. Haga clic en **Share** (*compartir*) y después en **Close** (cerrar) para copiar los archivos seleccionados a la carpeta local compartida.

El nombre de la carpeta compartida aparece en el nombre del cuadro de diálogo. En la imagen anterior, por ejemplo, se trata de la carpeta local (porque el signo `$` representa la carpeta del repositorio a la que está enlazada la carpeta local). En el cuadro de diálogo "Cambiar control de código fuente" puede ver y configurar la carpeta compartida (*imagen siguiente*) cambiando la ruta de acceso local y el enlace de servidor.



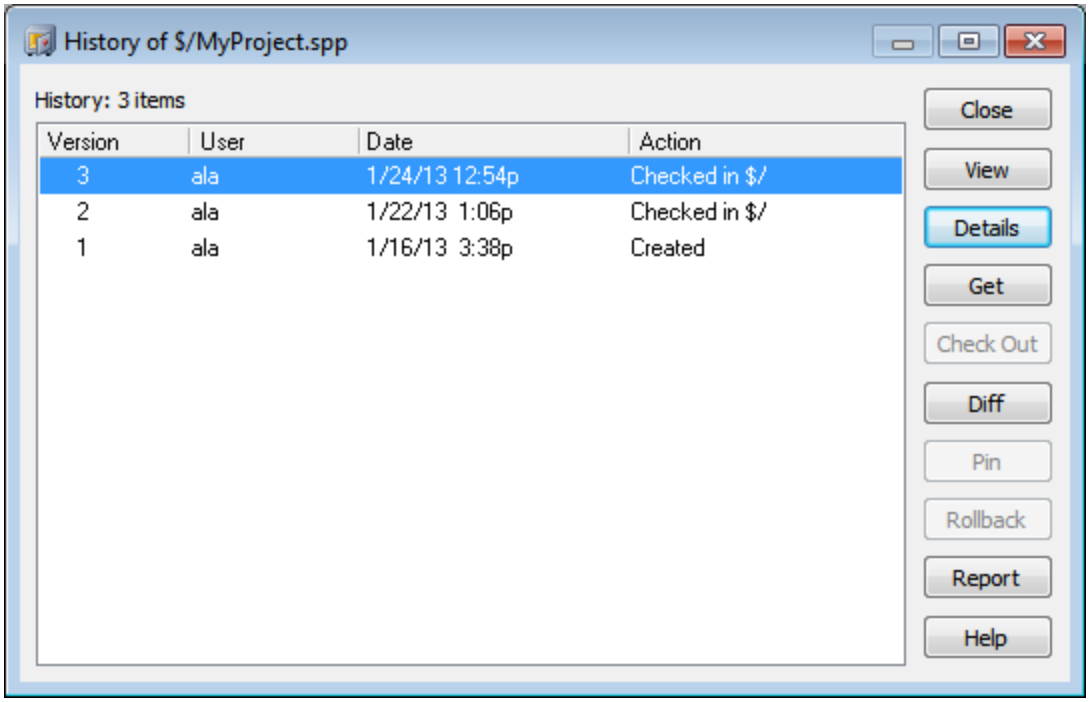
Para más información consulte la documentación de su sistema de control de código fuente.

28.3.6.10 Mostrar historial

El comando **Mostrar historial** activa la función del mismo nombre del sistema de control de código fuente activo. Muestra el historia del archivo seleccionado en la ventana Proyecto. Seleccione el título del proyecto para ver el historial del archivo de proyecto (archivo `.spp`). Puede ver información sobre las versiones previas de un archivo y las diferencias con la versión actual y recuperar versiones previas del archivo.

La siguiente imagen muestra el diálogo "Historial" del sistema de control de versiones Visual SourceSafe y enumera todas las versiones del archivo `MyProject.spp`.

© 2016-2022 Altova GmbH



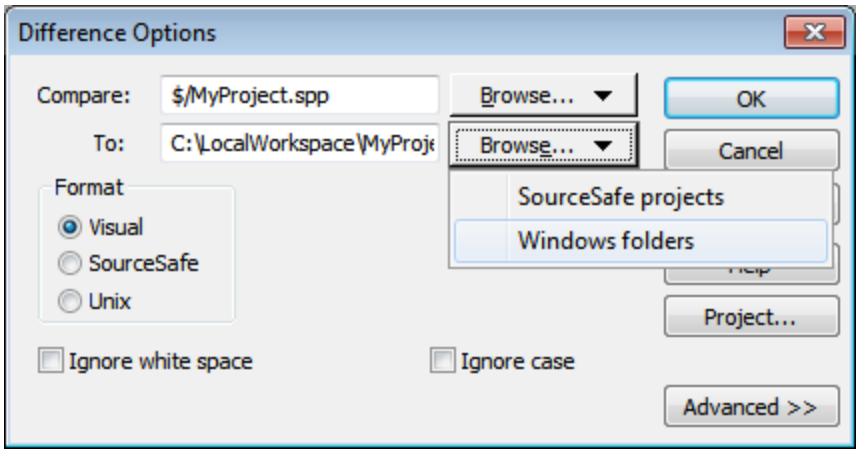
Este cuadro de diálogo permite compartir y obtener versiones determinadas del archivo en cuestión. Al hacer doble clic en un archivo de la lista se abre el cuadro de diálogo "History details" (Historial detallado). Estos son los botones de este cuadro de diálogo:

- ***Close:*** cierra el cuadro de diálogo.
- ***View:*** abre un cuadro de diálogo donde puede seleccionar el tipo de visor de archivos.
- ***Details:*** abre un cuadro de diálogo donde puede ver las propiedades[1315] del archivo activo actual.
- ***Get:*** recupera una de las versiones previas del archivo y la coloca en el directorio de trabajo.
- ***Check Out:*** desprotege una versión previa del archivo.
- ***Diff:*** abre el cuadro de diálogo Difference options[1313], donde puede definir las opciones de comparación de las dos versiones del archivo. Marque dos versiones del archivo en esta ventana y después haga clic en **Diff** para ver las diferencias.
- ***Pin:*** ancla o desancla una versión del archivo, para que pueda definir la versión de archivo que desea usar al comparar dos archivos.
- ***Rollback:*** revierte el archivo a la versión seleccionada.
- ***Report:*** genera un informe de historial que se puede enviar a una impresora, a un archivo o al portapapeles.
- ***Help:*** abre la ayuda en línea del complemento de control de código fuente.

28.3.6.11 Mostrar diferencias

El comando **Mostrar diferencias** está disponible si selecciona un archivo de la ventana Proyecto. Para seleccionar el archivo de proyecto (`.spp`) haga clic en el título del proyecto en la ventana Proyecto. El comando **Mostrar diferencias** inicia la herramienta de comparación del sistema de control de código fuente. Así puede ver las diferencias directamente en la aplicación de Altova.

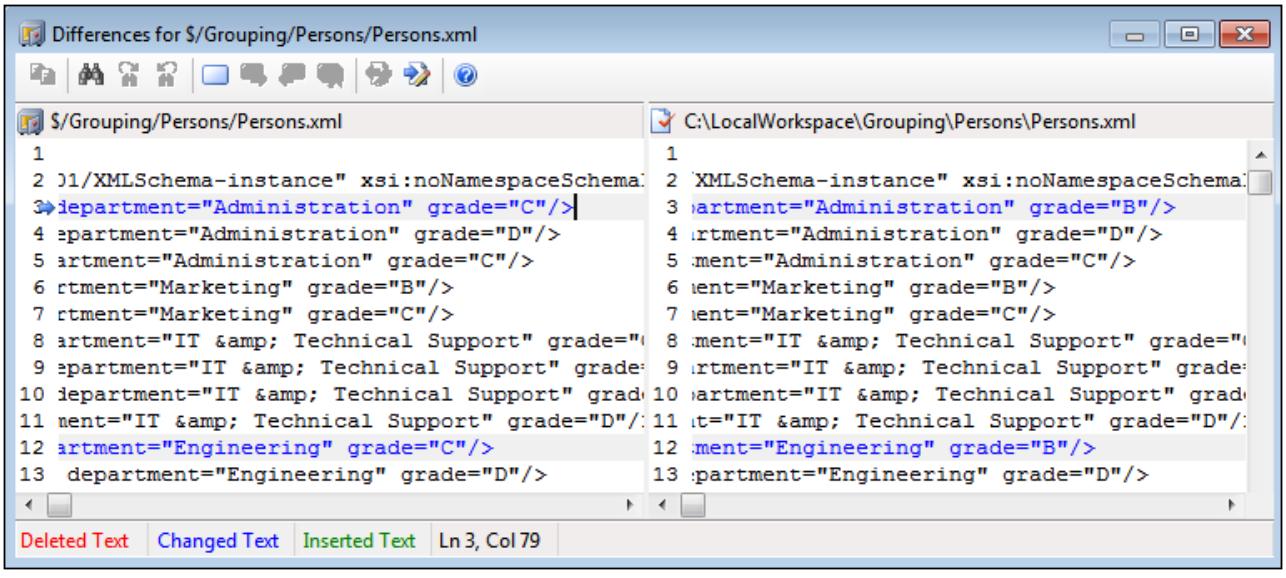
La imagen siguiente muestra la herramienta de comparación del sistema de control Visual SourceSafe.



En los campos de texto *Compare* y *To* aparecen por defecto la versión del repositorio y la versión local respectivamente. Para buscar otros archivos:

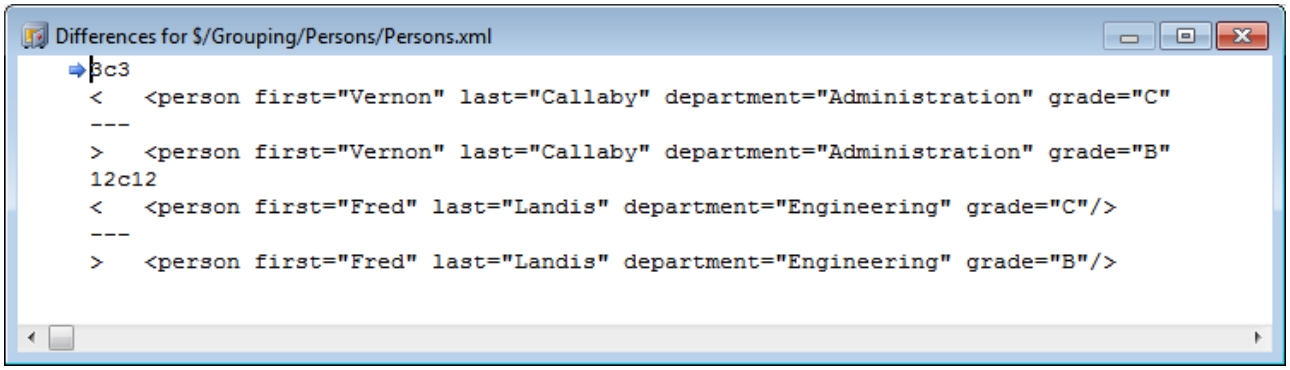
1. Seleccione la opción *SourceSafe projects* en la lista desplegable del cuadro combinado **Examinar** (para buscar archivos del repositorio) o la opción *Windows folders* (para buscar carpetas locales).
2. Busque los archivos y selecciónelos.

Seleccione las opciones que desea usar y haga clic en **OK** para iniciar la comparación. Los resultados aparecen en otra ventana. Las imágenes que aparecen a continuación muestran los resultados de una misma comparación en dos formatos diferentes.



La primera imagen muestra los resultados de la comparación de Visual SourceSafe en formato visual (ver la opción *Format* > *Visual* del cuadro de diálogo de la primera imagen). La imagen que aparece a continuación muestra el resultado en formato Unix. En ambos casos, se encontraron dos diferencias.

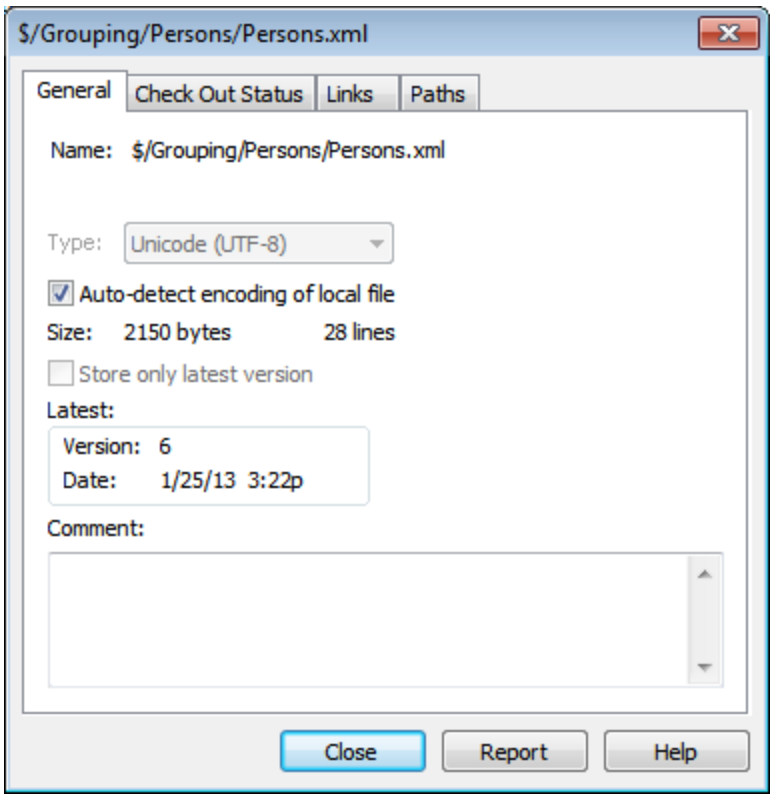
© 2016-2022 Altova GmbH



Para más información consulte la documentación de su sistema de control de código fuente.

28.3.6.12 Mostrar propiedades

El comando **Mostrar propiedades** muestra las propiedades del archivo que está seleccionado (*imagen siguiente*). Las propiedades que aparecen dependen del sistema de control de código fuente utilizado. La imagen siguiente corresponde al sistema Visual SourceSafe.



Recuerde que este comando se habilita solamente si selecciona un solo archivo.

Para más información consulte la documentación de su sistema de control de código fuente.

28.3.6.13 Actualizar estado

El comando **Actualizar estado** actualiza el estado de todos los archivos del proyecto independientemente de cuál sea su estado actual.

28.3.6.14 Administrador del control de código fuente

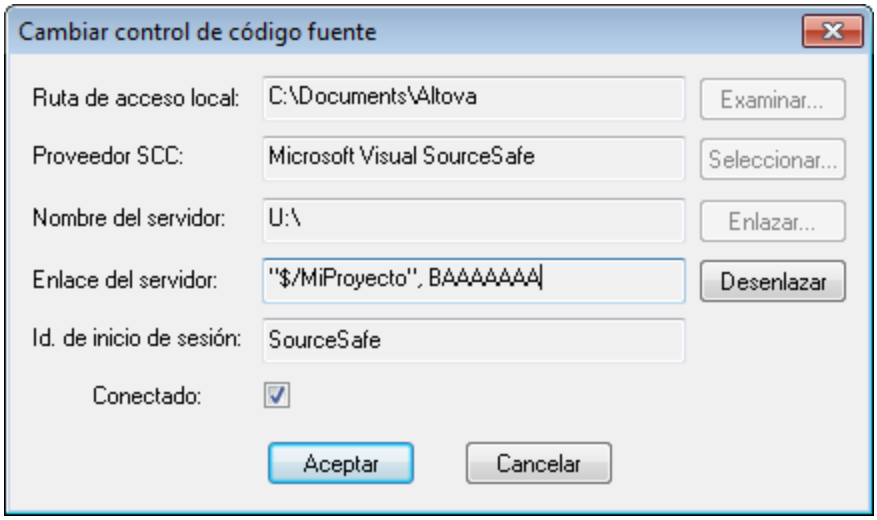
El comando **Administrador del control de código fuente** inicia el software de control de código fuente en su interfaz nativa.

28.3.6.15 Cambiar control de código fuente

El enlace actual es lo que usa el proyecto de aplicación activo para conectarse a la BD del control de código fuente. Por tanto, en enlace actual debe ser correcto. Esto significa que el archivo de proyecto de la aplicación (archivo `.spp`) debe estar en la carpeta local y la carpeta enlazada del repositorio debe ser la base de datos donde se almacenan los archivos de este proyecto. Por lo general, la carpeta enlazada y sus subcarpetas se corresponden con la carpeta de la copia de trabajo local y sus subcarpetas.

En el cuadro de diálogo "Cambiar control de código fuente" (*imagen siguiente*) puede cambiar el sistema de control (*Proveedor SCC*), la carpeta local (*Ruta de acceso local*) y el enlace al repositorio (*Nombre del servidor* y *Enlace del servidor*).

Las opciones solamente se pueden cambiar después de desenlazar el enlace actual. Para ello, pulse el botón **Desenlazar**. Ahora puede editar las demás opciones.



1. Con el botón **Examinar** puede buscar la carpeta local y con el botón Seleccionar puede seleccionar uno de los sistemas de control de código fuente instalados.

© 2016-2022 Altova GmbH

2. Después puede enlazar la carpeta local a una BD del repositorio. Para ello pulse el botón **Enlazar**. Aparece el cuadro de diálogo de conexión de sus sistema de control.
3. Si escribió un *Id. de inicio de sesión*, este identificador se pasa al sistema de control. De lo contrario, deberá escribir sus datos de inicio de sesión en el diálogo de conexión.
4. Seleccione la BD del repositorio a la que desea enlazar esta carpeta local.
5. Cuando termine haga clic en **Aceptar**.

28.3.7 Agregar archivos al proyecto



El comando **Proyecto** | **Agregar archivos al proyecto** agrega archivos al proyecto actual. Use este comando para agregar archivos a cualquier carpeta del proyecto. Puede seleccionar un solo archivo o un grupo de archivos en el cuadro de diálogo Abrir. Si agrega archivos al proyecto (en lugar de agregarlos a una carpeta del proyecto), estos se distribuyen en las carpetas del proyecto dependiendo de las extensiones de tipo de archivo definidas en el cuadro de diálogo Propiedades del proyecto[1328].

28.3.8 Agregar recurso global al proyecto

El comando **Proyecto** | **Agregar recurso global al proyecto** abre el cuadro de diálogo "Agregar recurso global", donde puede seleccionar un recurso global de tipo archivo o carpeta y agregarlo al proyecto. Si selecciona un recurso global de tipo archivo, el archivo se añade a la carpeta correspondiente según las extensiones de tipo de archivo definidas en el cuadro de diálogo Propiedades del proyecto[1328]. Si selecciona un recurso global de tipo carpeta, la carpeta se abre en un cuadro de diálogo "Abrir" y debe seleccionar un archivo. El archivo seleccionado en este cuadro de diálogo se añade a la carpeta de proyecto correspondiente según las extensiones de tipo de archivo definidas en el cuadro de diálogo Propiedades del proyecto[1328]. Para más información, consulte la sección del *Manual del usuario* dedicada a los recursos globales.

28.3.9 Agregar URL al proyecto



El comando **Proyecto** | **Agregar URL al proyecto** agrega una URL al proyecto actual. Las direcciones URL de un proyecto incluyen su objeto de destino en el proyecto. Si ejecuta una operación por lotes en una URL o en una carpeta que contiene una URL, XMLSpy recupera el documento de la URL y ejecuta en él la operación solicitada.

28.3.10 Agregar archivo activo al proyecto



El comando **Proyecto** | **Agregar archivo activo al proyecto** añade el archivo activo al proyecto actual. Si acaba de abrir un archivo desde el disco duro o desde una URL, puede agregar el archivo al proyecto con este comando.

28.3.11 Agregar archivo activo y relacionados al proyecto



El comando **Proyecto** | **Agregar archivo activo y relacionados al proyecto** añade al proyecto el archivo XML que está activo y todos los archivos relacionados con él. Si está trabajando con un documento XML que está basado en una DTD o en un esquema, este comando no sólo añade al proyecto el documento XML activo, sino también todos los archivos relacionados con él (por ejemplo, la DTD y todas las entidades externas analizadas a las que hace referencia la DTD).

Nota importante: no se consideran archivos relacionados los archivos a los que se hace referencia mediante instrucciones de procesamiento (por ejemplo, archivos XSLT).

28.3.12 Agregar carpeta de proyecto al proyecto



El comando **Proyecto** | **Agregar carpeta de proyecto al proyecto** añade una carpeta nueva al proyecto actual. Use este comando para agregar una carpeta nueva al proyecto actual o una subcarpeta a una carpeta de proyecto. Este comando también está disponible en el menú contextual que aparece al hacer clic con el botón derecho en una carpeta en la ventana Proyecto.

Nota: las carpetas de proyecto se pueden arrastrar y colocar en otra carpeta de proyecto o en otra posición del proyecto. Además, puede arrastrar carpetas desde el explorador de Windows hasta cualquier carpeta de la ventana Proyecto.

Nota: las carpetas de proyecto son verdes, mientras que las carpetas externas[1319] son amarillas.

© 2016-2022 Altova GmbH

28.3.13 Agregar carpeta externa al proyecto

El comando **Proyecto | Agregar carpeta externa al proyecto** añade una carpeta externa nueva al proyecto actual. Use este comando para agregar una carpeta de red o local al proyecto actual. Este comando también está disponible en el menú contextual que aparece al hacer clic con el botón derecho en una carpeta en la ventana Proyecto.

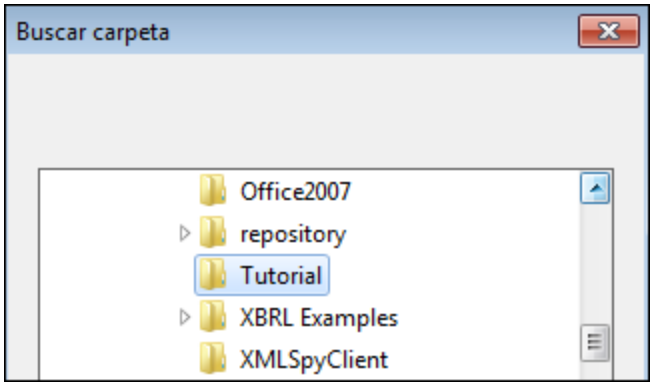
Nota: las carpetas externas son amarillas, mientras que las carpetas de proyecto[1318] son verdes.

Nota: los archivos de las carpetas externas no se pueden poner bajo control de código fuente.

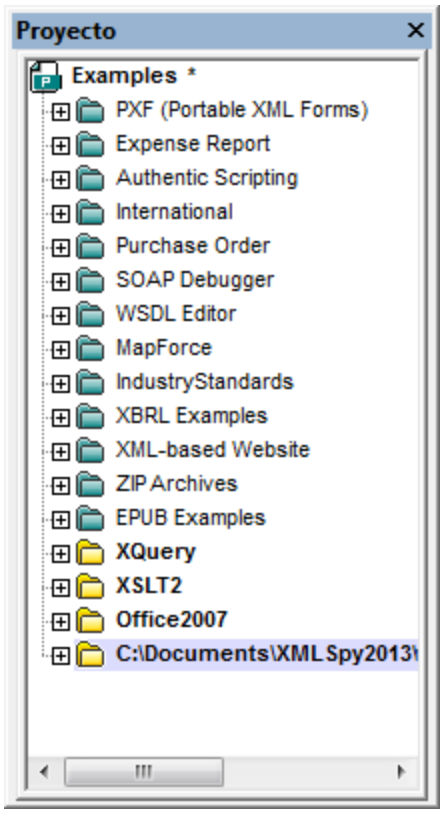
Agregar carpetas externas al proyecto

Siga estos pasos para agregar una carpeta externa al proyecto:

1. Seleccione la opción de menú **Proyecto | Agregar carpeta externa al proyecto**
2. En el cuadro de diálogo "Buscar carpeta" seleccione la carpeta que desea agregar y haga clic en **Aceptar**. La carpeta seleccionada aparece en la ventana Proyecto.



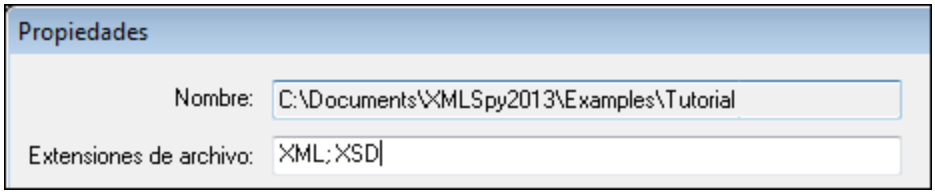
3. Haga clic en el icono + para ver el contenido de la carpeta.



Filtrar el contenido de las carpetas

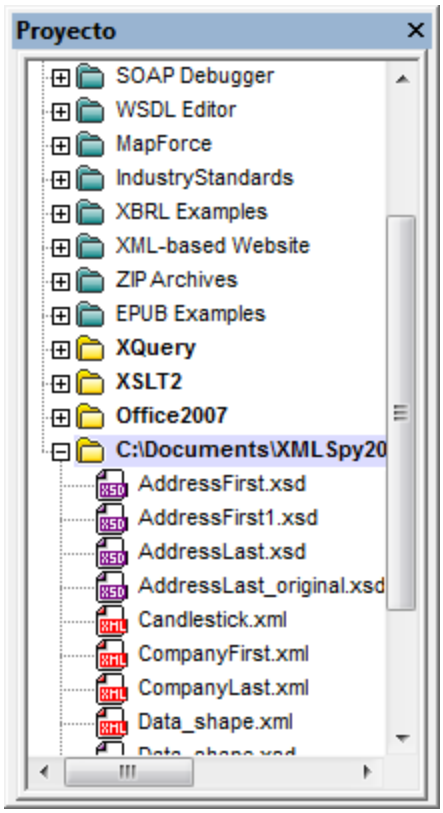
Siga estos pasos para filtrar el contenido de la carpeta:

1. Haga clic con el botón derecho en la carpeta local y seleccione la opción de menú **Propiedades**. Aparece el cuadro de diálogo Propiedades.



2. En el campo *Extensiones de archivo:* escriba las extensiones de archivo de los tipos de archivo que desea ver. Para definir varios tipos de archivo, separe las extensiones con un **punto y coma** (*XML y XSD en el ejemplo de la imagen anterior*).
3. Para confirmar haga clic en **Aceptar**.

© 2016-2022 Altova GmbH



Ahora la ventana Proyecto muestra solamente los archivos XML y XSD de la carpeta externa seleccionada.

Validar carpetas externas

Siga estos pasos para validar y comprobar si los documentos de sus carpetas externas tienen un formato XML correcto:

1. Seleccione qué tipo de archivos desea ver y validar en la carpeta externa (siguiendo los pasos descritos más arriba).
2. Seleccione la carpeta y haga clic en el comando de menú **XML | Comprobar formato XML** o **Validar el documento XML** (**F7** y **F8** respectivamente).
3. Corrija el error y ejecute de nuevo el proceso de validación para asegurarse de que se solucionó el problema.

Actualizar las carpetas de proyecto externas

Es posible que añada o elimine archivos en la carpeta local o de red. Cuando lo haga, será necesario actualizar la vista de la carpeta externa. Para ello, haga clic con el botón derecho en la carpeta externa y seleccione la opción **Actualizar** en el menú contextual.

Eliminar carpetas externas y sus archivos

Para eliminar la carpeta externa de la ventana Proyecto, selecciónela y pulse la tecla **Suprimir**. También puede hacer clic con el botón derecho en la carpeta externa y seleccionar el comando **Eliminar**. Recuerde que la carpeta sólo se elimina en la ventana Proyecto, pero no se elimina ni en el disco duro ni en la red.

Para eliminar un archivo de una carpeta externa es necesario eliminar el archivo en el disco duro o en la red. Para que el cambio se refleje en el proyecto es necesario actualizar el contenido de la carpeta (haga clic en la carpeta con el botón derecho y seleccione **Actualizar**).

Nota: las carpetas externas se pueden arrastrar y colocar en otra carpeta de proyecto o en otra posición del proyecto. Además, puede arrastrar carpetas externas desde el explorador de Windows hasta cualquier carpeta de la ventana Proyecto.

28.3.14 Agregar carpeta web externa al proyecto

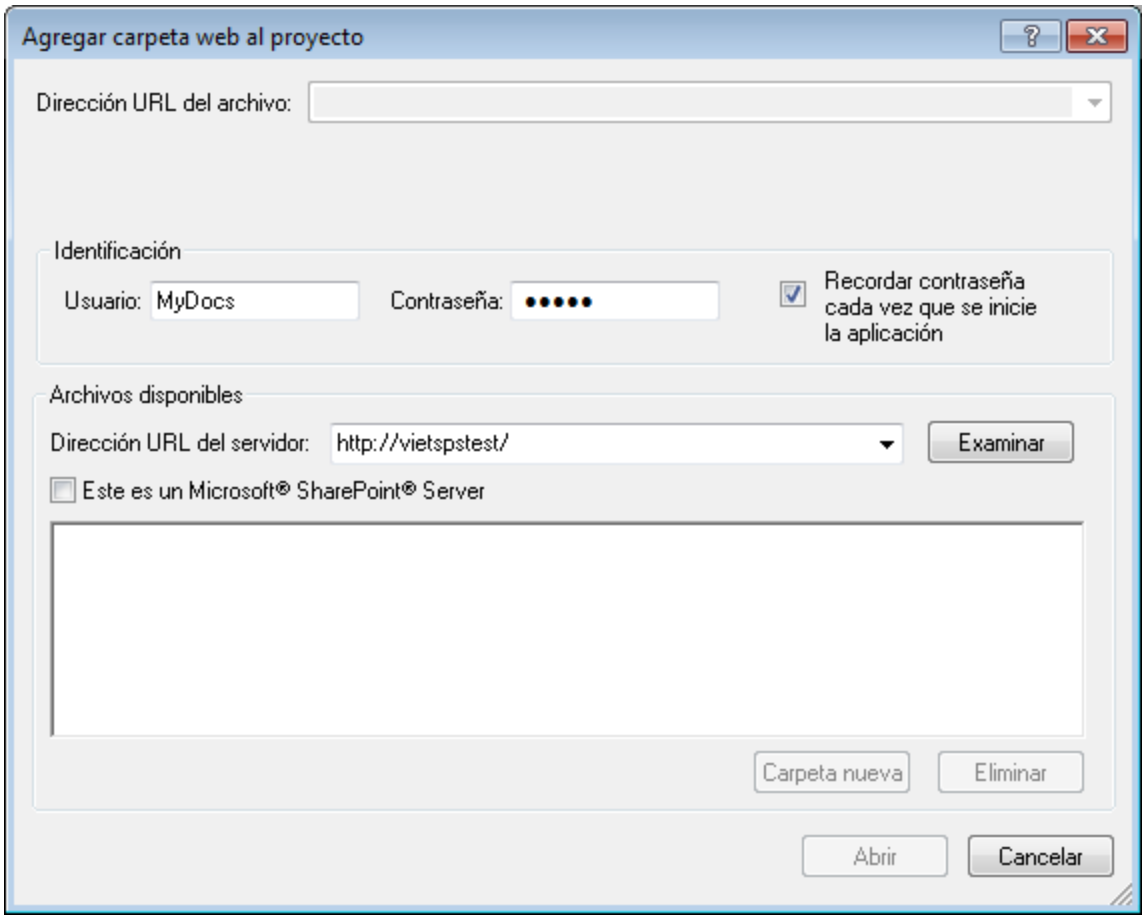
Este comando añade una carpeta web externa al proyecto actual. Este comando también está disponible en el menú contextual que aparece al hacer clic con el botón derecho en una carpeta de la ventana Proyecto. Recuerde que los archivos de carpetas externas no se pueden poner bajo control de código fuente.

Agregar carpetas web externas al proyecto

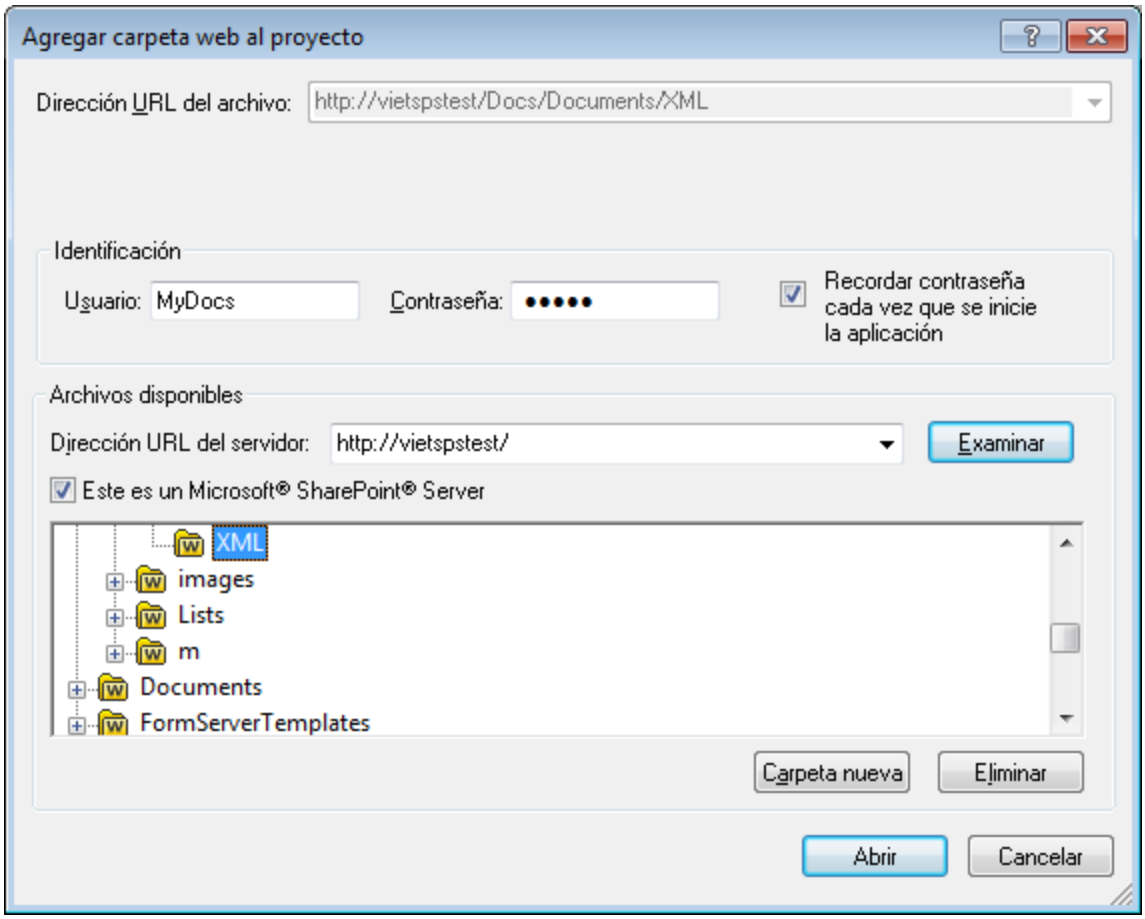
Siga estos pasos para agregar una carpeta web externa al proyecto:

1. Seleccione la opción de menú **Proyecto | Agregar carpeta web externa al proyecto**. Aparece el cuadro de diálogo "Agregar carpeta web al proyecto" (*imagen siguiente*).

© 2016-2022 Altova GmbH

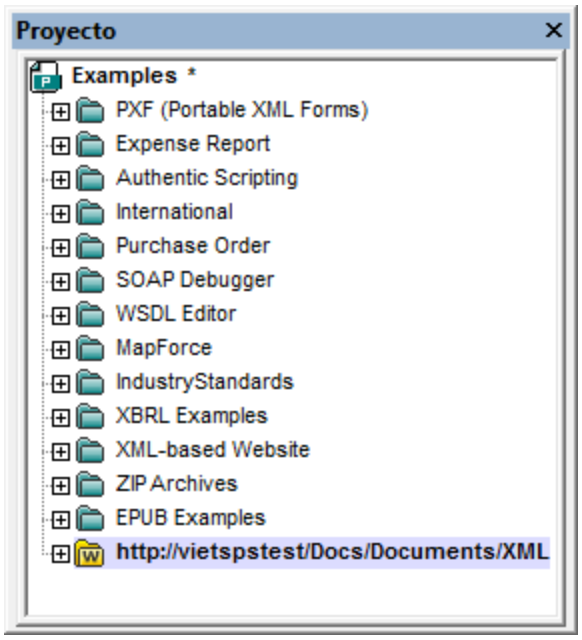


2. Haga clic en el campo *Dirección URL del servidor* y escriba la URL del servidor. Si se trata de un servidor Microsoft® SharePoint® Server, marque la casilla. Consulte el subapartado titulado ***Carpetas de los servidores Microsoft® SharePoint® Server*** de esta página para más información sobre cómo trabajar con archivos de este tipo de servidor.
3. Si el servidor está protegido con contraseña, escriba su Id. de usuario y contraseña en los campos correspondientes.
4. Haga clic en **Examinar** para conectarse al servidor y ver las carpetas que están disponibles.

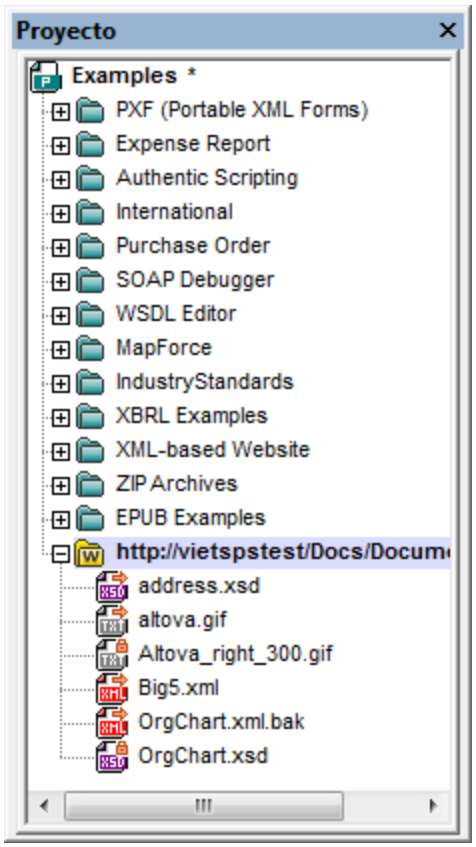


5. Haga clic en la carpeta que desea agregar al proyecto. El botón **Abrir** no se habilita hasta que no seleccione una carpeta. La URL de la carpeta aparece ahora en el campo *Dirección URL del archivo*.
6. Haga clic en **Abrir** para agregar la carpeta al proyecto.

© 2016-2022 Altova GmbH



7. Haga clic en el icono + para ver el contenido de la carpeta.



Filtrar el contenido de las carpetas

Para filtrar el contenido de una carpeta, haga clic con el botón derecho en la carpeta y seleccione **Propiedades** en el menú contextual. Aparece el cuadro de diálogo Propiedades. Haga clic en el campo *Extensiones de archivo* y escriba los tipos de archivo que desea ver en la carpeta (por ejemplo, archivos XML y XSD). Separe cada tipo de archivo con un punto y coma (por ejemplo, `xml; xsd; sps`). A partir de ahora la ventana Proyecto muestra solamente los tipos de archivo que tienen las extensiones indicadas.

Validar carpetas web externas

Para validar los archivos de una carpeta y comprobar si tienen un formato XML correcto, seleccione la carpeta y después haga clic en el icono **Comprobar formato XML**  o **Validar el documento XML**  (**F7** y **F8** respectivamente). La operación se ejecuta en todos los archivos que están visibles en la carpeta. Si un archivo no es válido o tiene un formato XML incorrecto, entonces el archivo se abre en la ventana principal para que pueda editarlo. Corrija el error y reinicie el proceso para volver a revisar el resto de la carpeta. Recuerde que también puede seleccionar varios archivos de la carpeta si mantiene pulsada la tecla **Ctrl** mientras hace clic en los archivos uno por uno. En este caso, al pulsar **F7** o **F8**, se revisan únicamente los archivos seleccionados.

Actualizar el contenido de las carpetas de proyecto

Es posible que se añadan o eliminen archivos en la carpeta web. Cuando esto ocurra, será necesario actualizar la vista de la carpeta externa. Para ello, haga clic con el botón derecho en la carpeta externa y seleccione la opción **Actualizar** en el menú contextual.

Eliminar carpetas y archivos

Puesto que lo que se añadió al proyecto es la carpeta web, solamente se puede eliminar del proyecto la carpeta web y no los archivos que están dentro de la carpeta. Para eliminar una carpeta web del proyecto, tiene dos opciones: (i) puede hacer clic con el botón derecho en la carpeta y seleccionar **Eliminar** o (ii) puede seleccionar la carpeta y pulsar la tecla **Suprimir**. Recuerde que la carpeta se elimina solamente en la ventana Proyecto, no en el servidor web.

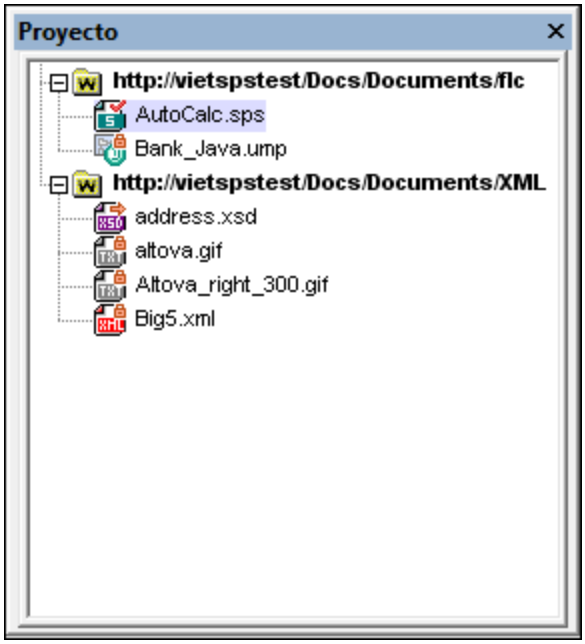
Nota: si en la ventana Proyecto hace clic con el botón derecho en un archivo de la carpeta web y pulsa la tecla **Suprimir**, el archivo no desaparece de la ventana Proyecto. Para que desaparezca es necesario eliminar el archivo en el servidor y después actualizar el contenido de la carpeta externa.

Carpetas de los servidores Microsoft® SharePoint® Server

Cuando añada al proyecto una carpeta de un servidor Microsoft® SharePoint® Server, los archivos de la carpeta se pueden proteger y desproteger con los comandos del menú contextual de la lista de archivos de la ventana Proyecto (*imagen siguiente*). Para usar estos comandos haga clic con el botón derecho en el archivo con el que desea trabajar y seleccione el comando pertinente (**Desproteger**, **Proteger** o **Deshacer desprotección**).

El Id. de usuario y la contraseña se pueden guardar en las propiedades de las carpetas del proyecto 1328 para que no sea necesario pasar por el proceso de verificación cada vez que acceda al servidor.

© 2016-2022 Altova GmbH



En la ventana Proyecto (*imagen anterior*), los iconos de archivo tienen símbolos que indican el estado de los archivos en el servidor:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

Recuerde que:

- Después de desproteger un archivo, puede editarlo en su aplicación de Altova y guardarlo con el comando **Archivo | Guardar (Ctrl+S)**.
- Puede proteger el archivo editado con el menú contextual de la ventana Proyecto (*imagen anterior*) o con el menú contextual que aparece al hacer clic con el botón derecho en la pestaña de archivo de la ventana principal de la aplicación (*imagen siguiente*).

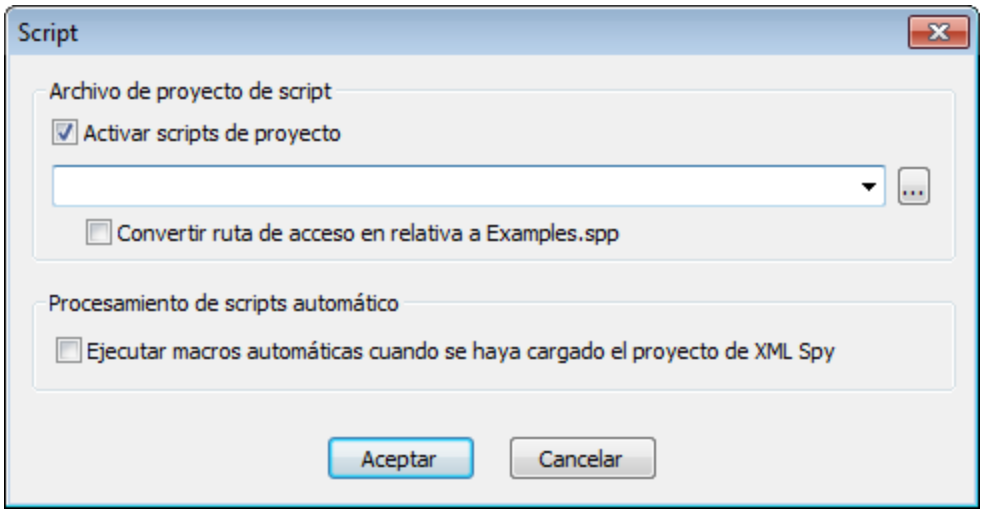


- Cuando otro usuario desprotege un archivo, este archivo no se puede desproteger.
- Cuando desproteja un archivo localmente, puede deshacer la desprotección con el comando **Deshacer desprotección** del menú contextual. Como resultado se devuelve el archivo al servidor sin ningún cambio.
- Si desprotege un archivo en una aplicación de Altova, no lo puede desproteger en otra aplicación de Altova porque se entiende que está desprotegido. A partir de ese momento los comandos disponibles en las aplicaciones de Altova compatibles con Microsoft® SharePoint® Server serían **Proteger** y **Deshacer desprotección**.

28.3.15 Configuración de script

Siga estos pasos para asignar un script de proyecto a un proyecto de XMLSpy:

1. En la IGU de XMLSpy, abra el proyecto de aplicación correspondiente.
2. Seleccione el comando de menú **Proyecto | Configuración de script**. Se abre el cuadro de diálogo Script (*imagen siguiente*).



3. Marque la casilla Activar scripts de proyecto y seleccione el proyecto correspondiente (archivo `.asprj`). Si quiere ejecutar ejecutar macros automáticas cuando se haya cargado el proyecto de XMLSpy, marque la casilla de la parte inferior del cuadro de diálogo.
4. Haga clic en **Aceptar** para finalizar.

Nota: para que el proyecto de script deje de estar asignado a un proyecto de XMLSpy desactive la casilla Activar scripts de proyecto.

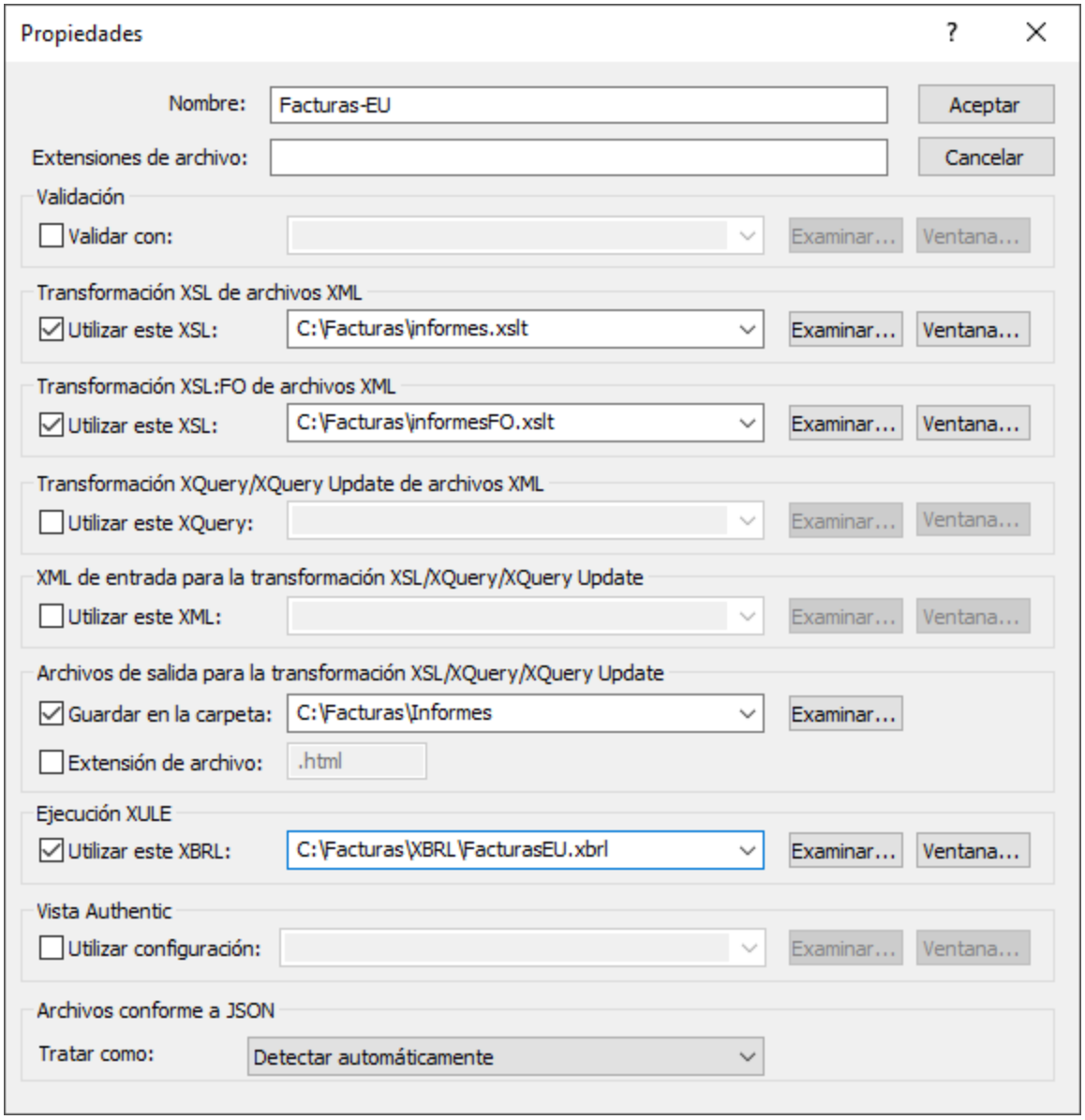
28.3.16 Propiedades



El comando **Proyecto** | **Propiedades** abre el cuadro de diálogo "Propiedades" del proyecto activo (*imagen siguiente*). Si hace clic con el botón derecho en una carpeta de la ventana Proyecto (en lugar de en la carpeta del proyecto), entonces se abre el cuadro de diálogo "Propiedades" de esa carpeta. A continuación describimos las opciones de este cuadro de diálogo.

Nota: si el archivo de proyecto está bajo control de código fuente, entonces aparece un mensaje preguntando si desea desproteger el archivo de proyecto (archivo `.spp`). Haga clic en **Aceptar** si desea editar las propiedades y guardarlas.

© 2016-2022 Altova GmbH



Opciones del cuadro de diálogo

Extensiones de archivo
Este campo se habilita cuando se trata del cuadro de diálogo "Propiedades" de una carpeta y no del proyecto. Cuando se añade un archivo al proyecto, se añadirá a la carpeta donde se definiera su extensión de archivo. Por ejemplo, imagine que añade el archivo `MiInforme.xml` al proyecto. Si definió la extensión de archivo .xml en la carpeta `Facturas-EU`, entonces el archivo `MiInforme.xml` se añadirá a esa carpeta. Si desea añadir archivos XML en varias carpetas distintas, deberá hacerlo a mano carpeta por carpeta.

Usuario y contraseña para carpetas externas
Si se trata de una carpeta externa (incluidas carpetas web externas), en este cuadro de diálogo también puede guardar el id. de usuario y la contraseña necesarios para acceder al servidor.

Validación
Aquí puede indicar qué archivo DTD, esquema XML o esquema JSON[754] debe utilizarse para validar[1337] los archivos que está en la carpeta (o en el proyecto entero).

Transformación XSL de archivos XML
Esta es la hoja de estilos XSLT que se debe utilizar para ejecutar la transformación XSL[1394] de los archivos XML de la carpeta (o del proyecto).

Transformación XSL-FO de archivos XML
Esta es la hoja de estilos XSLT que se debe utilizar para transformar en XSL-FO los archivos XML de la carpeta (o del proyecto).

Transformación XQuery/XQuery Update de archivos XML
El archivo XQuery o XQuery Update que se debe utilizar para ejecutar XQuery o XQuery Update en archivos XML de la carpeta.

XML de entrada para la transformación XSL/XQuery/XQuery Update
El archivo XML que se debe utilizar como entrada para las transformaciones XSLT o para ejecutar XQuery/XQuery Update, con los archivos de la carpeta correspondientes en cada caso.

Archivos de salida para la transformación XSL/XQuery/XQuery Update
El directorio de destino de las transformaciones y, opcionalmente, la extensión de archivo del documento resultante.

Ejecución XULE
El archivo de instancia XBRL que se procesa con el documento XULE que esté activo en la ventana de la aplicación XMLSpy.

Vista Authentic
En este campo puede especificar el archivo SPS que se debe usar para la representación de los archivos XML de la carpeta en la vista Authentic. Recuerde que los archivos XML deben ser válidos según el esquema utilizado para el archivo SPS.

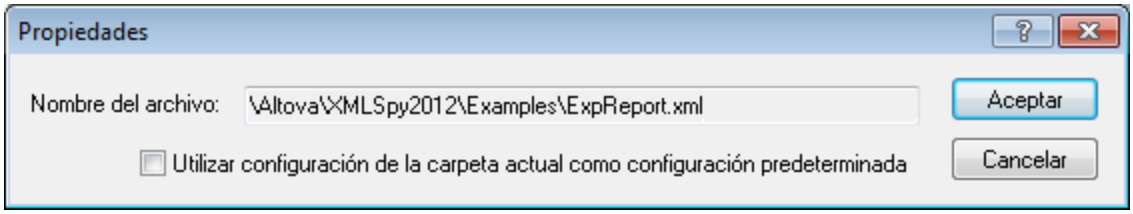
Archivos conforme a JSON
En esta propiedad se indica si una carpeta de proyecto contiene archivos de esquema JSON o archivos de instancia JSON, lo que puede ser útil para identificar archivos de esquema JSON si estos no están claramente identificados como tales con la palabra clave `$schema` y los archivos hacen referencia unos a otros. Puede elegir entre *Instancia JSON*, *Esquema JSON* o *Detectar automáticamente*. La opción predeterminada es *Detectar automáticamente* y hace que XMLSpy compruebe la estructura y el contenido de los archivos JSON para determinar de qué tipo son.

© 2016-2022 Altova GmbH

Notas sobre las propiedades de proyecto

Reglas de prioridad:

- Cuando se llevan a cabo validaciones o transformaciones XSLT/XQuery desde el menú contextual de una carpeta de proyecto, entonces los archivos de validación y transformación especificados en este cuadro de diálogo tienen prioridad sobre los archivos asignados al archivo XML. Además, las opciones definidas para las diferentes carpetas del proyecto tienen prioridad sobre las opciones definidas para carpetas de niveles superiores.
- Si un archivo está presente en varias carpetas del proyecto y se le asignaron archivos de validación o transformación diferentes en estas carpetas, entonces puede especificar qué asignación se utiliza cuando el archivo se procesa fuera del proyecto. Esto se hace de la siguiente manera: busque el archivo de la carpeta del proyecto cuya asignación desea utilizar. Haga clic con el botón derecho en el archivo en la carpeta correspondiente y seleccione **Propiedades** en el menú. Aparece un cuadro de diálogo (*imagen siguiente*). Aquí debe marcar la casilla *Utilizar configuración de la carpeta actual como configuración predeterminada* (la carpeta actual es la carpeta del proyecto donde está el archivo). Si esta casilla está deshabilitada, significa que la configuración de la carpeta actual ya está definida como configuración predeterminada. Recuerde que, si el archivo tiene una asignación local (es decir, la asignación está dentro del archivo propiamente dicho), entonces se utiliza esta asignación local y se ignora la configuración de la carpeta predeterminada.



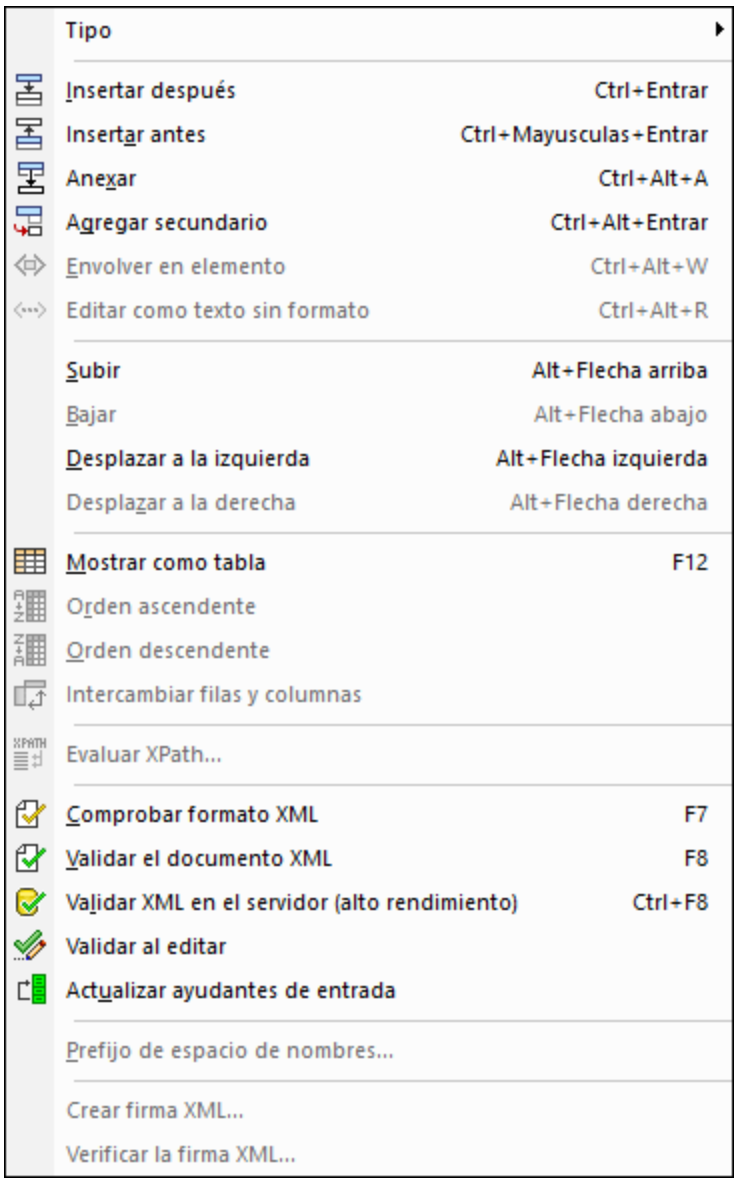
28.3.17 Proyectos usados recientemente

Este comando muestra el nombre de archivo y la ruta de acceso de los últimos nueve proyectos utilizados, para que pueda abrirlos rápidamente.

Recuerde que XMLSpy abre automáticamente el último proyecto utilizado[1584] nada más iniciarse la aplicación (**Herramientas | Opciones | Archivo,** *Proyecto | Abrir el último proyecto al iniciarse el programa*).

28.4 Menú XML

Los comandos del menú **XML** sirven para trabajar con documentos XML. Se trata de comandos para insertar o anexar elementos, modificar la jerarquía de elementos, establecer un prefijo de espacio de nombres y evaluar expresiones XPath en el contexto de un documento XML dado.



Entre las tareas XML más comunes se encuentran la comprobación de formato XML[1335] y la validación[1337] de documentos XML. Los comandos para estas operaciones se encuentran en este menú.

© 2016-2022 Altova GmbH

28.4.1 Tipo

El comando **Tipo** (**Ctrl + Tecla de menú**) muestra un menú que contiene tipos de datos JSON (*imagen siguiente*). Seleccione un tipo del menú para asignarlo a las celdas que estén seleccionadas en la Vista Cuadrícula.

Para más información sobre los tipos de edición consulte Editar el contenido de un documento JSON[686].

28.4.2 Insertar antes/después

Los comandos **Insertar después** e **Insertar antes** se habilitan si la selección actual de la Vista Cuadrícula permite que un componente se inserte. **Insertar después** inserta un componente del mismo tipo que el componente seleccionado debajo de este. **Insertar antes** inserta un componente del mismo tipo que el componente seleccionado encima de este. Si hace doble clic en un componente puede cambiarle el nombre. Para cambiar el tipo de nodo haga clic en el icono del elemento (a la izquierda de su nombre) y seleccione el tipo de nodo que quiera. También puede usar el comando **XML | Tipo**[1333].

Para más información sobre cómo editar la estructura de un documento XML consulte Estructura de un documento[299].

28.4.3 Anexar, Agregar secundario

Los comandos **Anexar** y **Agregar secundario** se habilitan si selecciona un nodo en la vista Cuadrícula. El comando **Anexar** agrega un nodo nuevo como último secundario del elemento seleccionado. El comando **Agregar secundario** anexa un nodo nuevo como secundario. Si hace doble clic en el elemento que acaba de agregar puede cambiarle el nombre. Para cambiar el tipo de nodo haga clic en el icono del elemento (a la izquierda de su nombre) y seleccione el tipo de nodo que quiera. También puede usar el comando **XML | Tipo**[1333].

Para más información sobre cómo editar la estructura de un documento XML consulte Estructura de un documento[299].

28.4.4 Envolver en matriz/objeto

Los comandos **Envolver en matriz** y **Envolver en objeto** envuelven el componente seleccionado en una matriz o un objeto, respectivamente. Para más información sobre cómo editar la estructura de un documento JSON consulte Editar la estructura de un documento JSON[685].

28.4.5 Editar como texto sin formato

El comando **Editar como texto sin formato** se habilita si se selecciona un nodo en la vista Cuadrícula. Este comando permite editar el texto del elemento seleccionado como texto sin formato. Esto puede ser útil, por ejemplo, si está editando contenido complejo, como código HTML.

Para más información consulte la sección Datos XML en la vista Cuadrícula[293].

28.4.6 Mover

Permite mover un componente hacia arriba, abajo, a la izquierda o a la derecha de su ubicación en la cuadrícula. Para ello debe seleccionar el comando correspondiente, que solo se activa si ese movimiento es posible.

28.4.7 Mostrar como tabla

El comando **Mostrar como tabla** se habilita al seleccionar un componente que se repite en la vista Cuadrícula. Este comando permite alternar la visualización de esos elementos que se repiten en las vistas Cuadrícula[169] y Tabla[309]. Este comando permite visualizar elementos que se repiten como una tabla cuyas filas representan cada una de las repeticiones y las columnas los nodos secundarios.

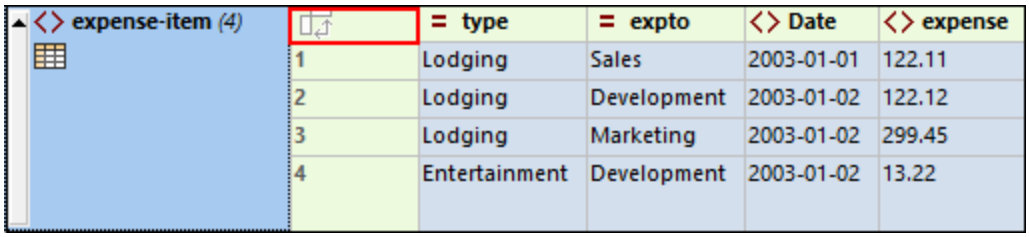
Consulte el apartado Vista Diseño de tabla[692] para más información.

28.4.8 Orden ascendente/descendente

Estos dos comandos se habilitan cuando se selecciona una columna de tabla de un componente. Para seleccionar una columna debe seleccionar su encabezado. El orden se basa en los valores de la columna.

28.4.9 Intercambiar filas y columnas

El comando **Intercambiar filas y columnas** se habilita en la Vista Diseño de tabla[692] si se selecciona la celda superior izquierda de una tabla (*en rojo en la imagen siguiente*). Este comando cambia las filas por las columnas y viceversa.



Consulte el apartado Vista Diseño de tabla[692] para más información.

© 2016-2022 Altova GmbH

28.4.10 Evaluar XPath



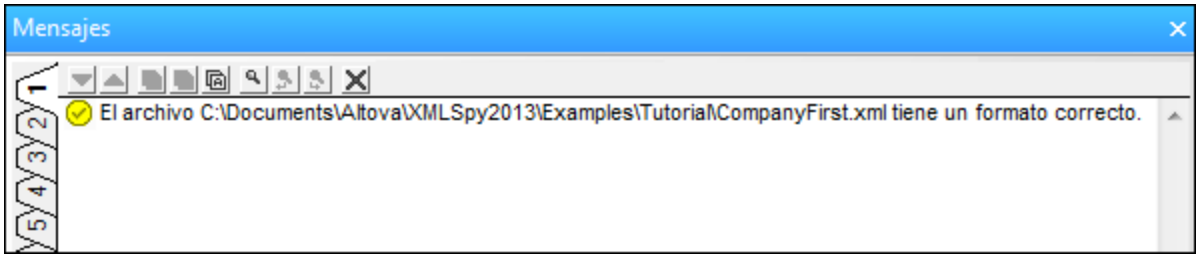
El comando **XML | Evaluar XPath** abre las ventanas de resultados (si no están abiertas ya) y activa la pestaña XPath de las ventanas de resultados[127]. En la pestaña XPath puede evaluar una expresión XPath en el documento activo y ver los resultados.

28.4.11 Comprobar formato XML

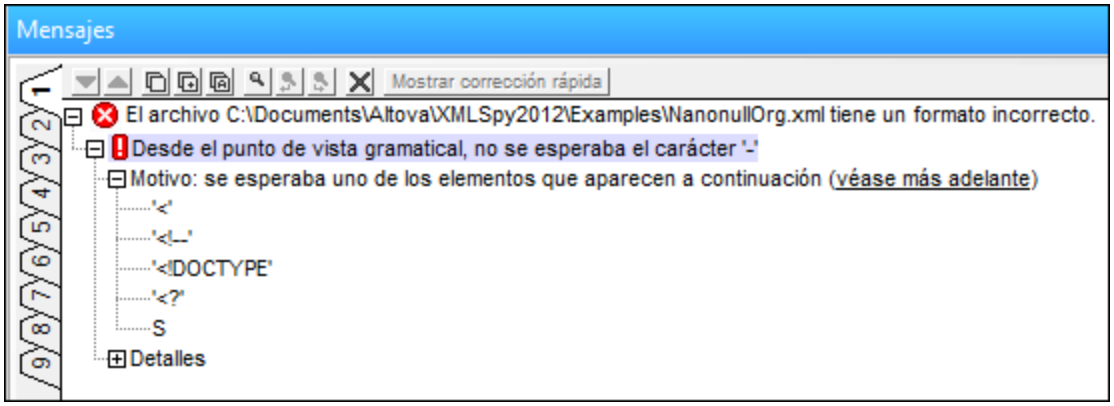
 **F7**

El comando **XML | Comprobar formato XML** (**F7**) revisa el documento activo y comprueba si el formato es correcto según las definiciones de la especificación XML 1.0. Es obligatorio que el documento XML tenga un formato XML correcto. XMLSpy hace una comprobación de formato XML cada vez que se abre o guarda un documento, o cuando se cambia de la vista Texto a otra vista. Puede usar este comando siempre que quiera para comprobar el formato XML de su documento.

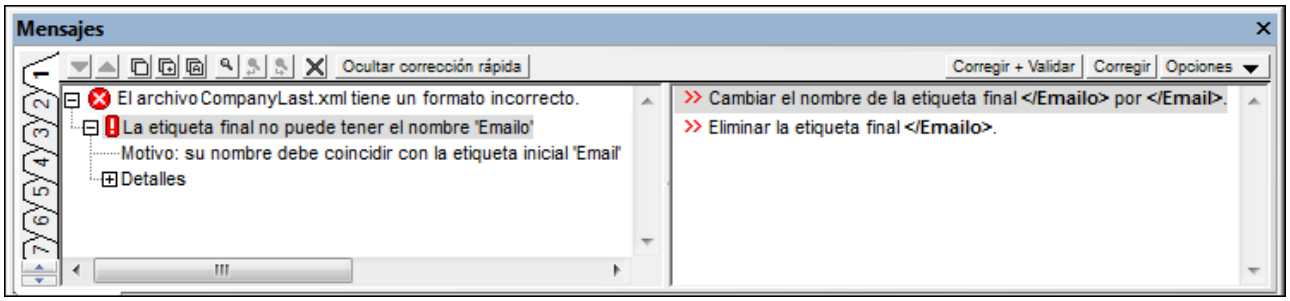
Si el formato XML del documento es correcto, aparece un mensaje parecido a este en la ventana Mensajes:



Si se detecta un error durante la comprobación de formato XML, en la ventana Mensajes aparece un mensaje de error explicando el problema:



A partir de la información disponible en el esquema, XMLSpy le sugiere opciones de corrección rápida si la comprobación de formato XML se realizó en la **vista Texto o Cuadrícula**. Para ver una lista de opciones de corrección rápida, pulse el botón **Mostrar corrección rápida** (*imagen anterior*). En la ventana Mensajes aparece un panel con las opciones de corrección rápida (*imagen siguiente*). Recuerde que la ventana Mensajes muestra los errores uno por uno.



En el panel de corrección rápida seleccione una de las sugerencias y haga clic en el botón **Corregir + Validar** o en el botón **Corregir** (*imagen anterior*). El error se corregirá con la opción seleccionada. Si lo prefiere, también puede hacer doble clic en la opción que más le convenga. Al hacer doble clic en la opción, pueden pasar dos cosas, dependiendo de las opciones elegidas en la lista desplegable *Opciones*: (i) se corrige el error o (ii) se corrige el error y se valida el documento. El comando **Corregir + Validar** es muy práctico porque el validador revisa todo el documento y no sólo detecta errores de validación sino también de formato XML.

Para ocultar el panel de corrección rápida, pulse el botón **Ocultar corrección rápida** (*imagen anterior*).

Nota: la ventana Mensajes tiene nueve pestañas. El resultado de la validación siempre aparece en la pestaña activa. Es decir, puede validar un documento XML en la pestaña 1, pasar a la pestaña 2 y validar otro documento XML y así sucesivamente. De este modo puede comparar los resultados obtenidos y guardados en cada pestaña. Si no cambia de pestaña y valida otro documento, los resultados de la pestaña 1 se sobrescriben con los resultados de la nueva validación.

Validación desde la ventana Proyecto

El comando **Validar el documento XML** también se puede aplicar a un archivo, carpeta o grupo de archivos del proyecto activo. Seleccione el archivo o carpeta en la ventana Proyectos y después haga clic en **XML |**

 © 2016-2022 Altova GmbH

Validar el documento XML (F8). Los archivos no válidos del proyecto se abren en la ventana principal y aparece el mensaje de error "El archivo no es válido".

Nota: la ventana Mensajes tiene nueve pestañas. El resultado de la comprobación de formato siempre aparece en la pestaña activa. Es decir, puede comprobar el formato XML de un documento en la pestaña 1, pasar a la pestaña 2 y comprobar el formato de otro documento XML y así sucesivamente. De este modo puede comparar los resultados obtenidos y guardados en cada pestaña. Si no cambia de pestaña y comprueba el formato de otro documento, los resultados de la pestaña 1 se sobrescribirán con los resultados de la nueva comprobación.

Por lo general, no está permitido guardar un documento XML cuyo formato es incorrecto, pero XMLSpy le permite guardar el documento de todos modos. Esta opción es muy práctica si desea posponer el trabajo y terminar el documento más adelante.

Nota: también puede usar el comando **Comprobar formato XML** con cualquier archivo, carpeta o grupo de archivos de la ventana Proyecto[122]. Para ello haga clic en el archivo, la carpeta o el grupo de archivos y haga clic en el icono **Comprobar formato XML**.

28.4.12 Validar el documento XML

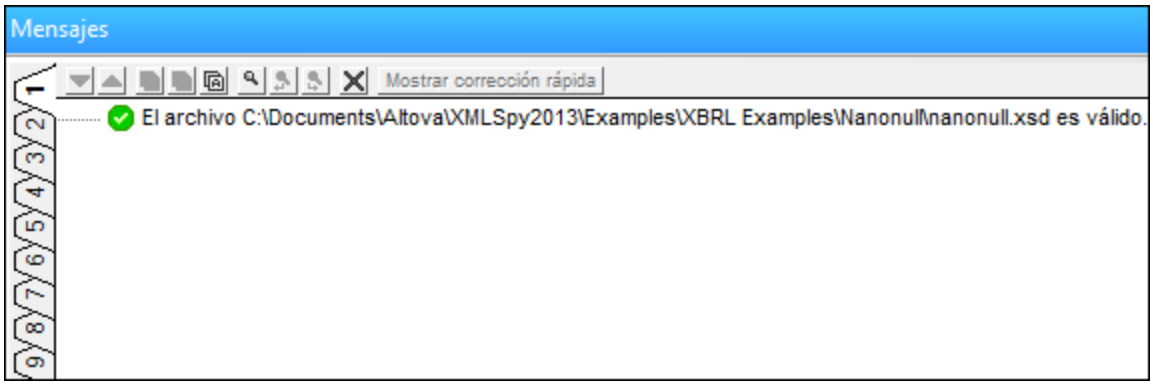


F8

El comando **XML | Validar el documento XML** (**F8**) sirve para validar documentos XML con esquemas XML, DTD y otros esquemas. La validación se lleva a cabo automáticamente al cambiar de la vista Texto a otra vista. Si lo desea, puede configurar la aplicación para que valide automáticamente los documentos al abrirlos o guardarlos (**Herramientas | Opciones | Archivo**). El comando **Validar el documento XML** también realiza una comprobación de formato XML antes de validar el documento. Es decir, si va a validar el documento, no hace falta usar el comando **Comprobar formato XML**[1335] antes del comando **Validar el documento XML**.

Nota: también puede activar esta opción con el comando **Validar al editar**[1344] para ir validando conforme edita los datos.

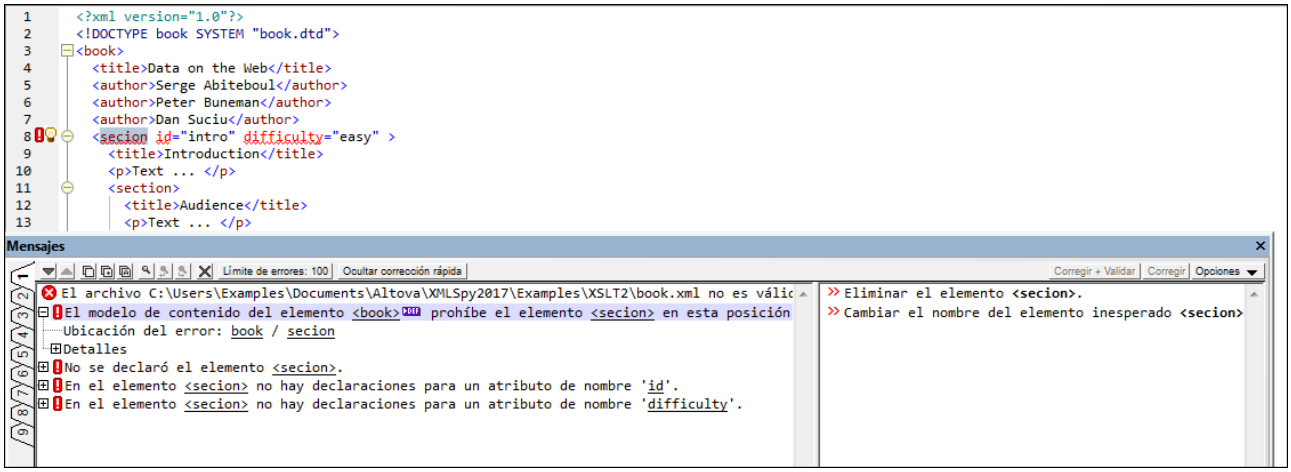
Si el documento es válido, aparece este mensaje en la ventana Mensajes:



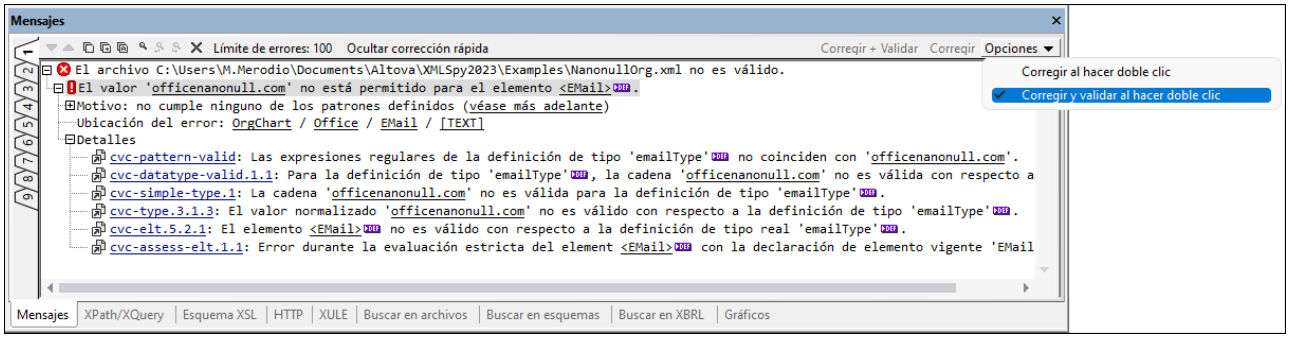
Si el documento no es válido, entonces aparece un mensaje describiendo el error. Para ir al nodo del documento XML que contiene el error, haga clic en el enlace que incluye el mensaje. Más abajo encontrará información sobre los mensajes de error e instrucciones para corregir errores de validación con la función de corrección rápida de XMLSpy.

Errores de validación y cómo corregirlos

Cuando aparece un error de validación en la ventana Mensajes, las causas del error aparecen en el panel izquierdo (imagen siguiente). Si selecciona un motivo de error en el panel izquierdo, en el panel derecho aparecen soluciones para el problema (si existe una solución). Las vistas que ofrecen sugerencias de corrección rápida son la **vista Texto** y la **vista Cuadrícula** y las sugerencias están basadas en datos disponibles en el esquema asociado. Para ver las sugerencias de corrección rápida haga clic en el botón **Mostrar corrección rápida**. Si no quiere ver las sugerencias de corrección rápida haga clic en **Ocultar corrección rápida**. Recuerde que si hay errores de formato XML (como desajustes entre la etiqueta inicial y final), éstos aparecen en la ventana Mensajes antes de los errores de validación. Por tanto, el botón **Mostrar/Ocultar corrección rápida** no se habilita hasta que se encuentra un error de validación (es decir, después de corregir todos los errores de formato XML).

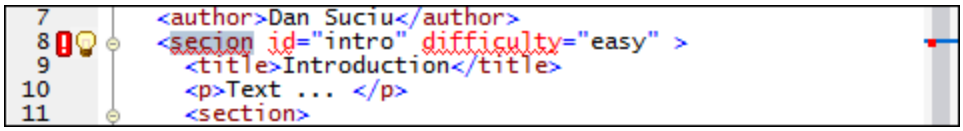


Hay dos maneras de aplicar la corrección rápida: (i) haciendo doble clic en la sugerencia de corrección rápida o (ii) seleccionando la sugerencia y haciendo clic en los comandos **Corregir** o **Corregir+Validar** (*imagen siguiente*). El comando **Corregir + Validar** seguirá validando el documento y detectará el siguiente error (si hay más errores).

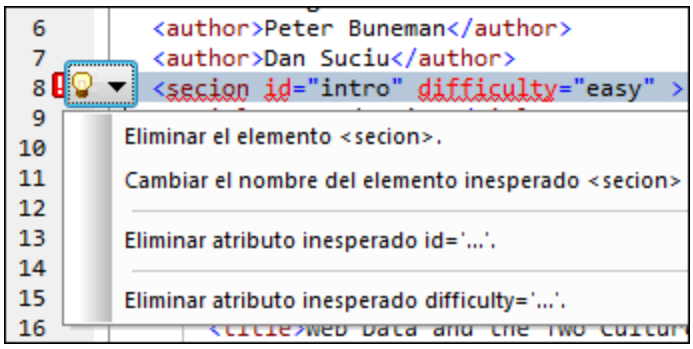


 © 2016-2022 Altova GmbH

La vista Texto incluye dos indicadores visuales más para señalar errores de validación (*imagen siguiente*): un (i) signo de exclamación rojo que aparece en el margen de numeración de líneas y un (ii) cuadrado rojo que aparece en la barra de desplazamiento (en el lado derecho de la ventana).



El icono en forma de bombilla que aparece junto al signo de exclamación rojo (*imagen anterior*) es el icono de corrección rápida. Si pasa el cursor por encima de este icono, aparecen todas las sugerencias de corrección rápida del error (*imagen siguiente*). Seleccione una sugerencia para aplicar la corrección rápida.



Nota: los indicadores de errores de validación y sugerencias de corrección rápida que se describen más arriba solamente se actualizan cuando se ejecuta el comando **XML | Validar el documento XML (F8)** y no se actualizan en segundo plano. Es decir, tras corregir un error de validación, vuelva a ejecutar el comando **Validar el documento XML (F8)** para asegurarse de que el error está solucionado.

Nota: la ventana Mensajes tiene nueve pestañas. El resultado de la validación siempre aparece en la pestaña activa. Es decir, puede validar el documento XML en la pestaña 1, pasar a la pestaña 2 y validar otro documento XML y así sucesivamente. De este modo puede comparar los resultados obtenidos y guardados en cada pestaña. Si no cambia de pestaña y valida otro documento, los resultados de la pestaña 1 se sobrescribirán con los resultados de la nueva validación.

Validación desde la ventana Proyecto

El comando **Validar el documento XML** también se puede aplicar a un archivo, una carpeta o un grupo de archivos del proyecto activo. En la ventana Proyecto seleccione el archivo, la carpeta o el grupo de archivos que desea validar. Después haga clic en **XML | Validar el documento XML** o pulse **F8**. Los archivos que no sean válidos se abren en la ventana principal y aparece un mensaje de tipo *El archivo no es válido* en la ventana Mensajes.

Validar documentos XML

Para validar un archivo XML, asegúrese de que el archivo está activo en la ventana principal y haga clic en **XML | Validar el documento XML** o pulse **F8**. El archivo XML se valida con el esquema al que se hace referencia en el archivo. Si no hay una referencia a un esquema, aparece un mensaje de error en la ventana Mensajes. Mientras el documento XML está abierto, el esquema se guarda en memoria (consulte el comando Vaciar caché en memoria[1369] del menú DTD/Esquema).

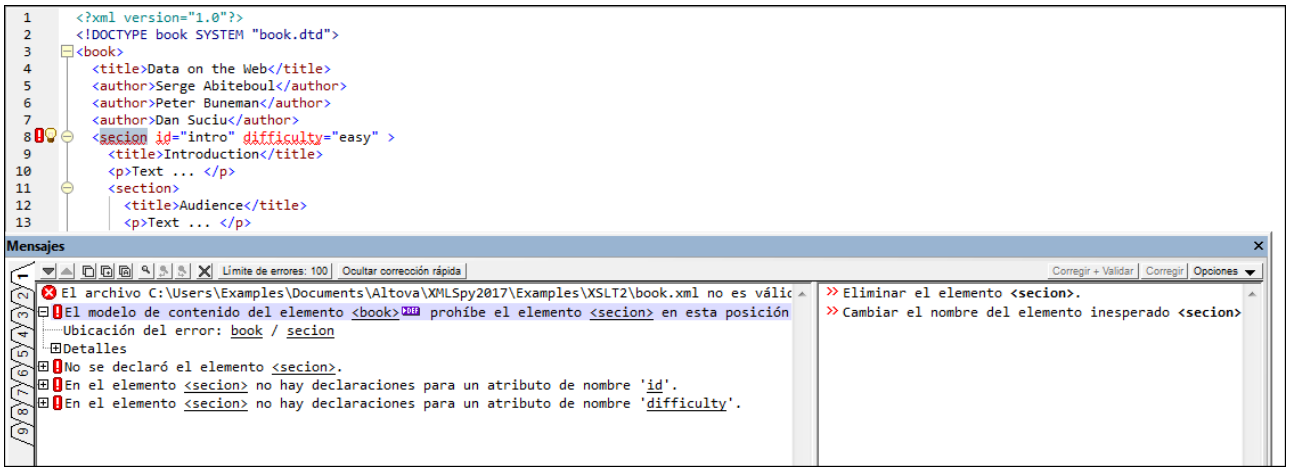
Validar documentos de esquema (archivos DTD y esquemas XML)

XMLSpy es compatible con los principales dialectos de esquema, incluidos DTD y XSD. Para validar un documento de esquema, asegúrese de que el documento está activo en la ventana principal y haga clic en **XML | Validar el documento XML** o pulse **F8**.

Mensajes de validación

Hay dos tipos de mensajes de validación:

- Si el esquema (la DTD o el esquema XML) es válido, aparece un mensaje de validación de tipo *El archivo es válido* en la ventana Mensajes.
- Si el esquema no es válido, aparece un mensaje de error en la ventana Mensajes (*imagen siguiente*).



En cada mensaje aparecen todos los motivos posibles del error. Por ejemplo, el error de la imagen anterior puede deberse a cuatro motivos. El primer motivo está expandido y los demás están contraídos. Por cada posible motivo se dan tres informaciones:

1. Una descripción del motivo que contiene enlaces a la correspondiente definición del documento de esquema asociado. Con este enlace puede ir a la definición de esquema correspondiente y ver por qué el documento no es válido.
2. La ruta de acceso del nodo que causó el error en el documento XML. Haga clic en cualquier nodo de esta ruta de acceso para resaltar el nodo en el documento.
3. Información detallada sobre el error y un enlace al párrafo correspondiente de la especificación de esquema. Aquí es donde aparecen las reglas de esquema que determinan la validez del documento.

Nota: si está validando el documento en la vista Texto y hace clic en un enlace de la ventana Mensajes, la definición correspondiente se resaltará en la vista Texto. Pero si está validando el documento en la vista Esquema y hace clic en un enlace, la definición se abrirá en la vista Esquema, donde podrá editar el componente directamente.

Catálogos

XMLSpy es compatible con un subconjunto del mecanismo de catalogación XML OASIS. El mecanismo de catalogación permite a XMLSpy recuperar de carpetas locales del usuario los esquemas (y hojas de estilos y otros archivos) usados con frecuencia. Esto incrementa la velocidad global de procesamiento, permite al

© 2016-2022 Altova GmbH

usuario trabajar sin conexión (es decir, sin estar conectado a una red) y mejora la portabilidad de los documentos (porque los identificadores URI se tienen que cambiar sólo en los archivos de catálogo).

A continuación describimos cómo funciona el mecanismo de catalogación en XMLSpy.

- XMLSpy carga un archivo llamado `RootCatalog.xml` que contiene una lista de los archivos de catálogo que se buscarán. El usuario puede modificar esta lista y añadir tantos archivos de catálogo como desee, escribiendo cada archivo en un elemento `nextCatalog`.
- XMLSpy busca cada uno de estos archivos de catálogo y sus URI se resuelven de acuerdo con las asignaciones especificadas en ellos. Recuerde que si la misma asignación aparece dos veces se pueden producir errores.
- XMLSpy viene con dos archivos de catálogo. Estos dos archivos y su funcionamiento se describen en el apartado Catálogos en XMLSpy 456.
- El identificador `PUBLIC` o `SYSTEM` de la instrucción `DOCTYPE` de su archivo XML se usará para la búsqueda en el catálogo. Para los esquemas más comunes se suele predefinir el identificador `PUBLIC` porque así solamente es necesario cambiar el URI en el archivo de catálogo cuando los documentos XML se usan en varios equipos.

Cuando escriba su archivo `CustomCatalog.xml` (o algún otro archivo de catálogo), use solamente el subconjunto del catálogo OASIS que aparece a continuación para que XMLSpy pueda procesar correctamente el catálogo. Cada uno de los elementos del subconjunto pueden llevar el atributo `xml:base`, que se usa para especificar el URI base del elemento.

```
<catalog...>
...
<public publicId="PublicID of Resource" uri="URL of local file"/>
<system systemId="SystemID of Resource" uri="URL of local file"/>
<rewriteURI uriStartString="StartString of URI to rewrite" rewritePrefix="String to
replace StartString"/>
<rewriteSystem systemIdStartString="StartString of SystemID" rewritePrefix="Replacement
string to locate resource locally"/>
<uri name="filename" uri="URL of file identified by filename"/>
...
</catalog>
```

Nota:

- El archivo `catalog.xml` de la carpeta `%AltovaCommonFolder%\Schemas\schema` 456 incluye referencias a las DTD que implementan especificaciones antiguas de XML Schema. No recomendamos que valide sus esquemas XML con estos esquemas. Los archivos DTD referenciados se incluyen con el único propósito de XMLSpy pueda mostrar información en los ayudantes de entrada, en caso de que desee crear documentos basados en estas recomendaciones antiguas (véase el siguiente punto).
- Si crea una extensión de archivo personal para un esquema concreto (por ejemplo, la extensión `.mihtml` para archivos HTML que deben ser válidos con respecto a la DTD HTML), entonces puede habilitar la edición inteligente para los archivos que tengan esta extensión con solo añadir una línea de texto al archivo `CustomCatalog.xml`. Por ejemplo, en el caso de la extensión `.mihtml`, añada el elemento `<spy:fileExtHelper ext="myhtml" uri="schemas/xhtml/xhtml1-transitional.dtd"/>` como secundario del elemento `<catalog>`. Esto habilita la edición inteligente (finalización automática, ayudantes de entrada, etc.) de los archivos `.mihtml` en XMLSpy de acuerdo con la DTD XHTML 1.0 Transitional.
- Para más información, consulte la especificación XML Catalogs.

Automatizar la validación de documentos XML con RaptorXML 2023

RaptorXML es una aplicación independiente de Altova para validar XML, transformar XSLT y ejecutar XQuery. Se puede usar desde la línea de comandos, mediante la interfaz de COM, en programas Java y en aplicaciones .NET.

Por tanto, puede automatizar sus tareas de validación con ayuda de RaptorXML. Por ejemplo, puede crear un archivo de proceso por lotes que llame a RaptorXML para ejecutar la validación en un conjunto de documentos y que envíe el resultado de la validación a un archivo de texto. Para más información, consulte la documentación de RaptorXML en nuestro sitio web.

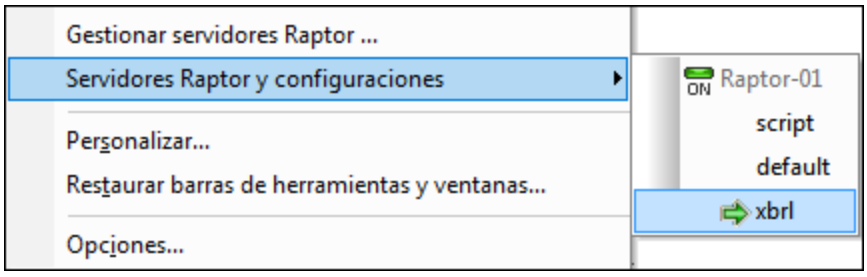
28.4.13 Validar XML en el servidor (alto rendimiento)

 **Ctrl+F8**

El comando **XML | Validar XML en el servidor (alto rendimiento) (Ctrl+F8)** valida el documento XML activo con el servidor RaptorXML que esté activo[1563] y su configuración activa[1563]. El comando lleva a cabo la validación inmediatamente y los resultados de la validación aparecen en la ventana Mensajes.

Nota: el rendimiento real dependerá del número de núcleos de procesador que utilice RaptorXML Server para la validación. Cuanto mayor sea el número de núcleos utilizados, más rápido será el procesamiento.

Si definió varias configuraciones en varios servidores, podrá seleccionar un servidor y una de sus configuraciones como configuración activa. La configuración activa se usará para todas las validaciones siguientes. Pase el puntero por encima del comando **Herramientas | Servidores Raptor y configuraciones** (*imagen siguiente*) para abrir el submenú que contiene todos los servidores que se añadieron, así como sus configuraciones. Seleccione la configuración de servidor que desea usar como configuración activa. Por ejemplo, en la imagen siguiente, se seleccionó la configuración `xbrl` del servidor `Raptor-01`. La fecha verde indica que se trata de la configuración activa.



El comando **Validar XML en el servidor (alto rendimiento) (Ctrl+F8)** también se puede invocar desde el ayudante de entrada Proyecto. Haga clic con el botón derecho en el proyecto, en una carpeta o en un archivo y seleccione **Validar XML en el servidor** para validar los datos XML o XBRL que estén en el objeto seleccionado.

Nota: la validación con RaptorXML se puede utilizar desde las vistas Texto, Cuadrícula y XBRL.

© 2016-2022 Altova GmbH

28.4.14 Validar archivos WSDL

Un documento WSDL también es un documento XML, por lo que se puede validar como XML o como WSDL. Esta lista contiene información importante sobre la validación de documentos WSDL[1337] en las ediciones Enterprise y Professional de XMLSpy:

- La edición Professional Edition realiza una validación de esquema básica. Es decir, trata el archivo WSDL como si fuera un archivo XML y lo valida con el esquema definido en `http://schemas.xmlsoap.org/wsdl/`.
- La edición Enterprise Edition realiza una validación más avanzada. En lugar de validar el archivo con el esquema http://schemas.xmlsoap.org/wsdl/, la edición Enterprise Edition usa el documento `http://www.altova.com/specs_wsdl.html#_document-s`, así como su propia lógica, para ofrecer más información en el contexto de WSDL. Por tanto, el mismo archivo WSDL puede ser válido en la edición Professional y no ser válido en la edición Enterprise (*véase el ejemplo*).
- Hay una diferencia entre `http://schemas.xmlsoap.org/wsdl/` y `http://www.altova.com/specs_wsdl.html#_document-s`. En el primero faltan las definiciones de los elementos de extensibilidad (la especificación y el esquema deberían ser idénticos pero no lo son. Esta diferencia parece ser un error del esquema W3C); este problema se resuelve en el esquema más reciente (que usa la edición Enterprise Edition).
- La edición Professional Edition usa `http://schemas.xmlsoap.org/wsdl/` para la validación, por lo que todos los elementos de extensibilidad se considerarán no válidos en la edición Professional Edition pero serán válidos en la edición Enterprise Edition, que usa `http://www.altova.com/specs_wsdl.html#_document-s`.
- Al ser el esquema W3C un esquema oficial publicado por el grupo de trabajo del W3C, los errores de este esquema desgraciadamente están fuera de nuestro control .

Ejemplo:

A continuación aparece un fragmento de un archivo WSDL. Observe que se ha declarado el elemento `getCityTime`. A este elemento se le hace referencia por error como `getCityTimes`. La edición Enterprise Edition comprueba si los elementos referenciados se declararon previamente en el archivo. La edición Profesional Edition no hace esto. Por tanto, en la edición Professional Edition este archivo sería válido (suponiendo que el resto del archivo es válido), pero en la edición Enterprise Edition no lo sería (suponiendo que en efecto `getCityTimes` no se declaró en otra parte del archivo).

```
<s:element name="getCityTime">
   <s:complexType>
      <s:sequence>
         <s:element minOccurs="0" maxOccurs="1" name="city" type="s:string"/>
      </s:sequence>
   </s:complexType>
</s:element>
<s:element name="abc">
   <s:complexType>
      <s:sequence>
         <s:element ref="getCityTimes"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

28.4.15 Validar al editar

El comando **Validar al editar** activa y desactiva el modo del mismo nombre, que valida el documento conforme usted teclea. Puede usar este modo tanto en la Vista Texto[145] como en la Vista Cuadrícula JSON[689] o en la Vista Authentic. También puede activar o desactivar este modo con el botón **Validar al editar** de la barra de herramientas o con la opción *Al editarlos* de la sección Archivo del cuadro de diálogo "Opciones"[1584].

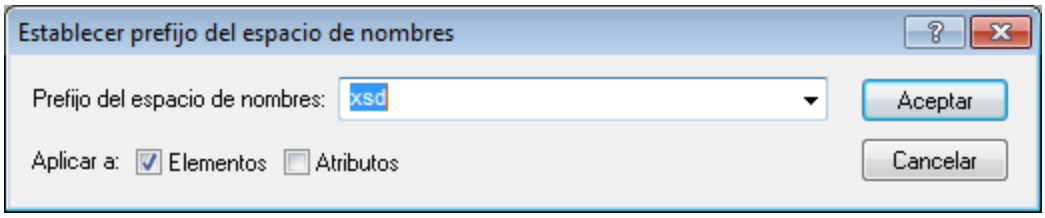
28.4.16 Actualizar ayudantes de entrada



El comando **Actualizar ayudantes de entrada** actualiza las ventanas de los ayudantes de entrada, es decir, se vuelve a cargar la DTD o el esquema subyacente. Si realizó cambios en un esquema XML o una DTD en la que se basa un documento XML que está abierto, recomendamos que actualice los ayudantes de entrada para que las opciones de edición inteligente reflejen los cambios realizados en el esquema.

28.4.17 Prefijo de espacio de nombres

El comando **XML | Prefijo de espacio de nombres** está disponible en la vista Cuadrícula solamente y abre un cuadro de diálogo donde puede definir el prefijo de espacio de nombres del elemento o atributo seleccionado. Si se trata de un elemento, el prefijo de espacio de nombres se aplica también a los elementos descendientes.

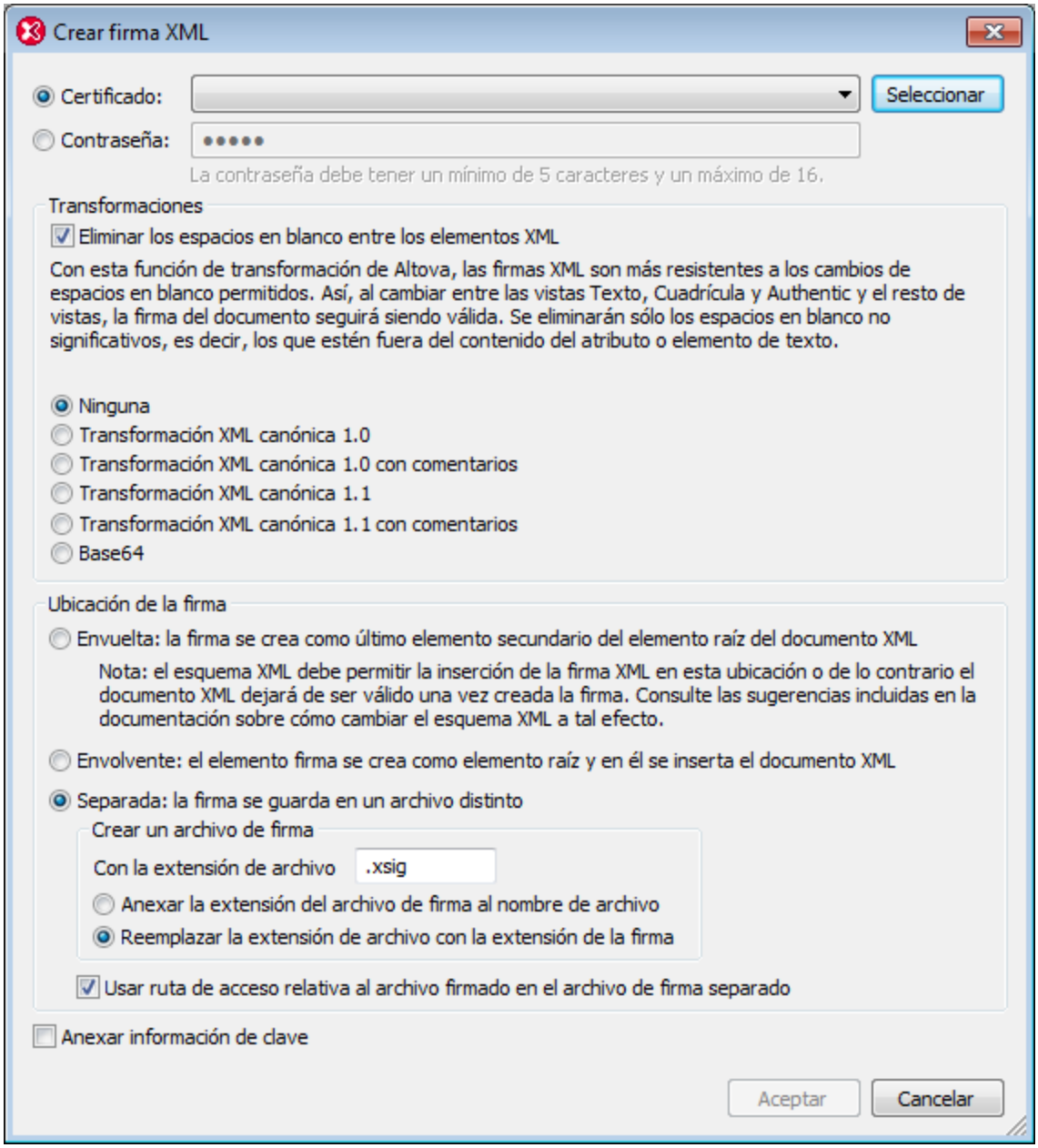


Puede indicar si el prefijo de espacio se aplica a elementos, atributos o ambos. El prefijo de espacio de nombres se aplica al elemento o atributo seleccionado. Si se trata de un elemento, se aplica también a los nodos descendientes del elemento.

28.4.18 Crear firma XML

El comando **Crear firma XML** está disponible en la vista Texto, Cuadrícula, Esquema, WSDL y XBRL y permite crear una firma XML para el documento XML activo. Al hacer clic en este comando se abre el cuadro de diálogo "Crear firma XML" (*imagen siguiente*), cuyas opciones se describen a continuación.

© 2016-2022 Altova GmbH



Método de autenticación: certificado o contraseña

La firma XML puede estar basada en un certificado o en una contraseña. Seleccione la opción que desea usar.

- *Certificado:* si desea usar un certificado, este debe tener una clave privada y debe estar ubicado en un almacén de certificados 420 al que se pueda tener acceso. La firma se genera usando la clave privada del certificado. Para verificar la firma, es necesario tener acceso al certificado (o a su versión de clave pública). La clave pública del certificado se usa para verificar la firma. Para seleccionar el certificado de clave privada-pública que desea usar, haga clic en el botón **Seleccionar** y busque el certificado. Para más información sobre el uso de los certificados, consulte el apartado Trabajar con certificados 419.

- *Contraseña:* escriba una contraseña de como mínimo cinco caracteres y 16 como máximo. A partir de este momento esta contraseña será necesaria para verificar la firma.

Transformaciones

Los datos XML se transforman y el resultado de la transformación se usa para crear la firma. Puede indicar qué algoritmo de canonización se aplica a los datos XML del archivo (al contenido de `SignedInfo`) antes de realizar los cálculos de la firma. Estas son las opciones disponibles:

- *Transformación XML canónica con o sin comentarios:* si se incluyen comentarios para el cálculo de la firma y se realizan cambios en los comentarios de los datos XML, entonces se produce un error de verificación. Si no se incluyen comentarios para el cálculo de la firma, puede incluir comentarios en el documento XML después de firmarlo sin que se produzcan errores de verificación como resultado.
- *Base64:* se entiende que el elemento raíz (o de documento) del documento XML tiene codificación Base64 y se lee en formato binario. Si el elemento raíz no tiene codificación Base64, entonces se produce un error o el elemento se considera vacío, dependiendo del tipo de elemento encontrado.
- *Ninguna:* no se realiza ninguna transformación y los datos XML del archivo binario guardados en el disco se pasan directamente para crear la firma. Si posteriormente se realizan cambios en los datos, entonces se produce un error y la firma no se puede verificar. No obstante, si marca la casilla *Eliminar los espacios en blanco,* entonces se eliminan todos los espacios en blanco y los cambios en los espacios en blanco no se tienen en cuenta. La principal diferencia entre usar la opción *Ninguna* y usar una opción *Transformación...* es que la canonización produce una secuencia de datos XML en la que se normalizan algunas diferencias, como el orden de los atributos. Por consiguiente, la transformación canónica normalizará determinados cambios, como cambios en el orden de atributos (y la firma se verifica correctamente). En cambio, si no usa ninguna transformación, este tipo de cambios se detecta y se produce un error de verificación.

Ubicación de la firma

La firma se puede colocar dentro del archivo XML o en un archivo separado. Estas son las opciones disponibles:

- *Envuelta:* se crea el elemento `Signature` como último elemento secundario del elemento raíz (de documento).
- *Envolvente:* se crea el elemento `Signature` como elemento raíz (de documento) y el documento XML se inserta como elemento secundario de la firma.
- *Separada:* la firma XML se crea en un archivo separado. En este caso, puede indicar la extensión de archivo del archivo de firma y si el nombre del archivo se crea: (i) anexando la extensión al nombre del archivo XML (por ejemplo, `test.xml.xsig`) o (ii) sustituyendo la extensión XML del archivo XML con la nueva extensión (por ejemplo, `test.xsig`). También puede indicar si la referencia al archivo XML en el archivo de firma debe ser una ruta de acceso relativa o absoluta.

Nota: las firmas para archivos de esquema XML (`.xsd`) y archivos XBRL solamente se pueden crear en archivos externos. Las firmas para archivos WSDL se pueden crear en archivos externos y después incluirse en el archivo WSDL.

Nota: si crea una firma XML en un archivo separado, el archivo XML y la firma están asociados por medio de una referencia incluida en el archivo de la firma. En estos casos, para verificar la firma es el archivo de firma el que debe estar activo y no el archivo XML.

© 2016-2022 Altova GmbH

Anexar información de clave

La casilla *Anexar información de clave* se habilita si la firma se basa en un certificado. Si la firma se basa en una contraseña, entonces no está disponible esta opción.

Si marca esta casilla, la información de la clave pública se coloca dentro de la firma. De lo contrario, la información de la clave no se incluye en la firma. La ventaja de incluir la información de la clave es que durante el proceso de verificación no será necesario tener acceso al certificado propiamente dicho (en concreto, la información de clave pública que contiene), puesto que la información de clave ya está presente en la firma.

28.4.19 Verificar la firma XML

La firma XML se verifica correctamente si no se realizaron cambios en el archivo XML desde que se firmó. Si se realizaron cambios, la verificación fallará. En XMLSpy las firmas XML se pueden verificar en varias circunstancias:

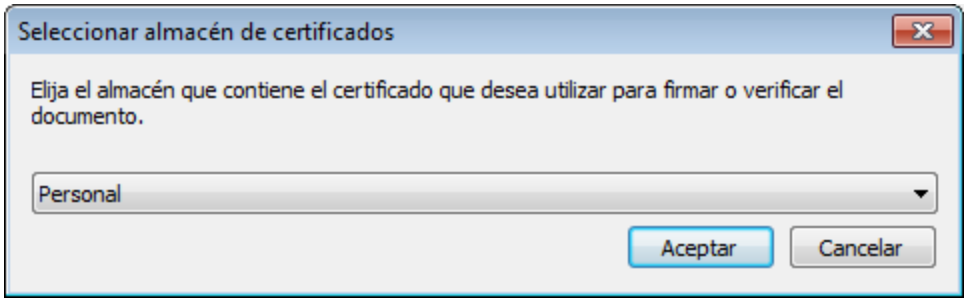
- El archivo XML contiene una firma basada en un certificado y la información de clave está incluida en la firma 1347
- El archivo XML contiene una firma basada en un certificado y la información de clave no está incluida en la firma 1347
- La firma está basada en un certificado y está en un archivo externo que incluye la información de clave 1348
- La firma está basada en un certificado y está en un archivo externo que no incluye la información de clave 1349
- El archivo XML contiene una firma basada en una contraseña 1349
- La firma está basada en una contraseña y está en un archivo externo 1349

El archivo XML contiene una firma basada en un certificado y la información de clave está incluida en la firma

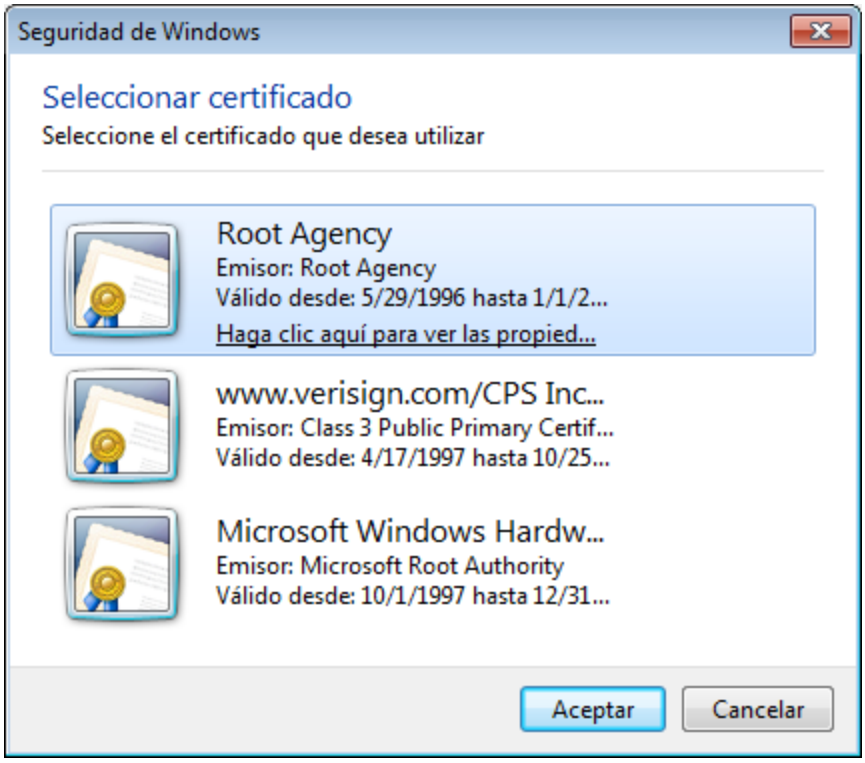
Para verificar una firma XML de estas características, abra el archivo XML en XMLSpy y asegúrese de que este archivo es el documento activo. Al hacer clic en el comando **XML | Verificar la firma XML** se ejecuta el proceso de verificación y el resultado aparece en la ventana de mensajes (aparece un mensaje que confirma si la verificación se realizó correctamente o si se produjo un error de verificación).

El archivo XML contiene una firma basada en un certificado y la información de clave no está incluida en la firma

Si una firma basada en un certificado no incluye la información de clave, XMLSpy solicita el certificado donde se puede leer la información de clave necesaria para verificar la firma. Para verificar la firma es necesario que el archivo XML sea el archivo activo en XMLSpy. Al hacer clic en el comando **XML | Verificar la firma XML**, aparece un diálogo donde debe seleccionar el almacén de certificados 419 donde está almacenado el certificado (*imagen siguiente*).



Después de seleccionar el almacén de certificados[419], haga clic en **Aceptar**. Aparece un cuadro de diálogo con todos los certificados del almacén seleccionado (*imagen siguiente*). Seleccione el certificado que desea usar en el proceso de verificación y haga clic en **Aceptar**.



A continuación se ejecuta el proceso de verificación y el resultado aparece en la ventana de mensajes.

La firma está basada en un certificado y está en un archivo externo que incluye la información de clave

Si una firma XML basada en un certificado está en un archivo externo, la firma se verifica con el archivo de firma que esté activo en XMLSpy. Al hacer clic en el comando **XML | Verificar la firma XML** se ejecuta el proceso de verificación y el resultado aparece en la ventana de mensajes (aparece un mensaje que confirma si la verificación se realizó correctamente o si se produjo un error de verificación).

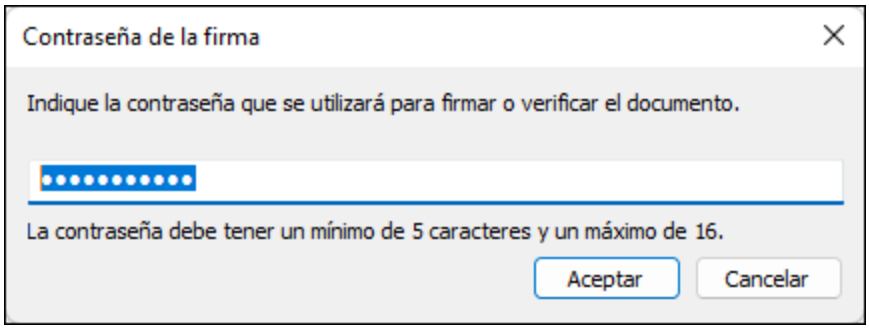
© 2016-2022 Altova GmbH

La firma está basada en un certificado y está en un archivo externo que no incluye la información de clave

Si una firma XML basada en un certificado está en un archivo externo, la firma se verifica con el archivo de firma que esté activo en XMLSpy. Al hacer clic en el comando **XML | Verificar la firma XML**, XMLSpy solicita el certificado donde se puede leer la información de clave necesaria para verificar la firma. Seleccione el certificado (siguiendo los pasos descritos en el párrafo *El archivo XML contiene una firma basada en un certificado y la información de clave no está incluida en la firma*[1347]). El proceso de verificación se ejecuta y el resultado aparece en la ventana de mensajes (aparece un mensaje que confirma si la verificación se realizó correctamente o si se produjo un error de verificación).

El archivo XML contiene una firma basada en una contraseña

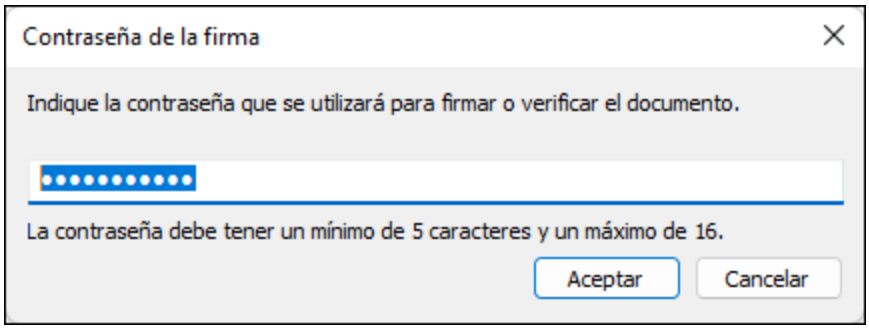
Si el archivo XML tiene una firma basada en una contraseña, la firma se verifica con el archivo XML que esté activo en XMLSpy. Al hacer clic en el comando **XML | Verificar la firma XML** aparece un cuadro de diálogo donde debe introducir la contraseña (*imagen siguiente*).



Introduzca la contraseña, que debe tener un mínimo de cinco caracteres y un máximo de 16, y haga clic en **Aceptar**. El proceso de verificación se ejecuta y el resultado aparece en la ventana de mensajes (aparece un mensaje que confirma si la verificación se realizó correctamente o si se produjo un error de verificación).

La firma está basada en una contraseña y está en un archivo externo

Si una firma XML basada en una contraseña está en un archivo externo, la firma se verifica con el archivo de firma que esté activo en XMLSpy. Al hacer clic en el comando **XML | Verificar la firma XML** aparece un cuadro de diálogo donde debe introducir la contraseña (*imagen siguiente*).

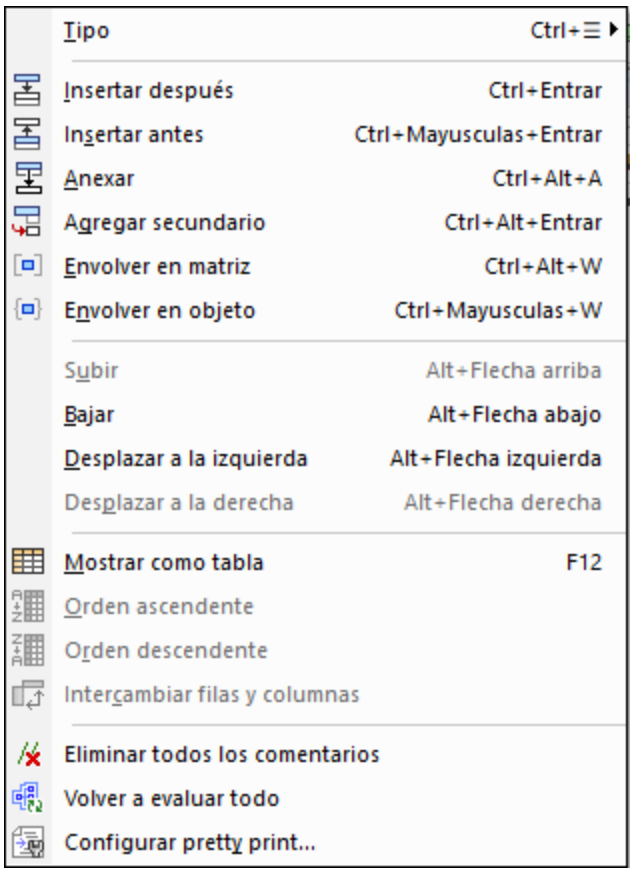


Introduzca la contraseña, que debe tener un mínimo de cinco caracteres y un máximo de 16, y haga clic en **Aceptar**. El proceso de verificación se ejecuta y el resultado aparece en la ventana de mensajes (aparece un mensaje que confirma si la verificación se realizó correctamente o si se produjo un error de verificación).

© 2016-2022 Altova GmbH

28.5 Menú JSON

El menú **JSON** contiene los comandos que se suelen utilizar cuando se trabaja con documentos JSON. La mayoría de estos comandos se usan cuando se trabaja en la Vista Cuadrícula JSON[679]. Si algún comando no se puede aplicar en el punto en el que se encuentra el cursor, entonces aparecerá como deshabilitado.



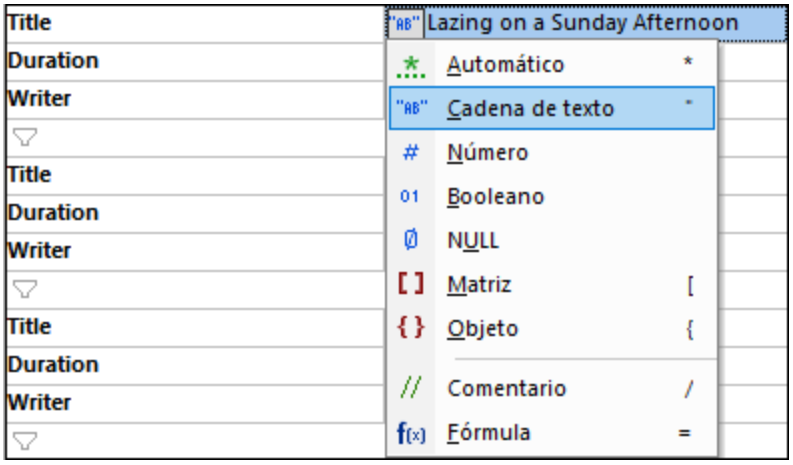
Los comandos de este menú se describen en las subsecciones de esta sección:

- Tipo[1351]
- Insertar, anexar, agregar secundario[1352]
- Envolver en matriz/objeto[1352]
- Mover[1353]
- Mostrar como tabla[1353]
- Orden ascendente/descendente[1353]
- Intercambiar filas y columnas[1353]
- Eliminar comentarios, volver a evaluar todo, Pretty-Print[1354]

28.5.1 Tipo

El comando **Tipo** (**Ctrl + Tecla de menú**) muestra un menú que contiene tipos de datos JSON (*imagen siguiente*). Seleccione un tipo del menú para asignarlo a las celdas que estén seleccionadas en la Vista

Cuadrícula. (La tecla **Menú** suele estar ubicada en la parte inferior derecha del teclado, junto a la tecla **Ctrl**. Su icono suele ser parecido a este:).



Para más información sobre los tipos de edición consulte Editar el contenido de un documento JSON[686].

28.5.2 Insertar, anexar, agregar secundario

Los comandos **Insertar**, **anexar** y **agregar secundario** se habilitan si la selección actual de la Vista Cuadrícula permite que un componente se inserte, anexe o añada como secundario.

- **Insertar después** inserta un componente del mismo tipo que el componente seleccionado debajo de este.
- **Insertar antes** inserta un componente del mismo tipo que el componente seleccionado encima de este.
- El comando **Anexar** anexa un componente del mismo tipo que el componente seleccionado en una fila de la cuadrícula por debajo de este.
- El comando **Agregar secundario** añade un componente secundario nuevo como último elemento. El tipo será el mismo que el del secundario que era el último hasta entonces.

Para más información sobre cómo editar la estructura de un documento JSON consulte Editar la estructura de un documento JSON[685].

28.5.3 Envolver en matriz/objeto

Los comandos **Envolver en matriz** y **Envolver en objeto** envuelven el componente seleccionado en una matriz o un objeto, respectivamente. Para más información sobre cómo editar la estructura de un documento JSON consulte Editar la estructura de un documento JSON[685].

© 2016-2022 Altova GmbH

28.5.4 Mover

Permite mover un componente hacia arriba, abajo, a la izquierda o a la derecha de su ubicación en la cuadrícula. Para ello debe seleccionar el comando correspondiente, que solo se activa si ese movimiento es posible.

28.5.5 Mostrar como tabla

El comando **Mostrar como tabla** se habilita al seleccionar un componente que se repite en la vista Cuadrícula. Este comando permite alternar la visualización de esos elementos que se repiten en las vistas Cuadrícula[169] y Tabla[309]. Este comando permite visualizar elementos que se repiten como una tabla cuyas filas representan cada una de las repeticiones y las columnas los nodos secundarios.

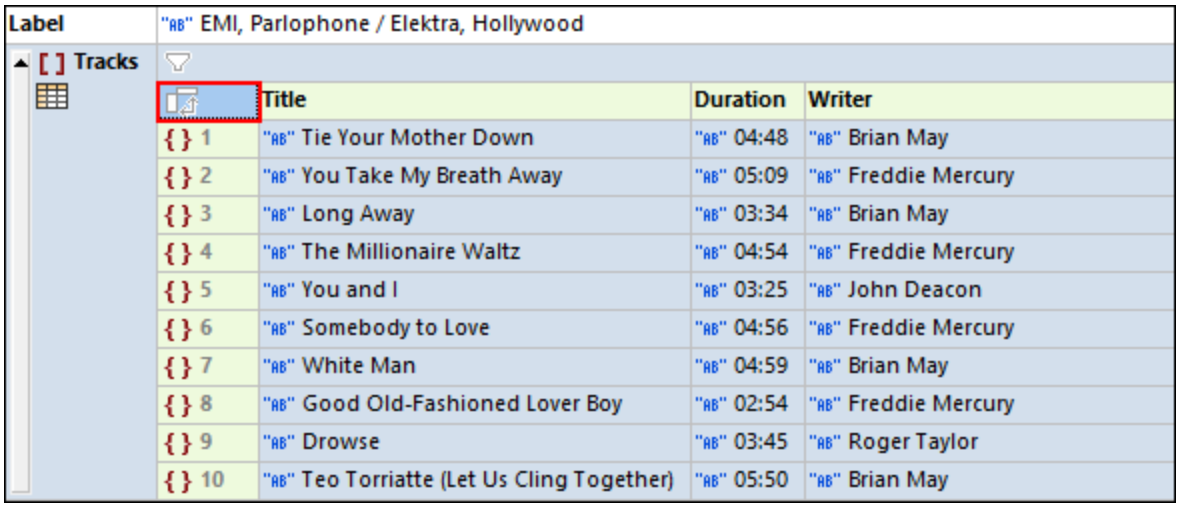
Consulte el apartado Vista Diseño de tabla[692] para más información.

28.5.6 Orden ascendente/descendente

Estos dos comandos se habilitan cuando se selecciona una columna de tabla de un componente. Para seleccionar una columna debe seleccionar su encabezado. El orden se basa en los valores de la columna.

28.5.7 Intercambiar filas y columnas

El comando **Intercambiar filas y columnas** se habilita en la Vista Diseño de tabla[692] si se selecciona la celda superior izquierda de una tabla (*en rojo en la imagen siguiente*). Este comando cambia las filas por las columnas y viceversa.



Label	"AB" EMI, Parlophone / Elektra, Hollywood		
[ ] Tracks			
	Title	Duration	Writer
{ } 1	"AB" Tie Your Mother Down	"AB" 04:48	"AB" Brian May
{ } 2	"AB" You Take My Breath Away	"AB" 05:09	"AB" Freddie Mercury
{ } 3	"AB" Long Away	"AB" 03:34	"AB" Brian May
{ } 4	"AB" The Millionaire Waltz	"AB" 04:54	"AB" Freddie Mercury
{ } 5	"AB" You and I	"AB" 03:25	"AB" John Deacon
{ } 6	"AB" Somebody to Love	"AB" 04:56	"AB" Freddie Mercury
{ } 7	"AB" White Man	"AB" 04:59	"AB" Brian May
{ } 8	"AB" Good Old-Fashioned Lover Boy	"AB" 02:54	"AB" Freddie Mercury
{ } 9	"AB" Drowse	"AB" 03:45	"AB" Roger Taylor
{ } 10	"AB" Teo Torriatte (Let Us Cling Together)	"AB" 05:50	"AB" Brian May

Consulte el apartado Vista Diseño de tabla[692] para más información.

28.5.8 Eliminar comentarios, volver a evaluar todo

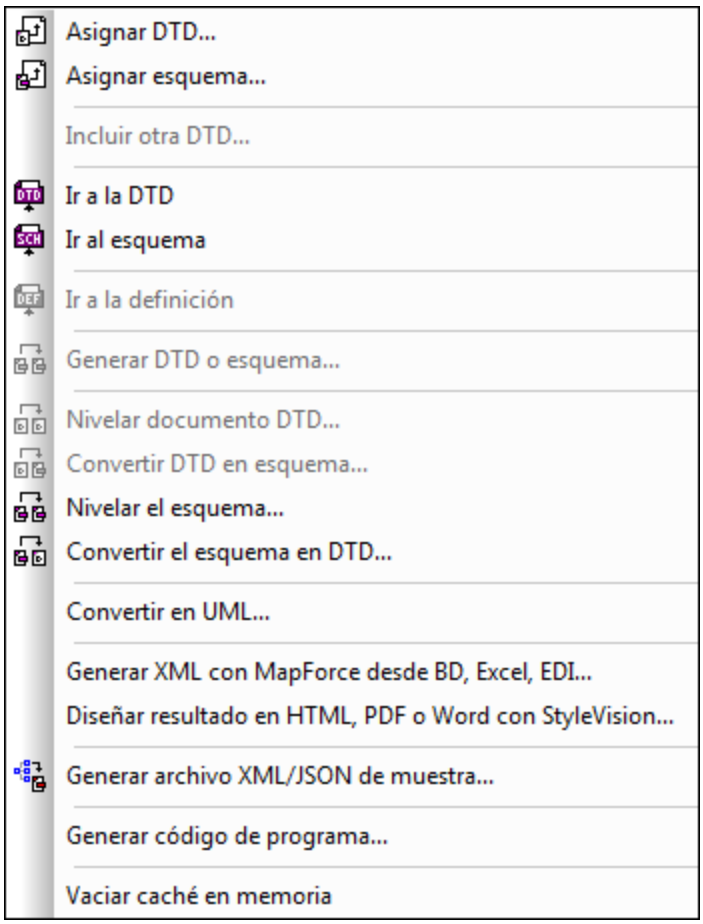
El comando **Eliminar comentarios** borra todos los comentarios. El comando se puede usar en la Vista Cuadrícula JSON.

El comando **Volver a evaluar todo** vuelve a evaluar todos los filtros[696] y las fórmulas[697]. Esto puede ser útil cuando el documento JSON accede a datos dinámicos que pueden cambiar con el tiempo (por ejemplo, tipos de cambio). El comando se puede usar en la Vista Texto y en la Vista Cuadrícula JSON.

© 2016-2022 Altova GmbH

28.6 Menú DTD/Esquema

El menú **DTD/Esquema** incluye comandos para trabajar de forma eficaz con documentos DTD y esquemas XML.



28.6.1 Asignar DTD



El comando **DTD/Esquema | Asignar DTD...** se habilita cuando el documento activo es un archivo XML. Este comando asigna una DTD al documento XML, permitiendo validarlo y habilitando las funciones de edición inteligente. Al hacer clic en este comando aparece un cuadro de diálogo donde puede seleccionar el archivo DTD que desea asignar. También puede seleccionar el archivo DTD por medio de un recurso global o desde una URL (haciendo clic en el botón **Examinar...** 1262) o puede seleccionar un archivo de las ventanas que están abiertas en XMLSpy (haciendo clic en el botón **Ventana...**). Recuerde que puede convertir la ruta de acceso del archivo DTD asignado en una ruta de acceso relativa marcando la casilla *Convertir ruta de acceso en*

relativa a... Cuando termine, el documento XML contendrá una declaración DOCTYPE que hace referencia a la DTD asignada. La declaración DOCTYPE será parecida a esta:

```
<!DOCTYPE main SYSTEM "http://link.xmlspy.com/spyweb.dtd">
```

Nota: puede asignar una DTD a un archivo XML nuevo nada más crearlo.

28.6.2 Asignar esquema



El comando **DTD/Esquema | Asignar esquema...** se habilita cuando el documento activo es un archivo XML. Este comando asigna un esquema XML al documento XML, permitiendo validarlo y habilitando las funciones de edición inteligente. Al hacer clic en este comando aparece un cuadro de diálogo donde puede seleccionar el esquema XML que desea asignar. También puede seleccionar el esquema por medio de un recurso global o desde una URL (haciendo clic en el botón **Examinar...** 1262) o puede seleccionar un archivo de las ventanas que están abiertas en XMLSpy (haciendo clic en el botón **Ventana...**). Recuerde que puede convertir la ruta de acceso del esquema asignado en una ruta de acceso relativa marcando la casilla *Convertir ruta de acceso en relativa a...* Cuando termine, el documento XML contendrá una asignación de esquema XML con los espacios de nombres necesarios. La asignación de esquema XML será parecida a esta:

```
xmlns="http://www.xmlspy.com/schemas/icon/orgchart"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xsi:schemaLocation="http://www.xmlspy.com/schemas/icon/orgchart
   http://schema.xmlspy.com/schemas/icon/orgchart.xsd"
```

28.6.3 Incluir otra DTD

El comando **DTD/Esquema | Incluir otra DTD...** permite incluir otra definición de tipo de documento (DTD) u otra entidad externa analizada dentro del subconjunto interno de una definición de tipo de documento o en cualquier documento DTD. Esto se hace definiendo la declaración de entidad externa analizada correspondiente y usando dicha entidad en esta línea:

```
<!ENTITY % navigation.dtd SYSTEM "S:\xml\navigation.dtd">
%navigation.dtd;
```

Al hacer clic en este comando aparece un cuadro de diálogo donde puede seleccionar el archivo DTD que desea incluir en su DTD.

Nota: este comando solamente está disponible en la vista Cuadrícula.

© 2016-2022 Altova GmbH

28.6.4 Ir a la DTD



El comando **DTD/Esquema | Ir a la DTD** abre la DTD en la que está basado el documento XML activo. Si el documento no tiene asignada ninguna DTD, entonces aparece un mensaje de error.

28.6.5 Ir al esquema



El comando **DTD/Esquema | Ir al esquema** abre el esquema XML en el que está basado el documento XML activo. Si el documento no tiene asignado ningún esquema XML, entonces aparece un mensaje de error.

28.6.6 Ir a la definición



El comando **DTD/Esquema | Ir a la definición** muestra la definición exacta de un elemento o atributo en el correspondiente documento DTD o esquema XML.

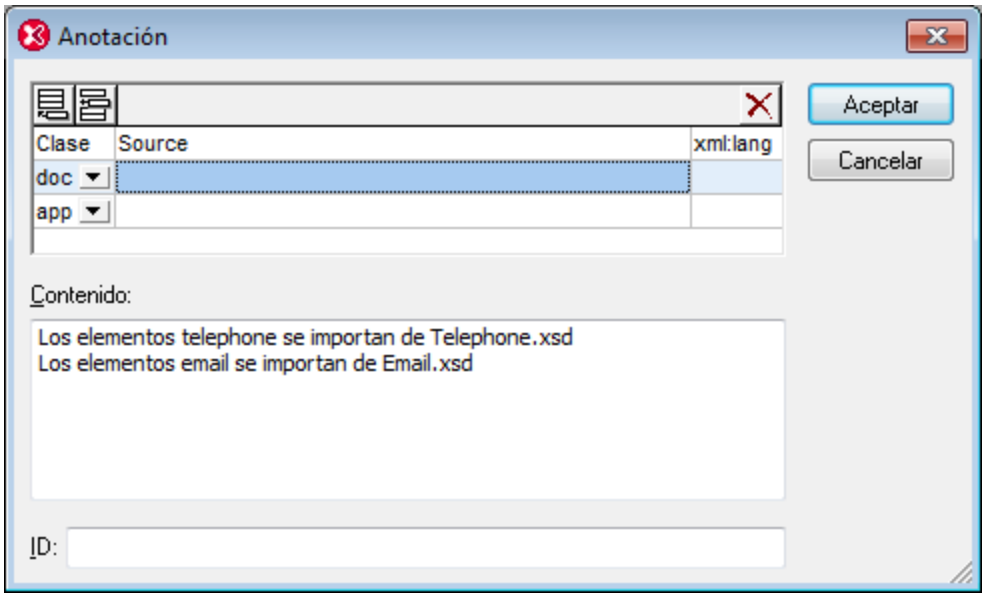
Para ver la definición del componente en la vista Cuadrícula:

1. Haga clic en el componente para seleccionarlo.
2. Seleccione la opción de menú **DTD/Esquema | Ir a la definición** o haga clic en el icono de la barra de herramientas.

Para ver la definición del componente en la vista Esquema:

- Con la tecla **Ctrl** pulsada, haga doble clic en el componente cuya definición desea ver.
- Otra opción es hacer clic en el componente y seleccionar la opción de menú **DTD/Esquema | Ir a la definición** (o el icono de la barra de herramientas).

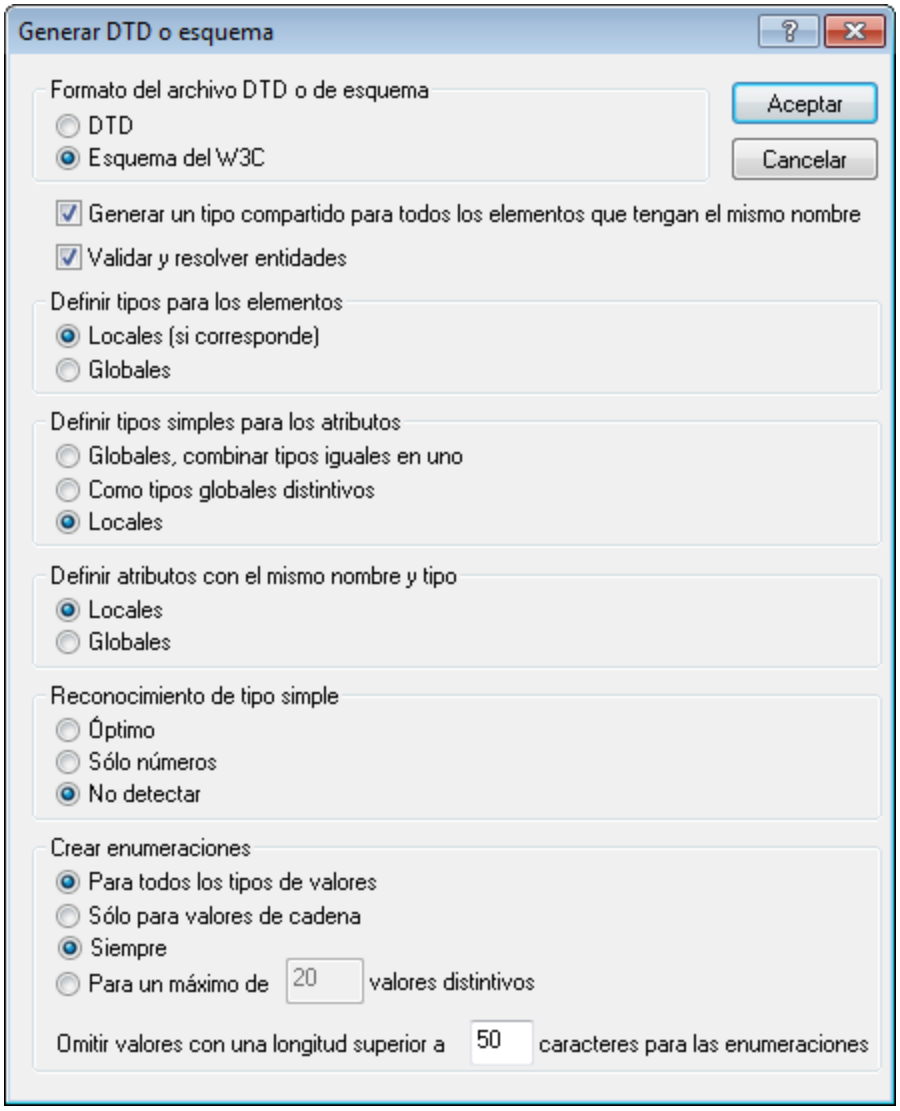
En ambos casos se abre el correspondiente documento DTD o esquema XML y se resalta la definición del componente.



28.6.7 Generar DTD o esquema

El comando **DTD/Esquema | Generar DTD o esquema** genera una DTD o un esquema XML nuevos a partir de un documento XML (o a partir de varios documentos XML de una carpeta de proyecto). Este comando es muy práctico si desea generar una DTD o un esquema XML a partir de sus documentos XML.

© 2016-2022 Altova GmbH



Si genera un esquema XML, están disponibles estas opciones:

- ***Elementos***: puede definir el tipo de los elementos local o globalmente (*Definir tipos para los elementos*). Si hay varios elementos con el mismo nombre, se puede declarar un tipo común para usarlo en la definición de estos elementos (*Generar un tipo compartido...*).
- ***Atributos***: los tipos simples de atributos (*Definir tipos simples para los atributos*) se pueden definir como (i) tipos globales comunes, (ii) tipos globales distintivos o (iii) tipos locales. Si hay varios atributos con el mismo nombre y el mismo tipo, estos atributos se pueden declarar local o globalmente.
- ***Reconocimiento de tipo simple:*** el reconocimiento automático de tipos (*Reconocimiento de tipo simple*) se puede configurar para que (i) reconozca todos los tipos posibles, (ii) reconozca sólo los tipos de datos numéricos o para que (iii) no reconozca ningún tipo de datos, en cuyo caso todos los tipos de datos se definen como `xs:string`.
- ***Resolución de entidades:*** en el documento XML las entidades pueden aparecer en el contenido de los elementos o en el valor de los atributos. Por tanto, esto repercute en los valores de enumeración, tanto si las entidades se resuelven como si no (*Validar y resolver entidades*). Además, algunas

entidades (sobre todo las entidades analizadas que contienen marcado) pueden afectar al modelo de contenido diferentemente, dependiendo de si se resuelven o no. Recuerde que antes de generar el esquema se comprueba si el formato XML del documento es correcto. Si el documento no tiene un formato XML correcto, se detiene el proceso de generación de esquemas.

- ***Enumeraciones:*** se pueden enumerar todos los tipos de valores o solamente los valores de cadena.

Si genera una DTD, solamente están disponibles las opciones relacionadas con la resolución de entidades y con las enumeraciones.

El comando **Generar DTD o esquema** suele ejecutarse en el documento que está activo en la ventana principal, pero también puede usar este comando con un archivo, una carpeta o un grupo de archivos de la ventana Proyecto.

Si están presentes elementos o atributos en más de un espacio de nombres, XMLSpy genera un esquema XML por cada espacio de nombres. Por tanto, es posible que se creen varios archivos en el disco.

28.6.8 Nivelar documento DTD

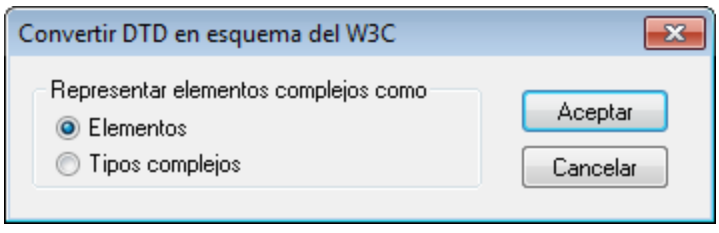
El comando **Nivelar documento DTD** se habilita cuando el documento activo es una DTD. Este comando crea un nuevo documento DTD plano, quitando las entidades de parámetro y generando una sola DTD a partir de un grupo de módulos. También suprime las secciones marcadas con `IGNORE` y elimina las entidades de parámetro no analizadas.

Al hacer clic en este comando se abre el cuadro de diálogo "Guardar", donde puede seleccionar la ubicación donde se debe guardar el archivo DTD generado. Haga clic en **Guardar** para llevar a cabo la conversión. El archivo DTD nivelado se genera y se abre en XMLSpy.

28.6.9 Convertir DTD en esquema

El comando **DTD/Esquema | Convertir DTD en esquema...** se habilita cuando el documento activo es una DTD y convierte el archivo DTD en un esquema XML (XSD).

Al hacer clic en este comando se abre el cuadro de diálogo "Convertir DTD en esquema del W3C" (*imagen siguiente*). En este diálogo puede seleccionar si los elementos complejos se convierten en elementos o en tipos complejos. Al hacer clic en **Aceptar** aparece otro diálogo donde debe elegir la ubicación donde se guarda el archivo XSD generado. Haga clic en **Guardar** para realizar la conversión. El archivo XSD se genera y se abre en XMLSpy.



Cuando convierta una DTD en un esquema XML, XMLSpy hace varias suposiciones, ya que la información disponible es limitada. En particular, los valores de ciertos componentes DTD se tratan de forma literal, en

 © 2016-2022 Altova GmbH

lugar de analizarse su semántica. Esto se debe a que el programa no puede saber cuál de todos los usos posibles es el uso deseado. En estos casos, recomendamos que realice los cambios necesarios en el esquema XML generado.

Sea como fuere, recomendamos que examine detenidamente el esquema XML o documento DTD generado por si fuera necesario corregirlo. A continuación mencionamos algunos aspectos del documento generado que quizás puedan mejorarse.

Tipos de datos de los atributos
Las DTD admiten solamente 10 tipos de datos de atributos, mientras que los esquemas XML admiten más de 40 tipos de datos además de los tipos de datos derivados. Quizás sea buena idea mejorar el esquema XML generado usando un tipo de datos más restrictivo, por ejemplo. Recuerde que, cuando un esquema XML se convierte en una DTD[1363], se pierde la información de tipos de datos.

Espacios de nombres
Las DTD no distinguen los espacios de nombres. Por tanto, si desea especificar espacios de nombre en una DTD debe codificarlos de forma rígida en el nombre de los elementos o atributos. Esto puede presentar problemas a la hora de convertir un esquema en otro esquema.

Entidades
La especificación XML Schema no tiene equivalentes para las declaraciones de entidad generales de las DTD. Cuando XMLSpy convierte una DTD en un esquema XML, omite las declaraciones de entidad.

Declaraciones de datos sin analizar
Las DTD y los esquemas XML usan mecanismos distintos a la hora de tratar datos sin analizar.

Las DTD usan este mecanismo:

- Se declara una notación compuesta por un nombre y un identificador, como por ejemplo:
  `<!NOTATION gif SYSTEM "image/gif">`
- Usted declara la entidad, por ejemplo:
  `<!ENTITY cover_img SYSTEM "graphics/cover_img.gif" NDATA gif>`
- Y después, por lo general, especifica un tipo de atributo de `ENTITY` en el atributo pertinente, por ejemplo:
  ```
  <!ELEMENT img EMPTY>
  <!ATTLIST img format ENTITY #REQUIRED>
  ```

Por el contrario, los esquemas XML usan este mecanismo:

- Declare una notación. La notación funciona igual que en la DTD.
  `<xs:notation name="gif" public="image/gif"/>`
  Recuerde que es obligatorio usar el atributo `public` que almacena el identificador. Puede usar el atributo opcional `system`, que almacena el identificador de sistema y suele ser un ejecutable que se encarga de los recursos del tipo de la notación.
- Puede asociar la declaración de notación con un valor de atributo dado usando el tipo de datos `NOTATION`.

  No obstante, no puede usar el tipo de datos `NOTATION` directamente, sino que debe derivar otro tipo de datos del tipo de datos `NOTATION`.

```
<xs:simpleType name="formatType">
  <xs:restriction base="xs:NOTATION">
    <xs:enumeration value="gif"/>
    <xs:enumeration value="jpeg"/>
  </xs:restriction>
</xs:simpleType>
```

Asocie el atributo con el tipo de datos derivado del tipo de datos `NOTATION`, por ejemplo:

```
<xs:complexType name="imgType">
  <xs:attribute name="height"/>
  <xs:attribute name="width"/>
  <xs:attribute name="location"/>
  <xs:attribute name="format" type="formatType" use="required"/>
</xs:complexType>
<xs:element name="img" type="imgType"/>
```

Cuando convierta una DTD en un esquema XML esto es lo que hace XMLSpy:

- Algo parecido a esto
  ```
  <!ATTLIST image format ENTITY #REQUIRED
  ...>
  ```
  se convierte en
  ```
  <xs:attribute name="format" type="xs:ENTITY" use="required"/>
  ```
- Y algo como esto
  ```
  <!NOTATION gif SYSTEM "image/gif">
  ```
  se convierte en
  ```
  <xs:notation name="gif" system="image/gif"/>
  ```

Por tanto, recomendamos que haga estos cambios:

1. En las notaciones de tipo `<xs:notation name="gif" system="image/gif"/>` sustituya `system` con `public` y añada un identificador de sistema opcional si es necesario.
2. Derive un tipo de datos del tipo de datos `NOTATION` tal y como explicamos más arriba para `formatType`.
3. Asocie el tipo de datos derivado al atributo correspondiente.

Nota importante: según la especificación XML Schema, no es necesario (o no es posible, según como se mire) declarar una entidad externa.

28.6.10 Nivelar el esquema

El comando **Nivelar el esquema** se habilita cuando el documento activo es un esquema XML. Este comando genera un archivo XSD plano nuevo, (i) añadiendo los componentes de los esquemas incluidos como componentes globales del esquema activo y (ii) eliminando los esquemas incluidos.

© 2016-2022 Altova GmbH

Este comando redirecciona al comando **Nivelar el esquema** 1390 del menú **Diseño de esquemas**. Como el comando **Nivelar el esquema** 1390 solo está disponible en la vista **Esquema**, la aplicación pregunta si desea cambiar a la vista **Esquema** o no.

28.6.11 Convertir el esquema en DTD

El comando **Convertir el esquema en DTD** se habilita cuando el documento activo es un esquema XML y convierte documentos de esquema XML (XSD) en documentos DTD.

Al hacer clic en este comando aparece el cuadro de diálogo "Guardar", donde puede seleccionar la ubicación donde se debe guardar el archivo DTD generado. Haga clic en **Guardar** para realizar la conversión. El archivo DTD se genera y se abre en XMLSpy.

Es importante recordar que:

1. Cuando convierta un esquema XML en una DTD, los prefijos de espacio de nombres utilizados en el esquema XML (no los URI de espacios de nombres ni las declaraciones de espacio de nombres) se transpasan a los nombres de los elementos y atributos correspondientes de la DTD.
2. Los analizadores XML omiten los espacios de nombres cuando se valida un documento XML con una DTD. Por este motivo las declaraciones de espacio de nombres no se convierten.
3. Los atributos `elementFormDefault` y `attributeFormDefault` del elemento `xs:schema` definen los elementos y atributos cuyos prefijos se incluyen en el proceso de conversión. Si el valor de estos atributos es `unqualified`, entonces se incluyen los prefijos de los elementos y atributos declarados globalmente. Si el valor de estos atributos es `qualified`, entonces se incluyen los prefijos de todos los nombres de elemento y atributo.
4. Los prefijos se convierten en su valor de cadena equivalente más un punto y coma. Los elementos y atributos de los espacios de nombres predeterminados se convierten en elementos y atributos cuyos nombres empiezan con la cadena: `default_NS_X`, siendo `X` un entero (empezando por 1 y con un valor máximo igual al número de espacios de nombres predeterminados usados en el esquema XML).
5. En la DTD los nombres de los elementos se componen de entidades de parámetro. Esto permite cambiar fácilmente el prefijo en la DTD si no es necesario cambiar el prefijo en el documento XML. Las definiciones de entidades de parámetro se pueden cambiar en el documento DTD directamente o reemplazando las definiciones de entidades de parámetro en el subconjunto DTD interno del documento XML.

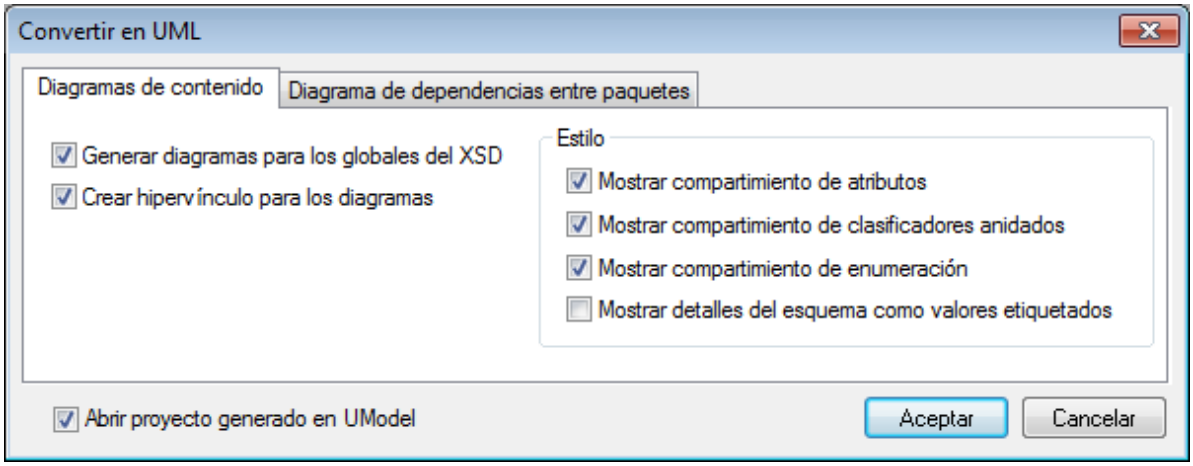
Nota importante: los espacios de nombres no tienen valor semántico en las DTD y los prefijos de espacio de nombres procedentes del esquema XML no son más que una parte léxica del nombre del elemento o atributo definido en la DTD.

28.6.12 Convertir en UML

El comando **DTD/Esquema | Convertir en UML** convierte un esquema XML del W3C en un proyecto de Altova UModel (archivo `.ump`). UMP es el formato nativo de Altova UModel, la aplicación de modelado de software de Altova. Los archivos UMP se pueden ver y editar en Altova UModel.

Para convertir un esquema en UML, siga estos pasos:

1. Con el esquema abierto, haga clic en el comando **Convertir en UML**. Aparece el cuadro de diálogo "Convertir en UML" (*imagen siguiente*).



2. En la pestaña *Diagramas de contenido*, seleccione la opción *Generar diagramas para los globales del XSD*. Esto hace que en el proyecto de UModel se genere un diagrama de modelo de contenido por cada componente global.
3. Seleccione las opciones pertinentes en el cuadro de diálogo (descritas más abajo en este apartado).
4. Si desea ver el proyecto en UModel inmediatamente, marque la casilla *Abrir proyecto generado en UModel.*
5. Haga clic en **Aceptar**.
6. Aparece el cuadro de diálogo "Guardar como". Busque la carpeta de destino, escriba el nombre del archivo UMP y haga clic en **Guardar**.

Opciones de conversión

Estas son las opciones que puede configurar en el cuadro de diálogo "Convertir en UML":

Pestaña ***Diagramas de contenido***:

- *Crear hipervínculo para los diagramas:* en cada diagrama se crea un hipervínculo a la entrada de dicho componente global en la vista Estructura del modelo. Así puede localizar rápidamente el componente en la jerarquía del diagrama.
- En el panel *Estilo*, puede seleccionar qué compartimientos se muestran/ocultan.

Pestaña ***Diagrama de dependencias entre paquetes***:

- *Generar diagrama*: define si se genera o no un diagrama de dependencias entre paquetes. Un diagrama de dependencias entre paquetes ofrece un resumen de todo el paquete, mostrando las relaciones existentes entre los componentes del paquete. El resto de opciones de esta pestaña solamente se habilitan si marca esta casilla.
- *Enlazar paquete al diagrama*: en el diagrama de paquetes se crea un enlace a la vista Estructura del modelo.
- Hay cuatro opciones de diseño: (i) diseño desorganizado (si no se marca la casilla *Diseño automático*); (ii) Diseño jerárquico (si marca la casilla *Diseño automático* y la opción *jerárquico* del menú desplegable); (iii) diseño por bloques (si marca la casilla *Diseño automático* y la opción *por bloques* del menú) y (iv) con espaciado uniforme (si marca la casilla *Diseño automático* y la opción *dirigido por fuerzas*). El diseño se puede modificar más adelante en UModel.

© 2016-2022 Altova GmbH

Nota: esta función de conversión en UML solamente funciona con los esquemas XML del W3C.

28.6.13 Generar XML con MapForce desde BD, Excel, EDI

El comando **DTD/Esquema | Generar XML con MapForce desde BD, Excel, EDI...** inicia la aplicación Altova MapForce si está instalada. La herramienta MapForce permite realizar asignaciones de datos entre un esquema XML y otro documento DTD, esquema XML o base de datos y después generar un archivo XML.

28.6.14 Diseñar resultado en HTML, PDF o Word con StyleVision

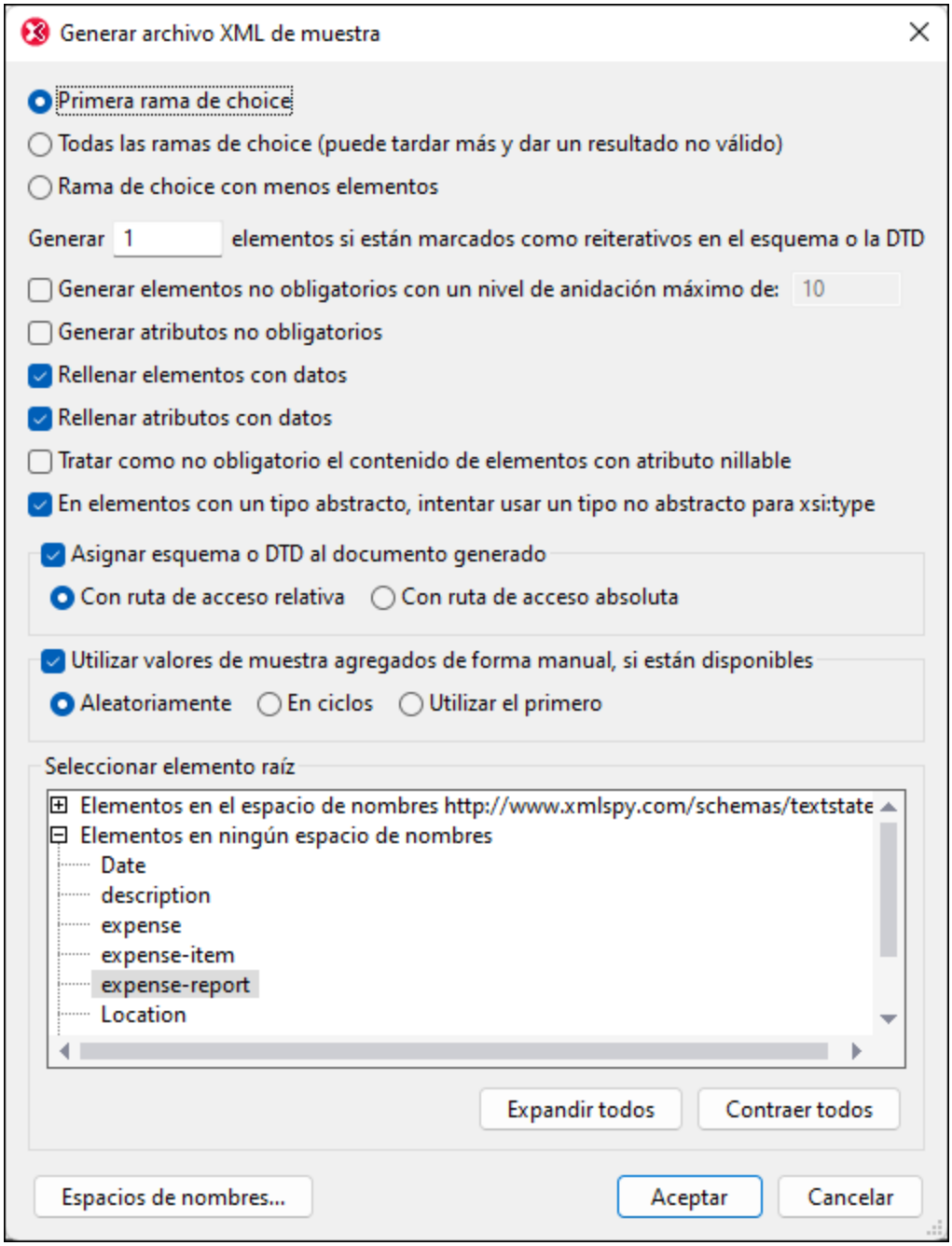
El comando **DTD/Esquema | Diseñar resultado en HTML, PDF o Word con StyleVision...** inicia la aplicación Altova StyleVision si está instalada. La herramienta StyleVision permite diseñar hojas de estilos para documentos de salida HTML, PDF y RTF.

28.6.15 Generar archivo XML/JSON de muestra

El comando **Generar archivo XML/JSON de muestra** se habilita en la vista Texto, Cuadrícula y Esquema y genera una instancia XML o JSON basada en el archivo de esquema que está activo. Si el archivo activo es una DTD o un esquema XML, entonces se genera una instancia XML. Si, por el contrario, el archivo activo es un esquema JSON, entonces se genera un archivo de instancia JSON.

Generar archivos XML de muestra

Puede generar archivos XML de muestra si el archivo activo es una DTD o un esquema XML. Al hacer clic en el comando **Generar archivo XML/JSON de muestra** aparece el cuadro de diálogo "Generar archivo XML de muestra" (*imagen siguiente*), donde puede configurar la generación del documento de muestra.



Elementos de grupos choice

Un grupo `choice` es un grupo de elementos de los cuales se puede usar cualquiera. Por ejemplo, si se define que el elemento llamado `artículos` tiene un grupo `choice` formado por tres elementos: `cd`, `dvd`, `book`, el elemento `artículos` puede tener cualquiera de estos tres elementos como secundario (que puede aparecer el número máximo de veces indicado en el atributo `maxOccurs` de dicho elemento).

En el cuadro de diálogo "Generar archivo XML de muestra" puede seleccionar si en el archivo XML de muestra se generan (i) el elemento de la primera rama del grupo `choice`, (ii) los elementos de todas las ramas o (iii) los

 © 2016-2022 Altova GmbH

elementos de la rama que tenga menos descendientes. Recuerde que si la opción *Todas las ramas* puede dar lugar a un documento no válido porque solo se permite una rama del grupo `choice`.

Si una de las ramas del grupo `choice` se puede repetir (es decir, si el atributo `maxOccurs` tiene un valor mayor que `1`), entonces puede especificar en el primer cuadro de texto del diálogo cuántos elementos repetibles se generan en el XML de muestra (un máximo de 99). Si el atributo `maxOccurs` del grupo `choice` tiene el valor `unbounded` (ilimitado) o un valor muy alto y marca la casilla Todas las ramas, entonces el atributo `maxOccurs` del grupo `choice` también está limitado por el número de elementos repetibles que especificó en el primer cuadro de texto del diálogo

Generar atributos no obligatorios
Si marca esta opción, se generan tanto los atributos obligatorios como los atributos no obligatorios definidos en el esquema. Al activar esta opción también se puede indicar el nivel de anidación que quiere. Puede aumentar la profundidad de anidación para que se generen elementos no obligatorios hasta el nivel que indique.

Generar X elementos si están marcados como reiterativos en el esquema o la DTD
Si rellena este cuadro de texto, se genera el número de elementos reiterativos indicados en el cuadro de texto. Esto afecta a todos los elementos, incluidos los elementos de los grupos `choice`.

Rellenar elementos y atributos con datos
Si marca esta opción, se insertan los valores de tipo de datos de los respectivos elementos y atributos. Por ejemplo, si un elemento tiene definido un tipo de datos `string`, el elemento recibe el valor de muestra `string`.

Elementos con el atributo nillable y elementos con un tipo abstracto
El contenido de los elementos con el atributo nillable se puede considerar no obligatorio. Los elementos con un tipo abstracto pueden usar un tipo no abstracto para su atributo `xsi:type`.

Asignar esquema al archivo XML generado
El esquema utilizado para generar el archivo XML se puede asignar al archivo XML generado con una ruta de acceso relativa o absoluta.

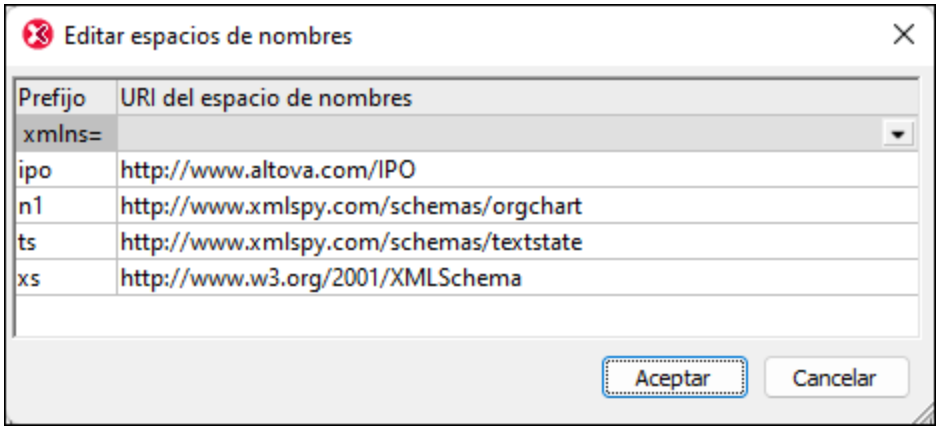
Utilizar valores de muestra agregados de forma manual, si están disponibles

Si el componente del esquema tiene asignados valores de muestra, entonces se usarán como valor o contenido del componente. A los componentes se les asignan valores de muestra desde el ayudante de entrada Facetas 234, en su pestaña *Muestras*. Puede especificar qué valores de muestra se usan para generar el archivo:

- Una selección aleatoria de valores de muestra.
- Cada valor de muestra de cada instancia del componente. Cada vez que se genere un archivo se empieza otra vez.
- Siempre el primer valor.

Elemento raíz
Si el esquema contiene más de uno, los elementos globales se enumeran en este panel y puede seleccionar el elemento raíz para el archivo XML de muestra.

Espacios de nombres
Haga clic en el botón **Espacios de nombres** para abrir el cuadro de diálogo "Editar espacios de nombres" (*imagen siguiente*). Los espacios de nombres que se definieron en el esquema y los espacios de nombres del estándar XML Schema aparecen en este cuadro de diálogo.

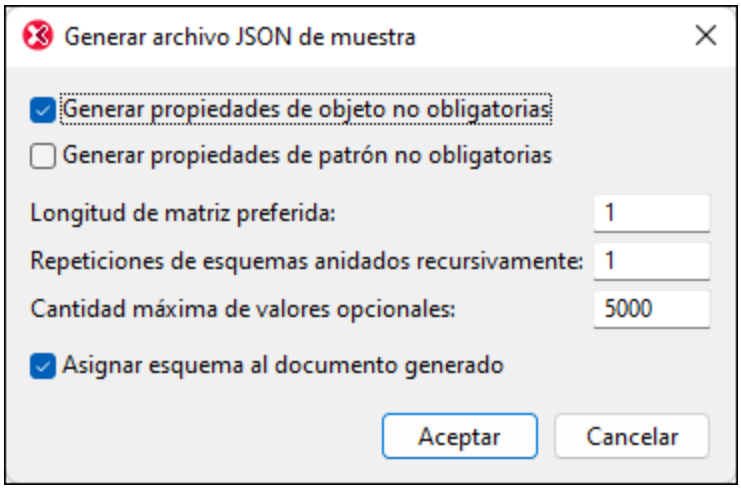


Aquí puede editar:

- El prefijo de espacio de nombres vinculado a cualquier espacio de nombres del documento. Los prefijos de espacio de nombres que se configuren en este cuadro de diálogo se usarán (en el archivo XML de muestra) para crear los prefijos de los nodos que estén en el espacio de nombres correspondiente. Por ejemplo, en la imagen anterior puede ver que los nodos que estén en el espacio de nombres `http://www.xmlspy.com/schemas/textstate` recibirán el prefijo **`ts:`** en el archivo de muestra que se generará.
- Puede configurar uno de los espacios de nombres del documento como espacio de nombres predeterminado (`xmlns=`). Para ello basta con seleccionar el espacio de nombres correspondiente en el cuadro combinado *xmlns=*. Los nodos que estén en el espacio de nombres seleccionado como espacio de nombres predeterminado se generarán sin prefijo de espacio de nombres.

Generar archivos JSON de muestra

Puede generar archivos JSON de muestra si el documento activo es un esquema JSON. Al hacer clic en el comando **Generar archivo XML/JSON de muestra** aparece el cuadro de diálogo "Generar archivo JSON de muestra" (*imagen siguiente*), donde puede configurar la generación del documento de muestra.



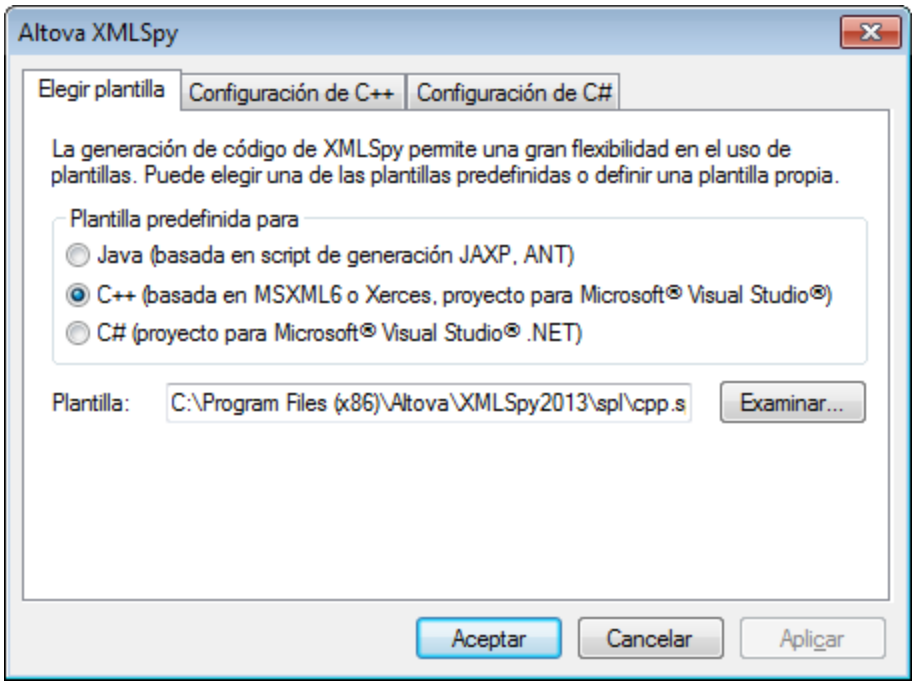
Puede elegir si se generan o no las propiedades de objetos no obligatorias[725], propiedades de patrón no obligatorias[725], puede especificar la longitud de las matrices[736] y establecer el número de repeticiones de

 © 2016-2022 Altova GmbH

definiciones recursivas. También puede elegir si el esquema JSON activo se asigna automáticamente[754] o no al archivo de muestra JSON generado.

28.6.16 Generar código de programa

El comando **DTD/Esquema | Generar código de programa** genera archivos de clase Java, C++ o C# a partir de las definiciones del documento de esquema activo (DTD o esquema XML).



Consulte la sección Generador de código[1138] del *Manual del usuario* para más información sobre la función de generación de código.

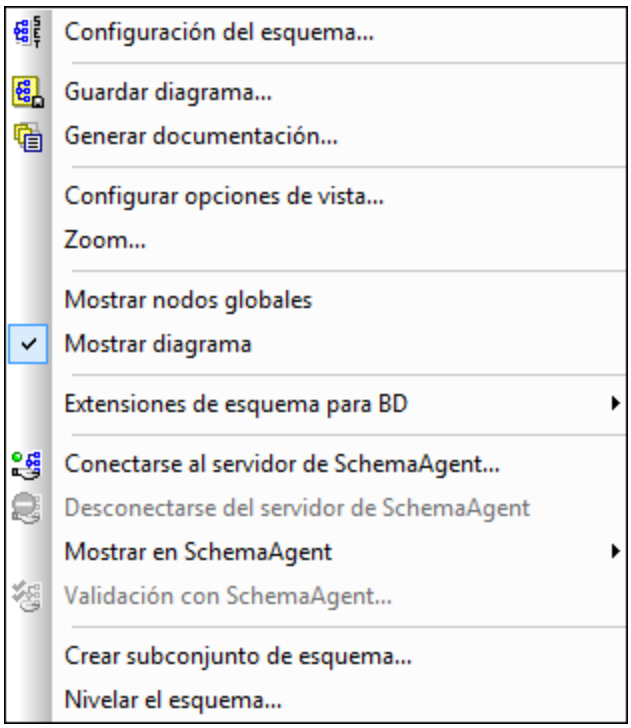
28.6.17 Vaciar caché en memoria

El comando **DTD/Esquema | Vaciar caché en memoria** elimina todos los documentos de esquema (DTD o esquema XML) de la memoria caché. Para agilizar el proceso de validación y edición inteligente, XMLSpy almacena en la memoria caché los documentos de esquema y las entidades externas analizadas que se usaron recientemente. La información sobre estos documentos en caché también aparece cuando se invoca el comando Ir a la definición[1357].

Vacíe el caché si no tiene suficiente memoria en el sistema o si recientemente utilizó documentos basados en esquemas distintos.

28.7 Menú Diseño de esquemas

El menú **Diseño de esquemas** sirve para configurar la vista Esquema de XMLSpy. Esta vista está disponible cuando el archivo activo es un documento de esquema XML y en ella puede diseñar esquemas XML.

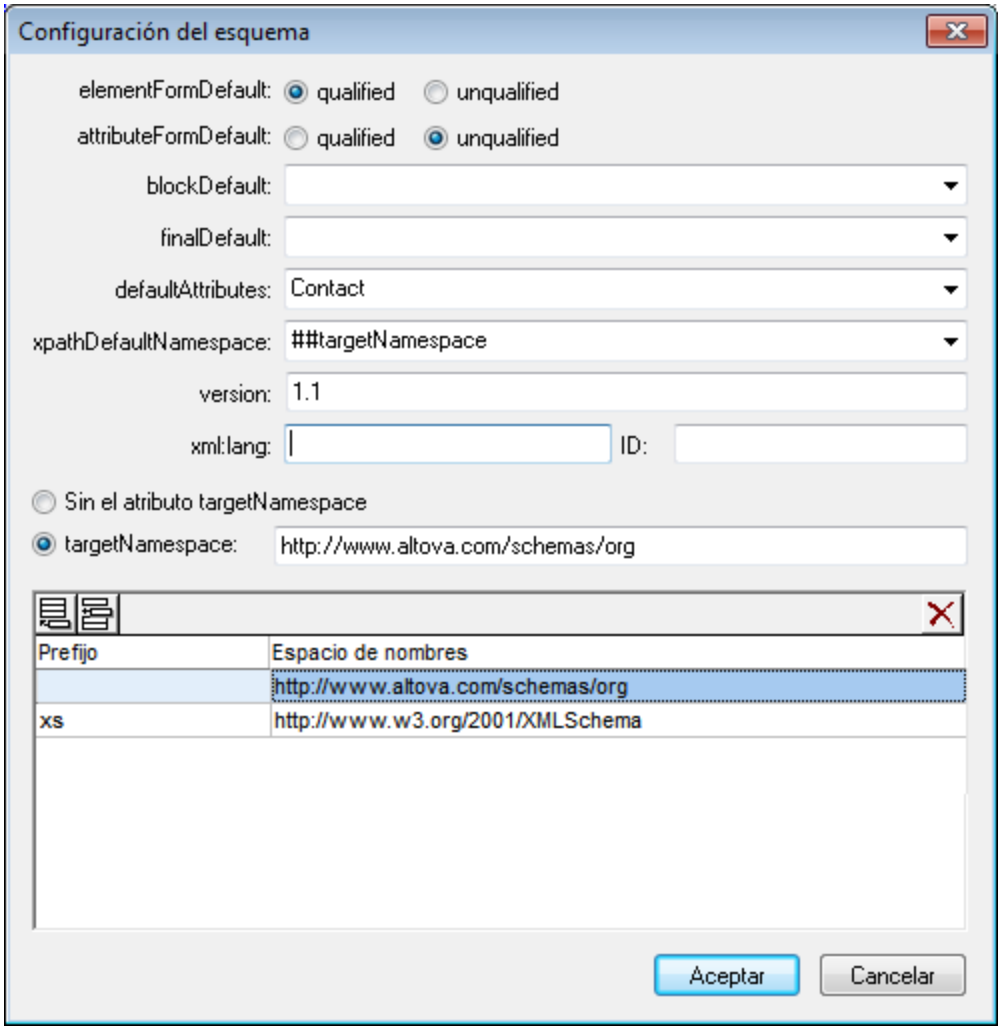


28.7.1 Configuración del esquema



El comando **Diseño de esquemas | Configuración del esquema** está disponible en la vista Esquema solamente y sirve para definir la configuración global del esquema activo. El esquema se configura definiendo los atributos del elemento `xs:schema`.

© 2016-2022 Altova GmbH



Las opciones de configuración definidas en el cuadro de diálogo anterior (con el modo XSD 1.1 activo) dan lugar a este elemento `xs:schema`:

```
<xs:schema xmlns="http://www.altova.com/schemas/org"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        targetNamespace="http://www.altova.com/schemas/org"
        elementFormDefault="qualified"
        attributeFormDefault="unqualified"
        xpathDefaultNamespace="##targetNamespace"
        version="1.1"
        defaultAttributes="Contact">
```

Nota: las opciones del cuadro de diálogo "Configuración del esquema" difieren ligeramente dependiendo del modo XSD activo. Si el modo activo es XSD 1.0, los atributos de XSD 1.1 no aparecen en el cuadro de diálogo.

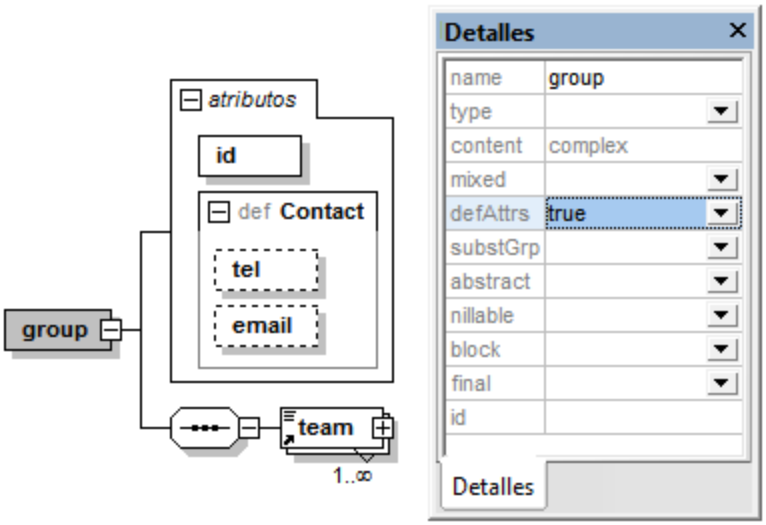
Los atributos `defaultAttributes` y `xpathDefaultNamespace` son características de la especificación XML Schema 1.1 y, por tanto, están disponibles en el modo XSD 1.1[173] solamente. Los demás atributos están disponibles tanto en el modo XSD 1.0 como en XSD 1.1.

El atributo `version`

Este atributo almacena la versión del documento y no se debe confundir con la versión XSD del documento[173].

El atributo `defaultAttributes`

El atributo **`defaultAttributes`** sirve para seleccionar un grupo de atributos como grupo de atributos predeterminado de todos los tipos complejos del esquema. El grupo de atributos predeterminado aparece en el modelo de contenido de estos tipos complejos. En la imagen siguiente, por ejemplo, el grupo de atributos `Contact` es el grupo de atributos predeterminado (*tal y como se configuró en el cuadro de diálogo de la imagen anterior*) y está disponible automáticamente en el elemento `group`. Para deshabilitar el grupo de atributos, cambie el valor de atributo `defaultAttributesApply` del tipo complejo por `false`. Esto se puede hacer cambiando la propiedad *defAttrs* en el ayudante de entrada Detalles del tipo complejo (*imagen siguiente*).



El atributo `xpathDefaultNamespace`

El atributo **`xpathDefaultNamespace`** sirve para establecer el espacio de nombres predeterminado para los elementos de las expresiones XPath utilizadas en el esquema. Si se configura en el cuadro de diálogo "Configuración del esquema", el atributo se aplica al elemento de nivel superior `xs:schema`. De este modo, el ámbito de la declaración es todo el documento. Puede invalidar la declaración en `xs:schema` con declaraciones en los elementos en los que se permite el atributo:

- `xs:assert` y `xs:assertion`
- `xs:alternative`
- `xs:selector` y `xs:field` (en restricciones de identidad)

También puede cambiar el espacio de nombres predeterminado de las expresiones XPath en el ayudante de entrada Detalles de los elementos de la lista anterior.

El atributo **`xpathDefaultNamespace`** puede tener uno de estos tres valores:

© 2016-2022 Altova GmbH

- `##targetNamespace`: el espacio de nombres predeterminado de las expresiones XPath será el mismo que el espacio de nombres de destino del esquema
- `##defaultNamespace`: el espacio de nombres predeterminado de las expresiones XPath será el mismo que el espacio de nombres predeterminado del esquema
- `##local`: no hay ningún espacio de nombres predeterminado para las expresiones XPath

Si no se declara un espacio de nombres predeterminado para las expresiones XPath en el documento, los elementos sin prefijo de las expresiones XPath no estarán en ningún espacio de nombres. La declaración de espacio de nombres para expresiones XPath no afecta a los atributos.

28.7.2 Guardar diagrama



El comando **Diseño de esquemas | Guardar diagrama** sirve para guardar en formato PNG y en la ubicación que quiera el diagrama del modelo de contenido que está activo en la ventana principal.

28.7.3 Generar documentación



El comando **Diseño de esquemas | Generar documentación** genera documentación detallada sobre el esquema XML o JSON (*imagen siguiente*) en formato HTML, Word, RTF y PDF. La documentación generada con este comando se puede cambiar o usar con libertad, no es necesario obtener permiso de Altova. La documentación generada incluye detalles sobre los componentes seleccionados en el cuadro de diálogo "Documentación del esquema" (que aparece al seleccionar el comando **Generar documentación**). Los elementos que están relacionados entre sí (como los elementos secundarios, los tipos complejos, etc.) suelen estar conectados por medio de hipervínculos en la documentación, para que pueda navegar de componente a componente. Los componentes que tienen un modelo de contenido también tienen hipervínculos a las definiciones del modelo de contenido. Recuerde que también se genera documentación de esquema para los **componentes de esquema incluidos e importados**. Las opciones de generación de documentación para esquemas XML se describen en el apartado *Opciones de documentación*[1374]. Las opciones de configuración de la documentación de esquemas JSON se describen en la sección *Generar documentación para esquemas JSON*[751].

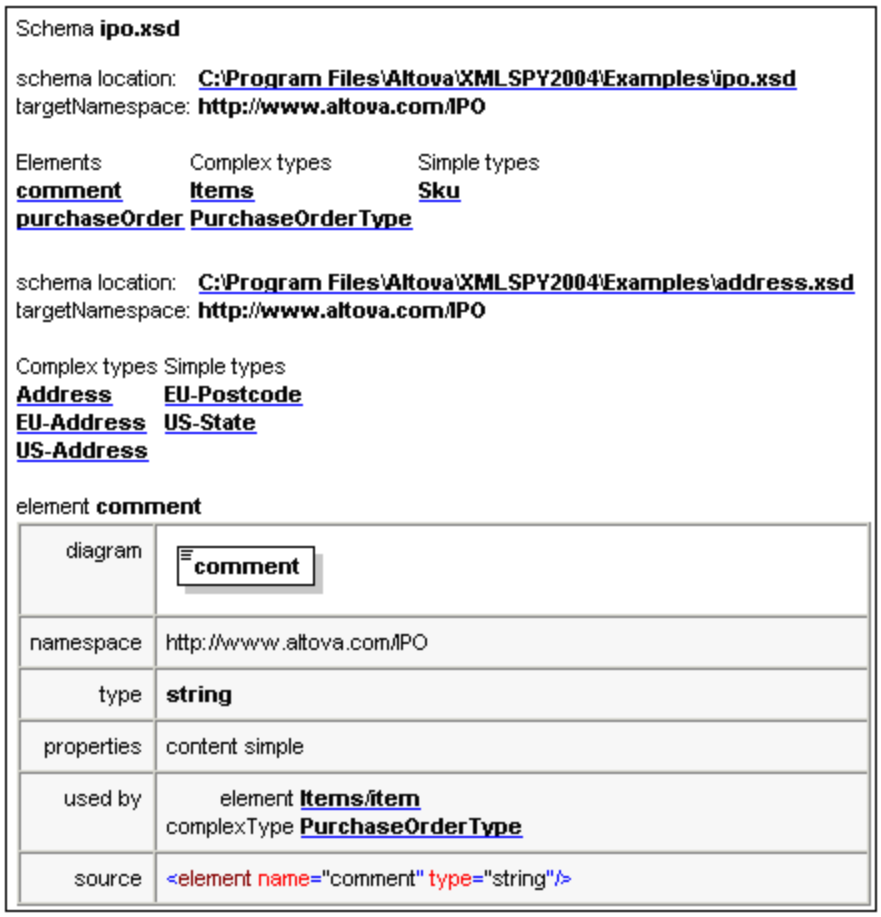
Debe tener en cuenta que las opciones de documentación[1374] tienen prioridad sobre las opciones de configuración elegidas en el cuadro de diálogo Configurar visualización del esquema[1378].

Nota: para generar la documentación en formato MS Word es necesario tener instalado Microsoft Word (versión 2000 o superior).

Para generar la documentación puede usar el diseño estándar fijo de XMLSpy o un archivo SPS de StyleVision. Si usa un archivo SPS de StyleVision puede personalizar el diseño de la documentación generada

y generarla también en formato PDF. Para más información, consulte el apartado Diseño definido por el usuario[1377].

Nota: para usar un archivo SPS para generar documentación de esquema es necesario tener instalado StyleVision.


Schema **ipo.xsd**

schema location: **C:\Program Files\Altova\XMLSPY2004\Examples\ipo.xsd**
targetNamespace: **http://www.altova.com/IPO**

Elements	Complex types	Simple types
comment	**Items**	**Sku**
purchaseOrder	**PurchaseOrderType**	

schema location: **C:\Program Files\Altova\XMLSPY2004\Examples\address.xsd**
targetNamespace: **http://www.altova.com/IPO**

Complex types	Simple types
Address	**EU-Postcode**
EU-Address	**US-State**
US-Address	

element **comment**

diagram	comment
namespace	http://www.altova.com/IPO
type	**string**
properties	content simple
used by	element **Items/item** complexType **PurchaseOrderType**
source	<element name="comment" type="string"/>

La imagen anterior muestra la documentación de un esquema generada con XMLSpy. Observe que al principio de la documentación aparece un índice (con todos los esquemas relacionados y con sus componentes globales organizados por tipo).

Nota: para generar documentación de documentos de esquema del W3C, XMLSpy usa versiones internas de estos documentos. Por consiguiente, la documentación no reflejará redefiniciones ni modificaciones de estos documentos de esquema.

28.7.3.1 Opciones de documentación

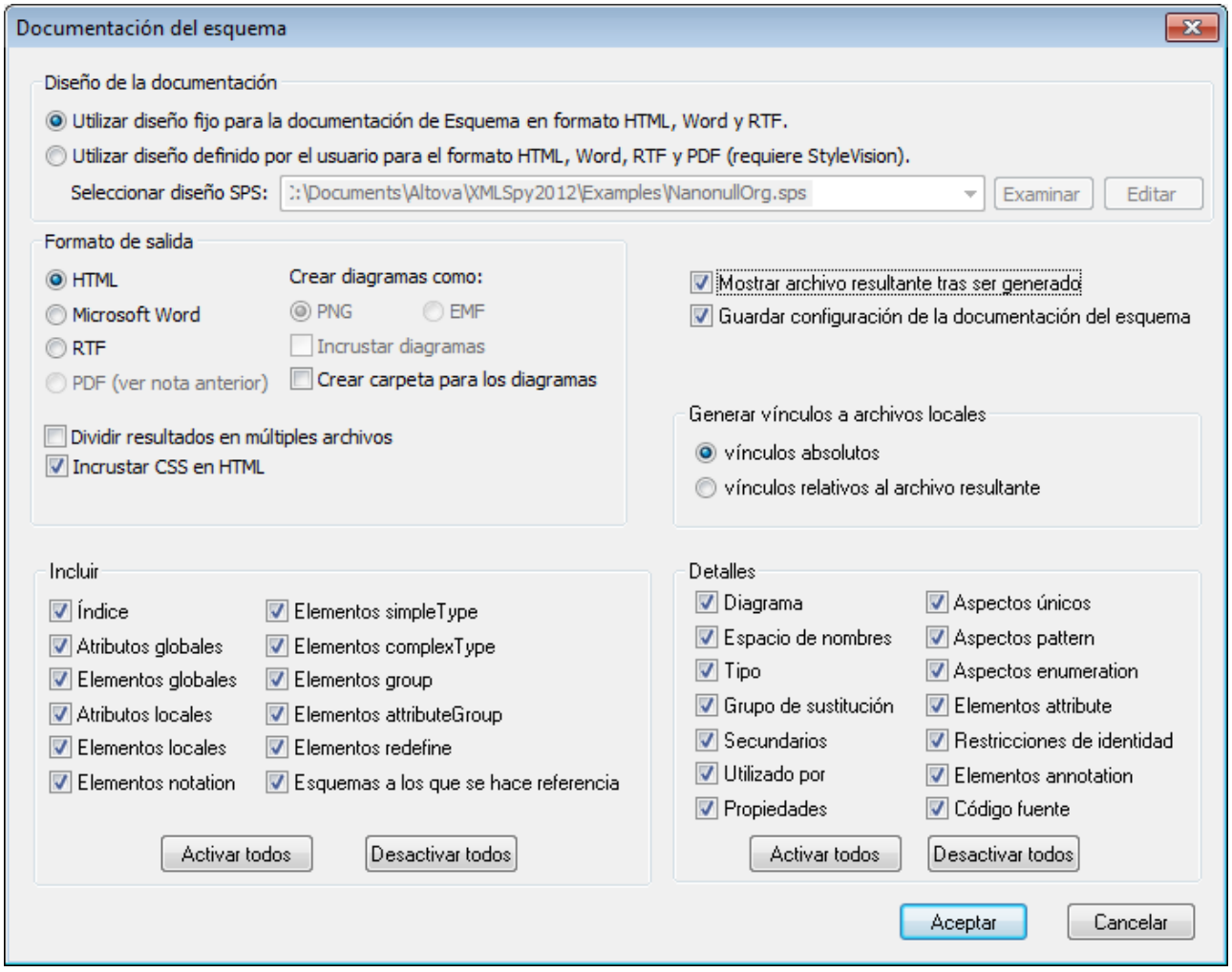
El comando **Diseño de esquemas | Generar documentación** genera documentación detallada sobre el esquema activo, ya sea un esquema XML o un esquema JSON. Esta sección explica cómo generar documentación para esquemas XML. El procedimiento para generar documentación para esquemas JSON es similar y se describe en la sección *Generar documentación para esquemas JSON*[751].

© 2016-2022 Altova GmbH

Generar documentación para esquemas XML

Si el documento activo es un esquema XML y hace clic en el comando **Diseño de esquemas | Generar documentación**, aparece el cuadro de diálogo "Documentación del esquema" (*imagen siguiente*), donde puede configurar la documentación antes de generarla.

En el panel *Diseño de la documentación* del cuadro de diálogo puede seleccionar si para generar la documentación se usa el diseño fijo de XMLSpy o un diseño personalizado y creado en un archivo SPS de StyleVision. Recuerde que la documentación se puede generar en formato PDF solamente si usa un archivo SPS de StyleVision, no con el diseño fijo. Para más información consulte el apartado Diseño definido por el usuario 1377.



A continuación explicamos el resto de opciones del cuadro de diálogo "Documentación del esquema":

- ***Formato de salida:*** en este panel puede configurar el formato de salida, que puede ser HTML, Microsoft Word, RTF o PDF. (El formato de salida PDF solamente está disponible si usa un archivo SPS de StyleVision para generar la documentación). Al hacer clic en **Aceptar**, se le pide un nombre para el archivo de salida y la ubicación donde desea guardarlo.

- Los documentos de Microsoft Word se crean con la extensión de archivo `.doc` si usa el diseño fijo para generar la documentación y con la extensión de archivo `.docx` si usa un archivo SPS de StyleVision.
- ***Dividir documentación en múltiples archivos:*** la documentación puede generarse en un solo archivo o dividirse en varios archivos. Si marca esta casilla, se genera un archivo por cada componente. Use las casillas del panel *Incluir* para especificar qué componentes se incluyen en la documentación. En la documentación de diseño fijo, se crean automáticamente hipervínculos entre los componentes relacionados entre sí.
- ***Incrustar CSS en HTML:*** si genera la documentación en formato HTML, las definiciones de estilos CSS se pueden guardar por separado en un archivo CSS o se pueden incrustar en el archivo HTML (en el elemento `<head>`). Si crea un archivo CSS, se crea con el mismo nombre que el archivo HTML, pero con la extensión `.css`. Active o desactive la casilla *Incrustar CSS en HTML* según corresponda.
- ***Incrustar diagramas:*** esta casilla se habilita si selecciona los formatos de salida MS Word, RTF o PDF. Si marca esta casilla, los diagramas se incrustan en el archivo de salida, en formato PNG. De lo contrario, los diagramas se crean como archivos PNG y la documentación incluye enlaces a estos archivos por medio de vínculos de objeto.
- ***Crear carpeta para los diagramas:*** si genera la documentación en formato HTML, todos los diagramas se crean como archivos PNG externos. Si marca la casilla *Crear carpeta para los diagramas*, se crea una carpeta en la carpeta donde está la documentación HTML y los archivos PNG se guardan dentro de ella. Esta carpeta tendrá un nombre tipo `NombreDelArchivoHTML_diagramas`. Si esta casilla no está marcada, los archivos PNG se guardan en el mismo directorio que la documentación HTML.
- ***Generar vínculos a archivos locales:*** en este panel puede elegir si los vínculos a archivos locales (como los archivos de imagen de los diagramas y el archivo CSS externo) son relativos o absolutos.
- ***Incluir:*** en este panel puede seleccionar qué componentes incluye la documentación. Todos los elementos marcados en este panel aparecerán en la documentación. Por ejemplo, si marca la casilla *Atributos locales*, la documentación incluirá una descripción para cada atributo local. Si marca la casilla *Índice*, todos los esquemas relacionados aparecerán al principio de la documentación y sus componentes globales aparecerán agrupados por tipos. Con los botones **Activar todos/Desactivar todos** puede seleccionar rápidamente todas las opciones del panel o anular la selección. Nota: las opciones elegidas en este panel no influyen en la representación gráfica del tipo de componente dentro de la documentación. La representación gráfica de los componentes viene dada por la configuración elegida en el cuadro de diálogo Configurar visualización del esquema[1378]. Es decir, si prefiere que los atributos no aparezcan en la representación gráfica de un elemento del esquema, entonces debe desactivar los atributos en el cuadro de diálogo Configurar visualización del esquema[1378].
- ***Detalles:*** en este panel puede seleccionar qué detalles se incluyen en la documentación de cada componente. Con los botones **Activar todos** y **Desactivar todos** puede activar/desactivar todas las opciones del panel con un solo clic.
- ***Mostrar archivo resultante tras ser generado:*** esta casilla está disponible para todos los formatos de salida. Si está marcada, los archivos resultantes se abren inmediatamente en la vista Explorador (documentación HTML), en MS Word (documentación Word) y en las aplicaciones predeterminadas para archivos .rtf (documentación RTF) y archivos .pdf (documentación PDF).

Valores de los parámetros

Si el archivo SPS de StyleVision contiene definiciones de parámetros, al hacer clic en **Aceptar** aparece un cuadro de diálogo con todos los parámetros definidos en el archivo SPS. Puede escribir nuevos valores de parámetro en este cuadro de diálogo para reemplazar los valores predeterminados que se asignaron en el archivo SPS.

© 2016-2022 Altova GmbH

28.7.3.2 Diseño definido por el usuario

Si lo prefiere, en lugar del diseño estándar fijo de XMLSpy, puede crear un diseño personalizado para la documentación del esquema. Este diseño personalizado se crea en un archivo SPS de StyleVision y se trata de una plantilla de diseño para el documento de salida.

Cómo crear un archivo SPS

Los archivos StyleVision Power Stylesheet (o SPS) se crean con Altova StyleVision. El archivo para generar documentación de esquema debe estar basado en el esquema XML que especifica la estructura de la documentación de esquema. Este esquema, que se llama `SchemaDocumentation.xsd`, viene con XMLSpy y está en la carpeta `C:\Documents and Settings\<nombreUsuario>\Mis Documentos\Altova\XMLSpy2023\Documentation\Schema`.

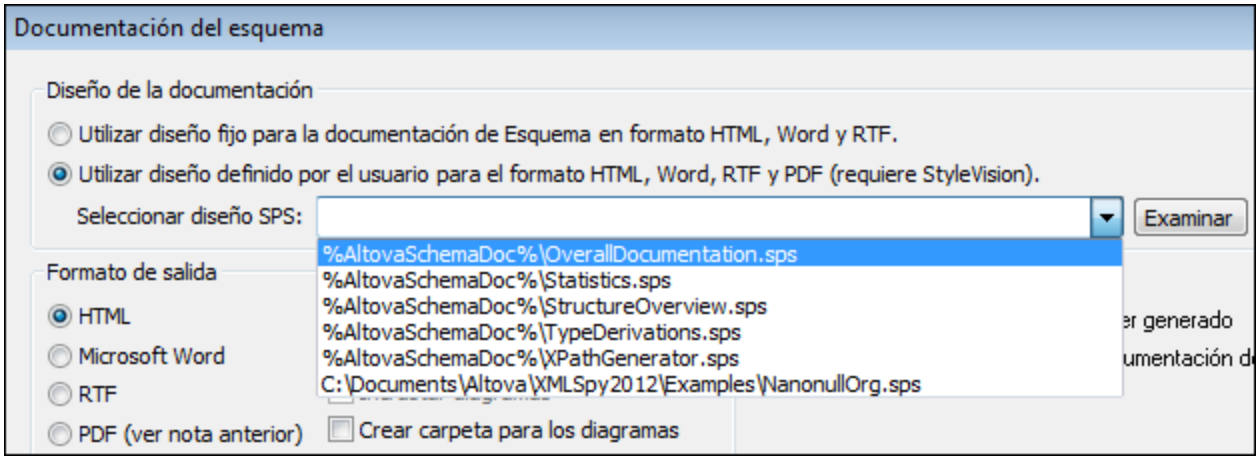
Para crear el diseño SPS en StyleVision arrastre nodos del esquema `SchemaDocumentation.xsd` hasta el panel de diseño y asígneles estilos y propiedades. También puede añadir otros componentes como vínculos, tablas e imágenes al diseño SPS. Es decir, todo el documento de salida se puede diseñar en el SPS. Para más información sobre cómo crear diseños SPS con StyleVision consulte el *Manual de usuario de StyleVision*.

La ventaja de usar un archivo SPS para generar documentación de esquema es que puede controlar por completo el diseño de la documentación. Además la documentación de esquema se puede generar en formato PDF solamente si usa un archivo SPS definido por el usuario. Si usa el diseño fijo de XMLSpy, la documentación no se puede generar en formato PDF.

Cómo seleccionar un archivo SPS para la documentación de esquema

Después de crearlo, puede usar el archivo SPS para generar la documentación de esquema. En el cuadro de diálogo "Documentación del esquema" (**Diseño de esquemas | Generar documentación**) puede seleccionar el archivo SPS que desea usar para generar la documentación del esquema. En el panel *Diseño de la documentación* de este cuadro de diálogo (*imagen siguiente*) seleccione el botón de opción *Utilizar diseño definido por el usuario*. Para buscar el archivo SPS haga clic en el botón **Examinar**. Cuando termine, haga clic en el botón **Aceptar**. Aparece el cuadro de diálogo "Guardar", seleccione la carpeta donde desea guardar el archivo de documentación y el nombre con el que desea guardar el archivo.

Nota: el archivo SPS debe localizar correctamente el esquema en el que se basa: `SchemaDocumentation.xsd` (tal y como se describe más arriba).



XMLSpy viene con varios diseños SPS editables para generar la documentación de esquema, que están en (Mis) Documentos[36]: `Altova\XMLSpy2023\Documentation\Schema\`. Se trata de estos archivos:

- `OverallDocumentation.sps`, que genera una documentación completa del esquema
- `Statistics.sps`, que ofrece el número de elementos globales y locales, los atributos y grupos de atributos y los tipos simples y complejos del esquema principal y de cada archivo de esquema
- `StructureOverview.sps`, que presenta la estructura de elementos globales y tipos complejos hasta un nivel determinado que puede configurar
- `TypeDerivations.sps`, que ofrece una lista en forma de árbol con los tipos simples y complejos y todos sus tipos derivados directos e indirectos
- `XPathGenerator.sps`, que genera todas las instrucciones XPath posibles hasta un nivel determinado que se puede configurar

Estos archivos (y los archivos SPS utilizados recientemente) están disponibles en la lista desplegable del cuadro combinado *Seleccionar diseño SPS* de la opción *Utilizar diseño definido por el usuario* (*imagen anterior*).

Si hace clic en el botón **Editar** del panel *Diseño de la documentación*, se inicia StyleVision y se abre el archivo SPS en una ventana de StyleVision. Para obtener una vista previa del documento resultante en StyleVision, necesita un archivo XML de trabajo. Los diseños SPS mencionados más arriba ya tienen asignado un archivo XMl de prueba llamado `Sample.xml` que viene con la aplicación y que está guardado en la carpeta (Mis) Documentos[36]:

```
Altova\XMLSpy2023\Documentation\Schema\SampleData
```

Nota: para generar documentación de esquema con un archivo SPS es necesario tener instalado StyleVision en el equipo.

28.7.4 Configurar opciones de vista

El comando **Diseño de esquemas | Configurar opciones de vista** permite configurar la vista del modelo de contenido y por eso solamente se habilita cuando está activa la vista del modelo de contenido de un componente. Al hacer clic en este comando, el cuadro de diálogo "Configurar visualización del esquema" aparece en la esquina inferior derecha de la ventana de XMLSpy, para que pueda ver el efecto que tienen en el

 © 2016-2022 Altova GmbH

la vista las opciones seleccionadas en este cuadro de diálogo. Los cambios no se guardan hasta que no haga clic en el botón **Aceptar** y afectan a la vista del modelo de contenido de todos los archivos de esquema XML que se abran a partir de ese momento.

Estas opciones también determinan el aspecto de la documentación del esquema y la copia impresa del esquema. Por ejemplo, si prefiere que los atributos no aparezcan en la representación gráfica de un elemento del esquema, entonces debe desactivar los atributos en el cuadro de diálogo Configurar visualización del esquema 1378.

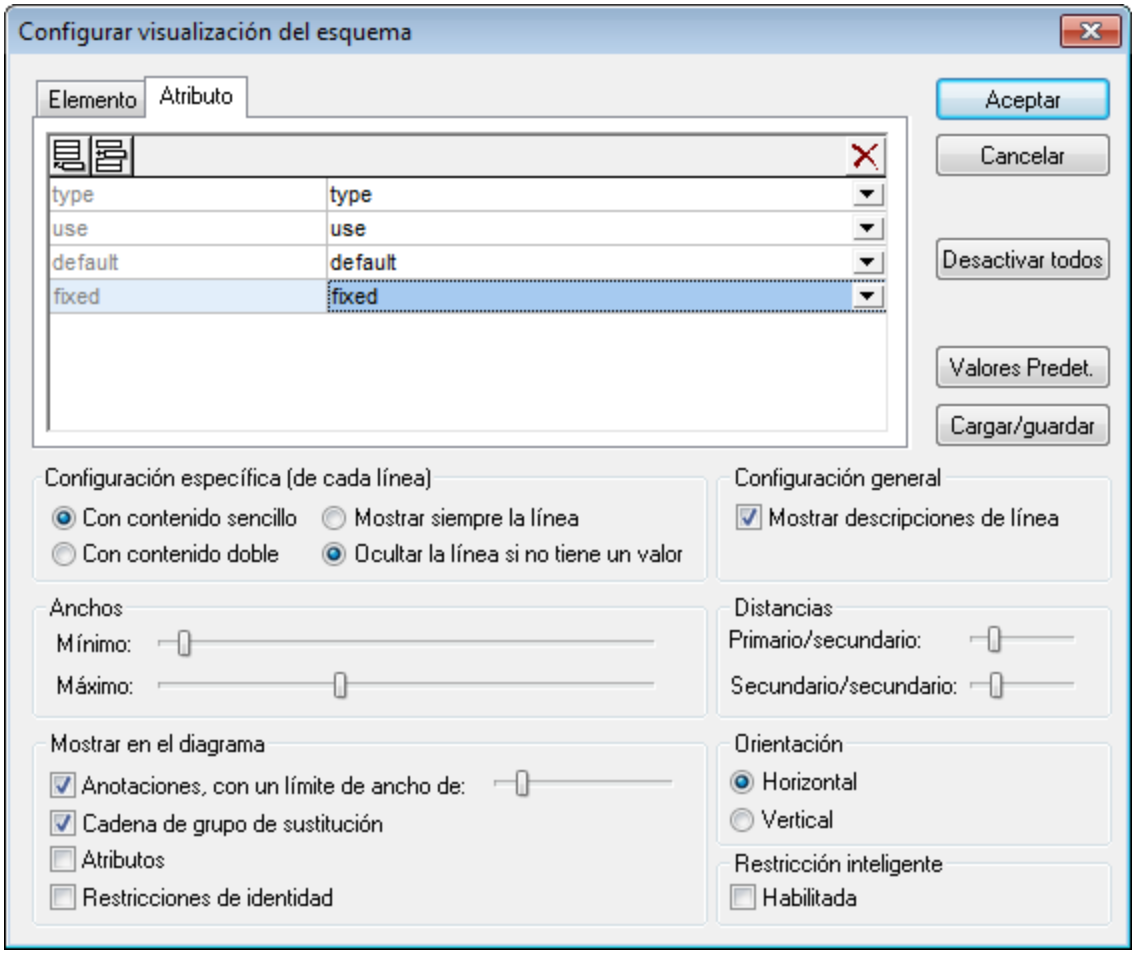
Nota: para más información sobre las opciones de configuración de la vista de diseño de esquemas JSON consulte la sección *Configurar la vista de diseño de esquemas JSON* 749.

Definir líneas de descripción de propiedades para el modelo de contenido

Puede definir qué propiedades de los elementos y atributos se muestran en la vista del modelo de contenido. Estas propiedades aparecen dentro del recuadro del componente, que se divide en celdas.

Siga estos pasos para definir las líneas de descripción de propiedades:

1. Seleccione **Diseño de esquemas | Configurar opciones de vista**. Aparece el cuadro de diálogo "Configurar visualización del esquema".
2. En la pestaña **Elemento** o **Atributo**, haga clic en el icono **Anexar**  o **Insertar**  para añadir un descriptor de propiedad. La línea del descriptor se añade en el cuadro de diálogo y en los recuadros de los elementos de la vista del modelo de contenido.
3. En el cuadro combinado seleccione la propiedad que desea mostrar (*imagen siguiente*).
4. Repita los pasos 2 y 3 tantas veces como propiedades desee mostrar en la vista del modelo de contenido.



La vista del modelo de contenido se actualiza y muestra los descriptores de propiedades en todos los elementos que tienen las propiedades elegidas.

Nota: En el caso de los atributos, la configuración definida aparece solamente cuando los atributos se muestran en el diagrama de la vista del modelo de contenido (y no en el panel inferior de la vista). La configuración definida afecta a todas las vistas del modelo de contenido que se abran a partir de ese momento.

Eliminar líneas de descripción de propiedades de la vista del modelo de contenido

Para eliminar descriptores de propiedades, seleccione el descriptor que desea eliminar en el cuadro de diálogo "Configurar visualización del esquema" y haga clic en el icono **Eliminar** .

Opciones para configurar la vista del modelo de contenido

La vista del modelo de contenido se puede configurar en el cuadro de diálogo "Configurar visualización del esquema". En los párrafos anteriores explicamos cómo añadir líneas de descripción de propiedades a la vista del modelo de contenido. Las demás opciones del cuadro de diálogo se describen a continuación.

© 2016-2022 Altova GmbH

Configuración específica (de cada línea)
Estas opciones se pueden configurar en cada una de las líneas, una por una. Primero seleccione la línea en el panel superior y después configure sus opciones. Puede definir si el descriptor de línea contiene contenido sencillo o doble y si las líneas del descriptor aparecen en todos los elementos o solamente en los elementos que tienen esa propiedad.

Configuración general
Esta opción oculta o muestra las líneas de los descriptores (es decir, el nombre de la propiedad).

Anchos
Estos controles deslizantes permiten establecer el tamaño mínimo y máximo que tienen los recuadros de los elementos en la vista del modelo de contenido. Si el texto del descriptor de propiedad es demasiado largo, puede ajustar el tamaño de los recuadros con estos controles.

Distancias
Estos controles deslizantes permiten definir la distancia horizontal y vertical que existe entre los elementos que aparecen en pantalla.

Mostrar en el diagrama
La casilla *Anotaciones* oculta o muestra el texto de las anotaciones. También puede definir el ancho del texto de las anotaciones con el control deslizante. Además puede ocultar o mostrar los grupos de sustitución. Si marca las casillas *Atributos* y *Restricciones de identidad*, los atributos y las restricciones de identidad aparecen en el diagrama del modelo de contenido. Si estas casillas no están marcadas, los atributos y las restricciones de identidad aparecen en el panel inferior de la vista del modelo de contenido.

Orientación
Estas opciones sirven para definir la orientación del diagrama en la pantalla.

Edición del modelo de contenido propiamente dicho
Puede cambiar las propiedades del elemento directamente en el diagrama del modelo de contenido. Para ello haga doble clic en la propiedad, introduzca un valor y pulse la tecla **Entrar**. En algunos casos podrá seleccionar una opción de una lista desplegable.

Botones del cuadro de diálogo "Configurar visualización del esquema"
En este cuadro de diálogo encontrará estos botones:

- El botón **Cargar/guardar** sirve para cargar y guardar las opciones de configuración elegidas en el cuadro de diálogo
- El botón **Valores Predet.** sirve para restaurar la configuración predeterminada
- El botón **Desactivar todos** vacía el panel que contiene la lista de propiedades

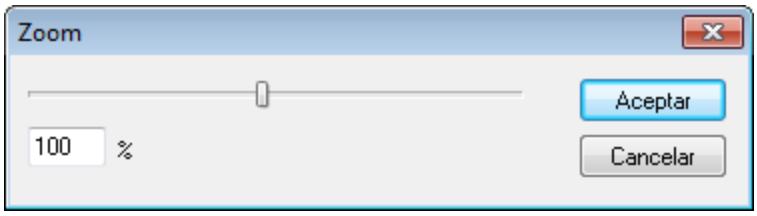
Habilitar restricciones inteligentes

Para habilitar las restricciones inteligentes 244, marque la casilla *Habilitada* del panel *Restricción inteligente.*

28.7.5 Zoom

El comando **Diseño de esquemas | Zoom** controla el nivel de zoom de la vista del modelo de contenido. Esta función es muy práctica si tiene un modelo de contenido muy grande y desea alejarse para ver todo el modelo

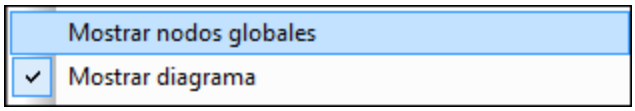
de contenido en la ventana principal. El factor de zoom se puede establecer entre un 10% y un 200% del tamaño real.



Para alejarse o acercarse a la vista del modelo de contenido, arrastre el control deslizante o haga clic en el cuadro de texto y escriba el valor porcentual.

28.7.6 Mostrar nodos globales

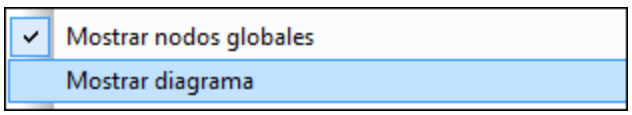
El comando **Diseño de esquemas | Mostrar nodos globales** cambia de la vista del modelo de contenido 190 a la vista general del esquema 182 para mostrar todos los componentes globales del esquema. Se trata de un comando de alternancia asociado al comando **Mostrar diagrama**. Una marca de verificación a la izquierda del comando indica cuál de estos dos comandos está seleccionado (*imagen siguiente*).



Otra opción para cambiar a la vista general del esquema es pulsar el icono **Mostrar nodos globales** situado en la esquina superior izquierda de la vista del modelo de contenido.

28.7.7 Mostrar diagrama

El comando **Diseño de esquemas | Mostrar diagrama** abre la vista del modelo de contenido 190 del componente global seleccionado, si este tiene un modelo de contenido. Los componentes globales que tienen un modelo de contenido (los tipos complejos, los elementos y los grupos de elementos) se identifican con el icono situado a la izquierda del componente. Este comando es un comando de alternancia asociado al comando **Mostrar nodos globales**. Una marca de verificación a la izquierda del comando indica cuál de estos dos comandos está seleccionado (*imagen siguiente*).



Hay dos maneras más de cambiar a la vista del modelo de contenido:

- Haga clic en el icono situado junto al componente cuyo modelo de contenido desea ver.
- Haga doble clic en el nombre del componente en el ayudante de entrada Componentes (situado en la parte superior derecha).

 © 2016-2022 Altova GmbH

28.7.8 Extensiones de esquema para BD

Al pasar el puntero por esta opción de menú aparece un submenú con comandos para trabajar con extensiones de esquema Oracle y MS SQL Server.

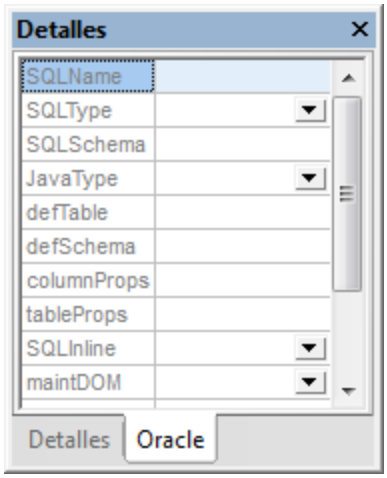
- Habilitar extensiones de esquema de Oracle 1383
- Configuración del esquema de Oracle 1384
- Habilitar extensiones de esquema de Microsoft SQL Server 1384
- Relaciones de esquema con nombre 1385
- Relaciones de elementos sin nombre 1386

28.7.8.1 Habilitar extensiones de esquema de Oracle

XMLSpy admite el uso de extensiones de esquema de Oracle para trabajar con Oracle 9i Project XDB. Si usa estas extensiones de esquema, puede configurar y personalizar cómo se almacenan los documentos XML en Oracle 9i Project XDB. A estos documentos XML se puede acceder después mediante consultas SQL y con otras herramientas. Para más información, visite el sitio web de Oracle.

Al hacer clic en el comando **Habilitar extensiones de esquema de Oracle** ocurre esto:

- El espacio de nombres XDB se declara en el elemento `schema` de esta forma: `xmlns:xdb="http://xmlns.oracle.com/xdb"`.
- Se crea la pestaña **Oracle** en el ayudante de entrada Detalles. Desde esta pestaña puede añadir atributos (incluidos los atributos propios de XDB) a elementos del esquema como `xsd:complexType` y `xsd:element`.

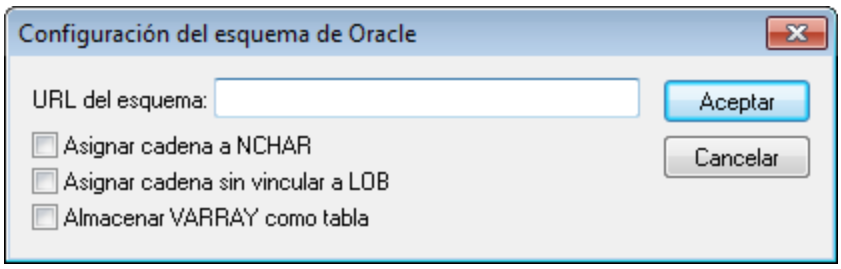


Puede definir extensiones de Oracle para tipos complejos, elementos y atributos. Para ello use el ayudante de entrada igual que siempre.

Nota: este comando de menú se puede activar o desactivar, es decir, las extensiones se pueden habilitar o deshabilitar. Cuando las extensiones de Oracle están habilitadas, aparece una marca de verificación a la izquierda del comando. Al deshabilitar las extensiones de Oracle (haciendo clic otra vez en el comando habilitado) se eliminan del archivo la declaración del espacio de nombres XDB y todas las extensiones XDB. Ya que esta acción no se puede deshacer, recibirá un mensaje de advertencia.

28.7.8.2 Configuración del esquema de Oracle

El comando **Configuración del esquema de Oracle** sirve para definir la configuración global de las extensiones de esquema de Oracle.



Para tener acceso a este cuadro de diálogo antes es necesario habilitar las extensiones de esquema de Oracle (con el comando **Habilitar extensiones de esquema de Oracle**[1383]).

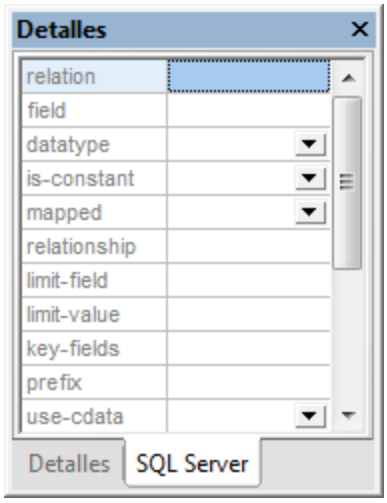
28.7.8.3 Habilitar extensiones de esquema de Microsoft SQL Server

XMLSpy admite el uso de extensiones de esquema de Microsoft SQL Server 2000 para trabajar con Microsoft SQL Server. Si usa estas extensiones de esquema, puede configurar y personalizar cómo se almacenan los documentos XML en Microsoft SQL Server. A estos documentos XML se puede acceder después mediante consultas SQL y con otras herramientas. Para más información, visite el sitio web de Microsoft.

Al hacer clic en el comando **Habilitar extensiones de esquema de Microsoft SQL Server** ocurre esto:

- El espacio de nombres de SQL Server se declara en el elemento `schema` de esta forma: `xmlns:sql="urn:schemas-microsoft-com:mapping-schema"`.
- Se crea la pestaña **SQL Server** en el ayudante de entrada Detalles. Desde esta pestaña puede añadir atributos a elementos del esquema como `xsd:element`.

© 2016-2022 Altova GmbH



Cuando seleccione un componente para el que se puedan definir extensiones de SQL Server, entonces se habilita la pestaña **SQL Server** del ayudante de entrada Detalles. Use el ayudante de entrada igual que siempre.

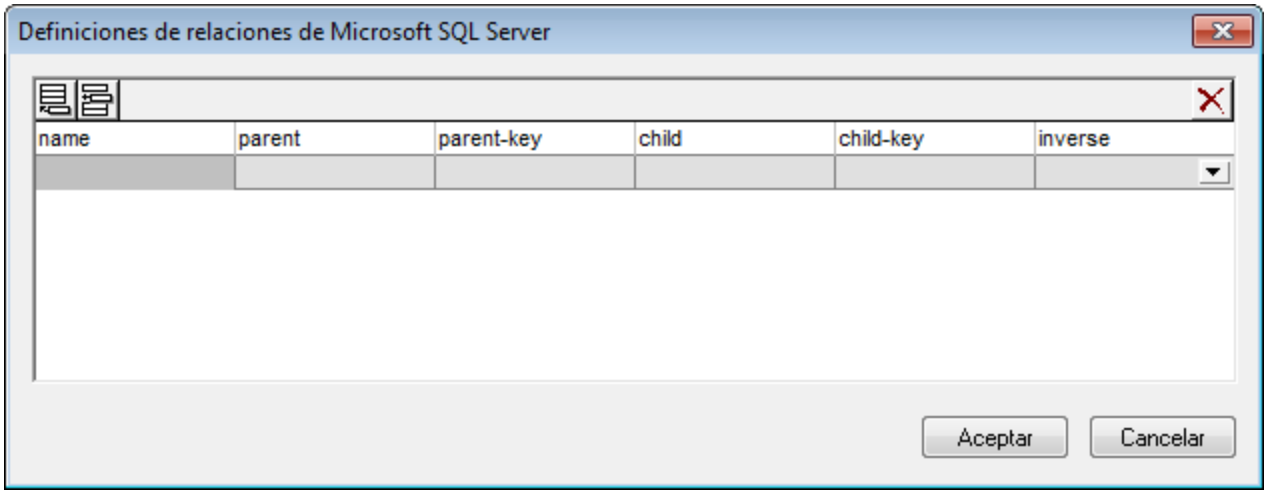
Nota: este comando de menú se puede activar o desactivar, es decir, las extensiones se pueden habilitar o deshabilitar. Cuando las extensiones de SQL Server están habilitadas, aparece una marca de verificación a la izquierda del comando. Al deshabilitar las extensiones de SQL Server (haciendo clic otra vez en el comando habilitado) se eliminan del archivo la declaración del espacio de nombres SQL Server y todas las extensiones SQL. Ya que esta acción no se puede deshacer, recibirá un mensaje de advertencia.

28.7.8.4 Relaciones de esquema con nombre

El comando **Relaciones de esquema con nombre** sirve para definir relaciones con nombre que ofrecen la información necesaria para crear la jerarquía del documento. Para poder usar este comando antes es necesario habilitar las extensiones de esquema de SQL Server, haciendo clic en el comando **Habilitar extensiones de esquema de Microsoft SQL Server**.

Para crear una relación de esquema con nombre:

1. Haga clic en el icono **Insertar** o **Anexar** para añadir una fila nueva al cuadro de diálogo.
2. Haga clic en el campo pertinente y escriba el nombre de la relación.
3. Para terminar haga clic en **Aceptar**.



Como resultado se genera un elemento SQL `relationship` y se coloca justo después de la declaración de espacio de nombres.

Nota: para eliminar una fila del cuadro de diálogo, seleccione la fila y haga clic en el icono Eliminar .

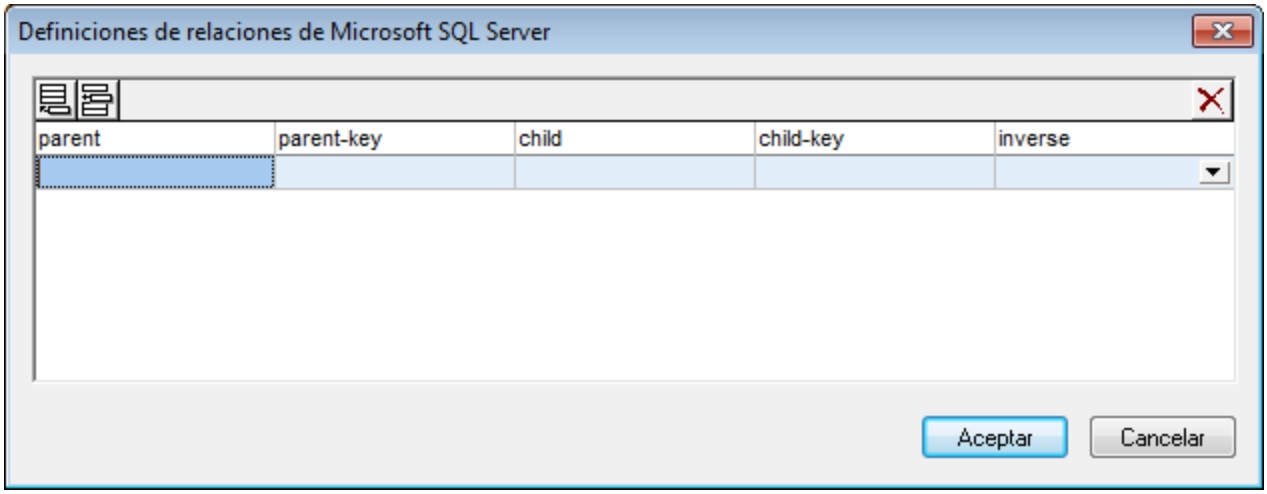
28.7.8.5 Relaciones de elementos sin nombre

El comando **Relaciones de elementos sin nombre** sirve para definir relaciones sin nombre que ofrecen la información necesaria para crear la jerarquía del documento. Para poder usar este comando antes es necesario habilitar las extensiones de esquema de SQL Server, haciendo clic en el comando **Habilitar extensiones de esquema de Microsoft SQL Server**.

Para crear una relación de esquema sin nombre:

1. Haga clic en el icono **Insertar** o **Anexar** para añadir una fila nueva al cuadro de diálogo.
2. Haga clic en el campo pertinente y escriba su nombre.
3. Para terminar haga clic en **Aceptar**.

© 2016-2022 Altova GmbH

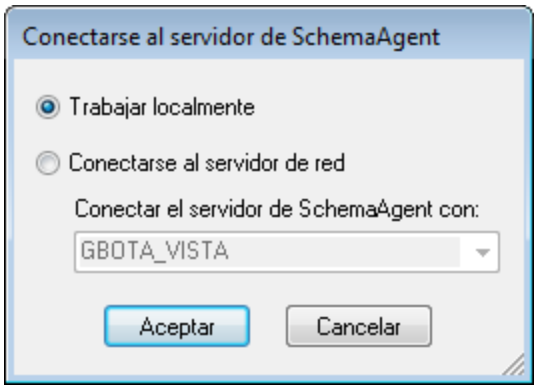


Como resultado se genera un elemento SQL `relationship` en el elemento de esquema que está seleccionado.

Nota: para eliminar una fila del cuadro de diálogo, seleccione la fila y haga clic en el icono Eliminar .

28.7.9 Conectarse al servidor de SchemaAgent

El comando **Diseño de esquemas | Conectarse al servidor de SchemaAgent Server** se habilita cuando está activo un documento XML y sirve para conectarse a un servidor de SchemaAgent Server. Para conectarse a un servidor de SchemaAgent es necesario tener instalado Altova SchemaAgent. Al hacer clic en este comando, aparece el cuadro de diálogo "Conectarse al servidor de SchemaAgent":



En este diálogo puede elegir si usa el servidor local (el servidor de SchemaAgent que viene con Altova SchemaAgent) o un servidor de red (el producto Altova SchemaAgent Server, que se puede descargar gratis). Si selecciona la opción *Trabajar localmente*, el servidor local de SchemaAgent se inicia nada más hacer clic

en **Aceptar** y se establece una conexión. Si elige la opción *Conectarse al servidor de red*, SchemaAgent Server tiene que estar funcionando para poder establecer la conexión.

Cuando está conectado a SchemaAgent Server, XMLSpy funciona como cliente de SchemaAgent y ofrece potentes funciones de edición y gestión de esquemas. Para más información sobre SchemaAgent y sobre cómo instalar y conectarse a SchemaAgent Server, consulte el apartado SchemaAgent[463] de la sección *DTD y Esquemas XML* del manual. Para más información sobre cómo instalar y trabajar con estos dos productos, consulte el *Manual del usuario de SchemaAgent* que viene con estos productos.

Después de conectarse a SchemaAgent Server, XMLSpy muestra un mensaje con información sobre la conexión en la barra superior de la ventana principal. A partir de ahora puede acceder a todos los esquemas y componentes de esquema de las rutas de búsqueda (carpetas) definidas para el servidor de SchemaAgent al que está conectado XMLSpy.

Nota: para que la conexión se establezca correctamente, es necesario tener instalada en el mismo equipo que XMLSpy la aplicación SchemaAgent Client con una licencia válida.

28.7.10 Desconectarse del servidor de SchemaAgent



El comando **Desconectarse del servidor de SchemaAgent Server** se habilita tras establecerse la conexión al servidor SchemaAgent. Al seleccionar este comando, XMLSpy se desconecta del servidor de SchemaAgent.

28.7.11 Mostrar en SchemaAgent

La opción de menú **Mostrar en SchemaAgent** sirve para ver en SchemaAgent el esquema activo y, si lo desea, los esquemas vinculados con él. (Para poder usar esta función es necesario tener instalado SchemaAgent en el mismo equipo que XMLSpy). Los esquemas se abren en SchemaAgent en un diseño nuevo.

Al pasar el ratón por encima de la opción de menú **Mostrar en SchemaAgent** aparece un submenú donde puede elegir qué esquemas se muestran en SchemaAgent. Estas opciones se describen en el apartado SchemaAgent[472] de la sección *DTD y esquemas XML* del manual del usuario.

28.7.12 Validación con SchemaAgent

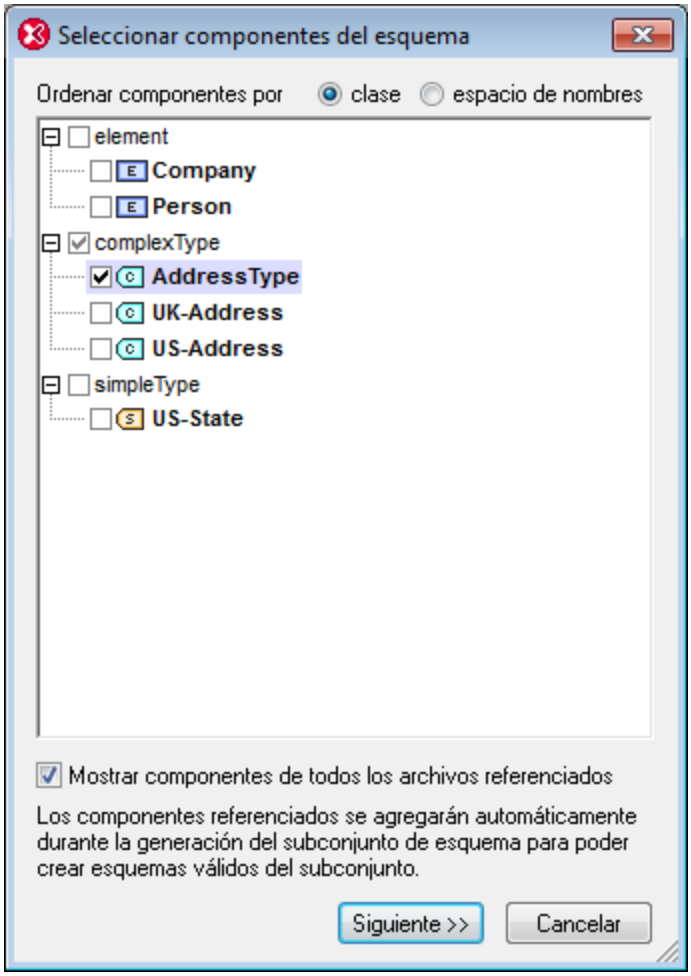


© 2016-2022 Altova GmbH

El comando **Validación con SchemaAgent** sirve para validar el esquema activo, así como los esquemas relacionados con él. Esta función se describe más detalladamente en el apartado Validación con SchemaAgent[472] de la sección *Vista Esquema* del manual del usuario.

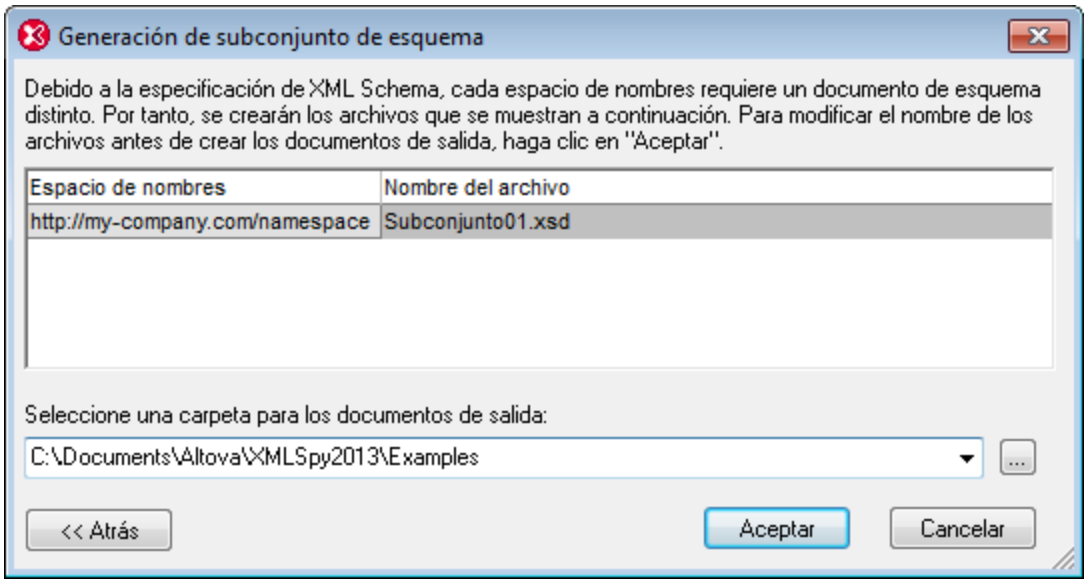
28.7.13 Crear subconjunto de esquema

Al hacer clic en el comando **Crear subconjunto de esquema** aparece el cuadro de diálogo "Seleccionar componentes del esquema" (*imagen siguiente*). En este cuadro de diálogo puede seleccionar qué componentes desea usar para crear un subconjunto de esquema. Después de seleccionarlos, haga clic en **Siguiente**. (Observe que, si marca la casilla que aparece en la parte inferior del diálogo, el diálogo enumera también los componentes de archivos referenciados, para que pueda seleccionarlos.)



En el siguiente cuadro de diálogo que aparece (*imagen siguiente*), escriba el nombre que desea darle a los archivos del paquete del subconjunto de esquema. También es necesario especificar en qué carpeta se guardan los archivos del subconjunto de esquema. El paquete del subconjunto de esquema puede tener múltiples archivos si alguno de los componentes que se crean es un componente importado en el esquema original. Por cada espacio de nombres del subconjunto de esquema se crea un archivo de esquema distinto. Los nombres de archivo que aparecen en este cuadro de diálogo son los nombres de los archivos originales.

Pero, si desea guardar los archivos en la misma carpeta que los archivos originales, entonces debe usar nombres de archivo nuevos porque no está permitido sobrescribir los archivos originales.



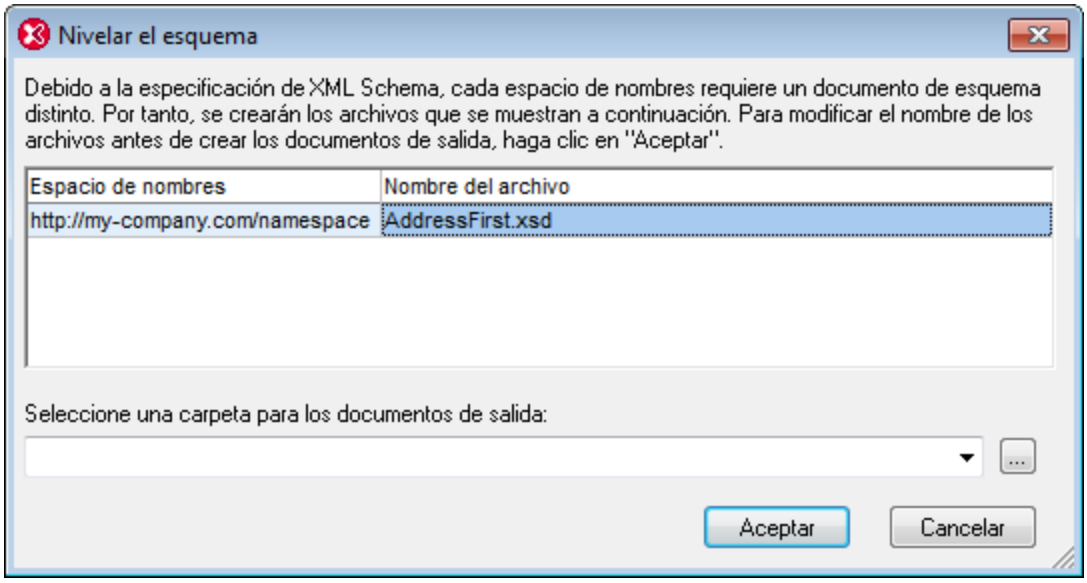
Al hacer clic en **Aceptar**, se abre en la vista Esquema el archivo del subconjunto de esquema que tiene el mismo espacio de nombres que el archivo activo. El resto de archivos del paquete se crean pero no se abren en la vista Esquema.

28.7.14 Nivelar el esquema

El proceso de nivelar el esquema activo en la vista Esquema consiste en (i) añadir los componentes de todos los esquemas incluidos como componentes globales del esquema activo y (ii) eliminar los esquemas incluidos.

Para nivelar el esquema activo, seleccione el comando **Diseño de esquemas | Nivelar el esquema**. Aparece el cuadro de diálogo "Nivelar el esquema" (*imagen siguiente*), que contiene los nombres de los archivos incluidos, uno por cada espacio de nombres que habrá en el esquema una vez nivelado. Estos nombres predeterminados se corresponden con los nombres de los archivos originales. Pero, si desea guardar los archivos en la misma carpeta que los archivos originales, entonces debe usar nombres de archivo nuevos porque no está permitido sobrescribir los archivos originales. En el cuadro de diálogo también puede seleccionar la carpeta donde se deben guardar el esquema nivelado y los archivos asociados.

© 2016-2022 Altova GmbH



Al hacer clic en **Aceptar** se abre el archivo de esquema nivelado en la vista Esquema.

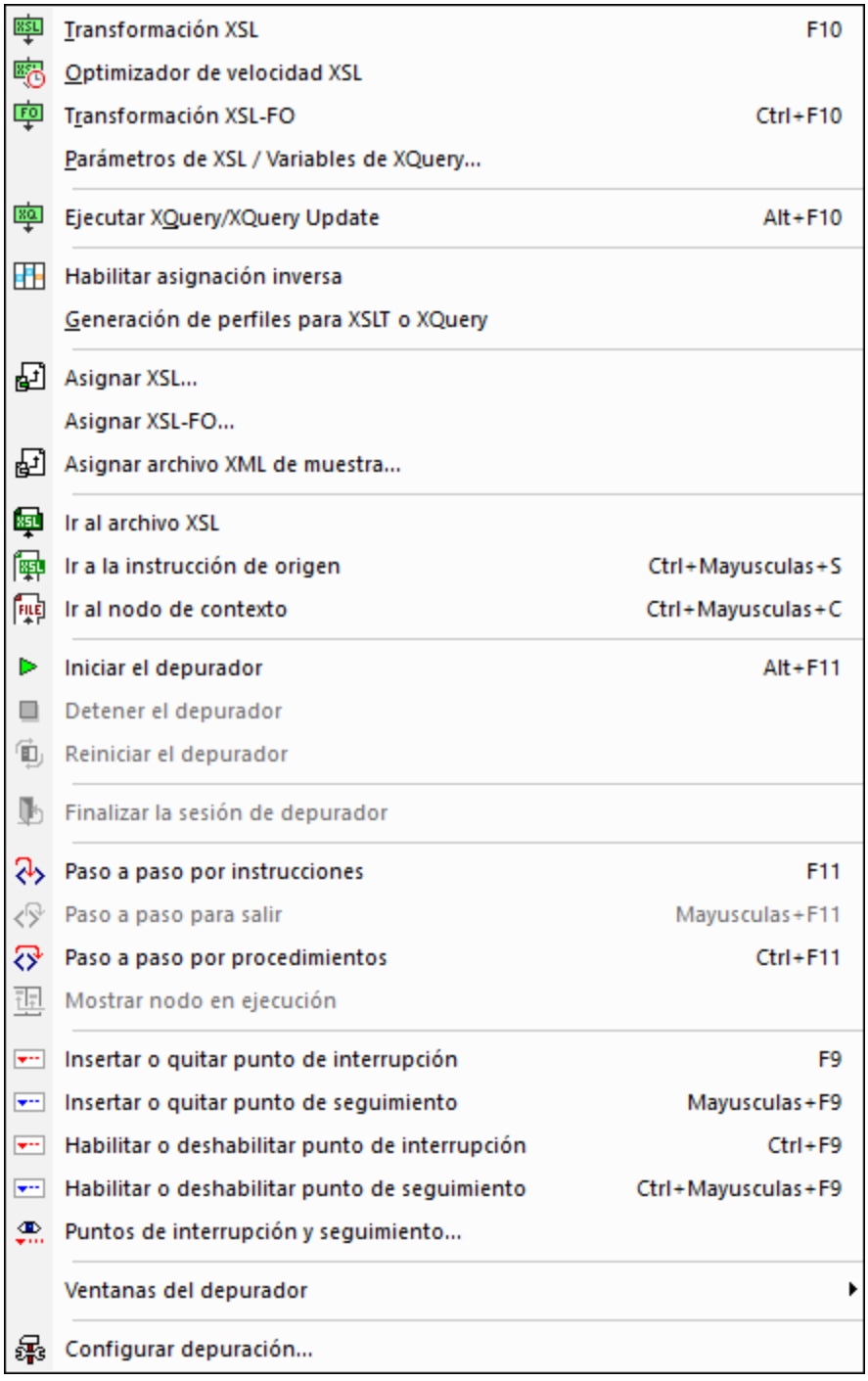
28.8 Menú XSL/XQuery

El lenguaje de transformación XSL permite especificar cómo se debe convertir un documento XML en otros documentos XML o archivos de texto. Un tipo de documento XML que se genera con un documento XSLT es el documento FO, que más adelante se puede procesar para generar archivos PDF. XMLSpy contiene procesadores XSLT integrados (para XSLT 1.0, 2.0 y 3.0) y puede conectarse a un procesador FO del sistema del usuario para transformar archivos XML y generar archivos en diferentes formatos de salida. La ubicación del procesador FO debe indicarse en la sección *XSL* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**[1612]) para que XMLSpy pueda utilizarlo.

XMLSpy también incluye un motor integrado XQuery, que se puede usar para ejecutar documentos XQuery (con o sin referencia a un documento XML).

Los comandos necesarios para realizar estas transformaciones están disponibles en el menú **XSL/XQuery**. Además, este menú incluye los comandos necesarios para trabajar con el depurador XSLT/XQuery de Altova.

© 2016-2022 Altova GmbH


Transformación XSL F10
Optimizador de velocidad XSL
Transformación XSL-FO Ctrl+F10
Parámetros de XSL / Variables de XQuery...
Ejecutar XQuery/XQuery Update Alt+F10
Habilitar asignación inversa
Generación de perfiles para XSLT o XQuery
Asignar XSL...
Asignar XSL-FO...
Asignar archivo XML de muestra...
Ir al archivo XSL
Ir a la instrucción de origen Ctrl+Mayusculas+S
Ir al nodo de contexto Ctrl+Mayusculas+C
Iniciar el depurador Alt+F11
Detener el depurador
Reiniciar el depurador
Finalizar la sesión de depurador
Paso a paso por instrucciones F11
Paso a paso para salir Mayusculas+F11
Paso a paso por procedimientos Ctrl+F11
Mostrar nodo en ejecución
Insertar o quitar punto de interrupción F9
Insertar o quitar punto de seguimiento Mayusculas+F9
Habilitar o deshabilitar punto de interrupción Ctrl+F9
Habilitar o deshabilitar punto de seguimiento Ctrl+Mayusculas+F9
Puntos de interrupción y seguimiento...
Ventanas del depurador
Configurar depuración...

28.8.1 Transformación XSL



F10

El comando **XSL/XQuery | Transformación XSL** transforma un documento XML usando una hoja de estilos XSLT. La transformación se puede realizar usando el motor XSLT integrado de Altova (el motor XSLT 1.0 para las hojas de estilos XSLT 1.0, el motor XSLT 2.0 para las hojas de estilos XSLT 2.0 y el motor XSLT 3.0 para hojas de estilos XSLT 3.0), el módulo MSXML de Microsoft o un procesador XSLT externo. El procesador que se usa junto con este comando se especifica en la sección *XSL* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).

Si el documento XML contiene una referencia a una hoja de estilos XSLT, esta hoja de estilos se usa para la transformación. (Para asignar una hoja de estilos XSLT al documento XML utilice el comando Asignar XSL. Si el documento forma parte de un proyecto, puede asignar una hoja de estilos XSLT a cada carpeta del proyecto, usando el cuadro de diálogo Propiedades. Haga clic con el botón derecho en la carpeta de proyecto o en el archivo que desea transformar y seleccione el comando **Transformación XSL**.) Si no se asignó ninguna hoja de estilos al archivo XML, aparece un mensaje solicitando la hoja de estilos XSLT que se debe usar. También puede seleccionar un archivo mediante un recurso global o una URL (haga clic en el botón **Examinar**) o un archivo de las ventanas XMLSpy que están abiertas (haga clic en el botón **Ventana**).

Automatizar transformaciones XSLT con RaptorXML 2023

RaptorXML es una aplicación independiente de Altova para validar XML, transformar XSLT y ejecutar XQuery. Se puede usar desde la línea de comandos, por una interfaz COM, en programas Java y en aplicaciones .NET. Por tanto, puede automatizar las tareas de transformación XSLT con RaptorXML. Por ejemplo, puede crear un archivo de procesamiento por lotes que llame a RaptorXML para transformar un grupo de documentos. Para más información, consulte la documentación de RaptorXML.

Transformaciones con archivos ZIP como destino

Para desviar los documentos de salida de las transformaciones a un archivo ZIP (incluidos los documentos OOXML, como `.docx`, por ejemplo) es necesario especificar el protocolo ZIP en la ruta de acceso del archivo de salida. Por ejemplo:

```
nombreArchivo.zip|zip/nombreArchivo.xxx

nombreArchivo.docx|zip/nombreArchivo.xxx
```

Nota: puede que sea necesario crear la estructura de directorios antes de ejecutar la transformación. Si está generando archivos para un archivo Open Office XML, es necesario comprimir los ficheros del archivo para poder crear el archivo OOXML de nivel superior (por ejemplo, `.docx`).

28.8.2 Optimizador de velocidad XSL



El comando **Optimizador de velocidad XSL** se habilita si el documento activo es un archivo XML o XSLT. Este comando inicia el optimizador de velocidad XSL, que analiza si es posible realizar transformaciones más

rápidas con la hoja de estilos analizada. Así es como funciona: el optimizador ejecuta la hoja de estilos en un conjunto de datos (es decir, en documentos XML) y analiza su rendimiento. A partir de este análisis se deriva una estrategia de optimización que se puede guardar con la hoja de estilos XSLT (en forma de instrucción de procesamiento al final de la hoja de estilos). La hoja de estilos optimizada se puede utilizar a partir de ahora para realizar transformaciones más rápidas.

Al hacer clic en este comando, XMLSpy solicita que seleccione un documento XML (si el documento activo es un XSLT) o una hoja de estilos XSLT (si el documento activo es un XML). Al hacer clic en **Aceptar** se inicia el análisis. Si la hoja de estilos XSLT o el documento XML ya tiene asignado un documento XML[1404] o un documento XSLT[1403], respectivamente, el análisis empieza inmediatamente.

Para aprender a usar el optimizador consulte la sección Optimizador de velocidad XSL[503]. Además puede configurar el optimizador en la sección[1612] *XSL: Optimizador de velocidad XSL*[1612] del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).

28.8.3 Transformación XSL-FO



Ctrl+F10

FO es un formato XML que describe documentos paginados. Los procesadores FO, como el procesador FOP de Apache XML Project, toman un archivo FO de entrada y generan un archivo PDF de salida. Esto significa que el proceso para generar documentos PDF a partir de documentos XML tiene dos fases:

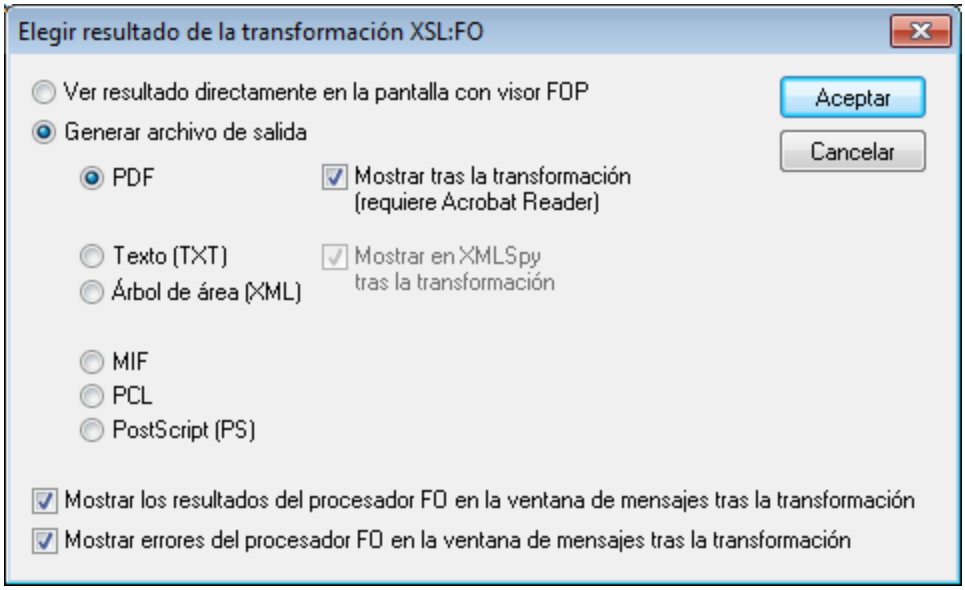
1. Primero el documento XML se transforma en un documento FO usando una hoja de estilos XSLT.
2. Después el procesador FO procesa el documento FO y genera un archivo PDF.

El comando **XSL/XQuery | Transformación XSL-FO** transforma un documento XML o un documento FO en PDF.

- Si ejecuta el comando **Transformación XSL-FO** en un documento XML de origen, entonces se ejecutan los dos pasos descritos anteriormente en secuencia, uno después del otro. Si en el documento XML no hay una referencia a la hoja de estilos XSLT necesaria para transformarlo en documento FO, la aplicación solicita que le asigne una hoja de estilos. Recuerde que también puede seleccionar un archivo mediante un recurso global o una URL (botón **Examinar**[1262]) o seleccionar uno de los archivos que están abiertos en las ventanas de XMLSpy (botón **Ventana**). La transformación del documento XML en XSL-FO la realiza el procesador XSLT que se indica en la sección[1612] *XSL*[1612] del cuadro de diálogo "Opciones" (**Herramientas | Opciones**). El procesador XSLT predeterminado es el procesador XSLT integrado de XMLSpy. El documento FO resultante lo procesa el procesador FO que se indica en la sección[1612] *XSL*[1612] en el cuadro de diálogo "Opciones" (**Herramientas | Opciones**).
- Si el comando **Transformación XSL-FO** se ejecuta en un documento FO, el documento se procesa con el procesador FO indicado en la sección[1612] *XSL*[1612] del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).

Resultado de la transformación XSL:FO

Al hacer clic en el comando **Transformación XSL-FO** aparece el cuadro de diálogo "Elegir resultado de la transformación XSL:FO" (*imagen siguiente*). Recuerde que si el documento activo es un documento XML que no tiene asignado ninguna hoja de estilos XSLT, XMLSpy primero le pide un archivo XSLT.



El resultado generado por el procesador FO se puede ver directamente en pantalla usando un visor FOP o puede generar un archivo de salida en uno de estos formatos: PDF, texto, árbol de área, MIF, PCL o PostScript. Si lo desea, los mensajes del procesador FO y los mensajes de error del procesador pueden aparecer en la ventana Mensajes. Para activar estas opciones marque las casillas correspondientes situadas al final del cuadro de diálogo.

Nota:

- A no ser que desactive la casilla correspondiente en el programa de instalación de XMLSpy, el procesador FOP de Apache XML Project se instalará en la carpeta `C:\ProgramData\Altova\SharedBetweenVersions`. Y si el procesador está instalado, su ruta de acceso aparecerá automáticamente en la sección *XSL* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**). Si quiere puede cambiar esta ruta de acceso para usar otro procesador FO distinto.
- El comando **Transformación XSL-FO** no solamente se puede usar con el archivo activo en la ventana principal. También puede usarlo con cualquier archivo o carpeta del proyecto activo. Para ello, seleccione los archivos o carpetas en la ventana Proyecto, haga clic con el botón derecho y seleccione el comando **Transformación XSL-FO**. Para la transformación se usa la hoja de estilos XSLT asignada a la carpeta de proyecto seleccionada.

28.8.4 Parámetros de XSL / Variables de XQuery

El comando **XSL/XQuery | Parámetros de XSL / Variables de XQuery** abre el cuadro de diálogo "Parámetros de entrada de XSL y variables externas de XQuery" (*imagen siguiente*). Aquí puede escribir el nombre de los parámetros que desea pasar a la hoja de estilos XSLT y las variables externas de XQuery que desea pasar al documento XQuery, así como sus correspondientes valores. En XMLSpy estos parámetros se usan de esta manera:

© 2016-2022 Altova GmbH

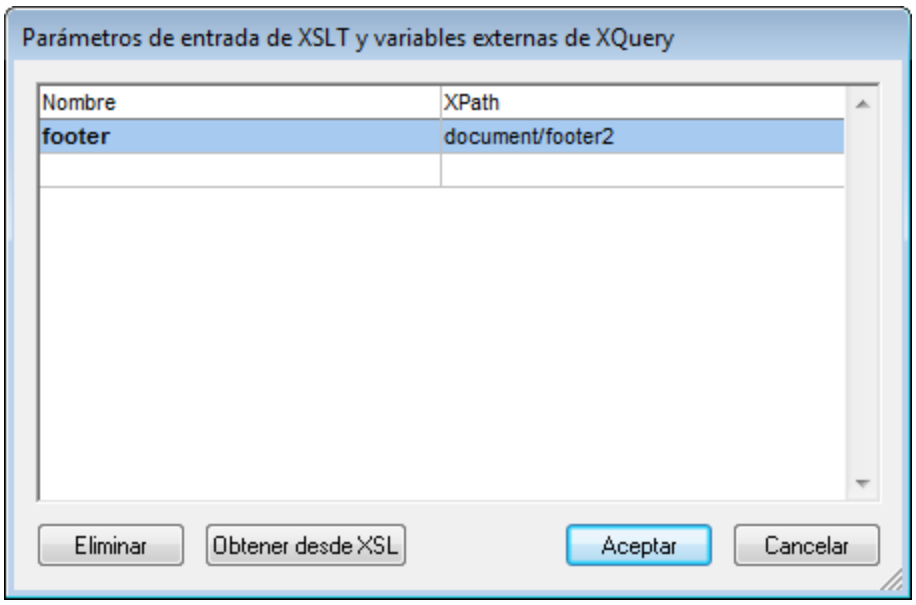
- Cuando se usa el comando **XSL/XQuery | Transformación XSL** para transformar un documento XML, los valores del parámetros guardados en este cuadro de diálogo se pasan al archivo XSLT seleccionado y utilizado para la transformación.
- Cuando se usa el comando **XSL/XQuery | Ejecutar XQuery** para procesar un documento XQuery, los valores de las variables externas de XQuery guardadas en este cuadro de diálogo se pasan al documento XQuery para la ejecución.

Nota: los parámetros y las variables que introduzca en este cuadro de diálogo se pasan solamente al motor XSLT integrado de Altova. Por tanto, si usa MSXML o un motor externo, estos parámetros no se pasan al motor de procesamiento.

Nota: no es erróneo definir una variable externa de XQuery (o un parámetro de XSLT) en el diálogo "Parámetros de entrada de XSL y variables externas de XQuery" y después no utilizarla en el documento XQuery. Tampoco es erróneo definir un parámetro de XSLT (o una variable externa de XQuery) en este cuadro de diálogo y después no usarlo en una transformación XSLT.

Cómo usar los parámetros de XSLT

El valor del parámetro puede ser una expresión XPath sin comillas o una cadena de texto delimitada por comillas. Si el documento activo es un archivo XSLT, entonces está habilitado el botón **Obtener desde XSL**. Al hacer clic en este botón se insertan en el cuadro de diálogo los parámetros declarados en el archivo XSLT y sus valores predeterminados. Es decir, este botón le permite incluir rápidamente los parámetros declarados y después cambiar sus valores predeterminados si lo necesita.



Nota: una vez introducidos en este cuadro de diálogo, los pares de parámetros y valores se usan para todas las transformaciones subsiguientes hasta que se eliminen del cuadro de diálogo o se reinicie la aplicación. Los parámetros introducidos en este cuadro de diálogo se definen para toda la aplicación y se pasan al documento XSLT correspondiente en cada transformación que se realice a partir de ese momento a través del entorno IDE. En otras palabras:

- los parámetros no están asociados a ningún tipo de documento en particular
- los parámetros introducidos en el cuadro de diálogo "Parámetros de entrada de XSL y variables externas de XQuery" se borran cuando se cierra XMLSpy.

Ejemplo de uso de parámetros de XSLT

En este ejemplo tenemos un documento XML que contiene los nombres de países y sus respectivas capitales:

```
<document>
   <countries>
      <country name="USA" capital="Washington DC"/>
      <country name="UK" capital="London"/>
      <country name="France" capital="Paris"/>
      <country name="Russia" capital="Moscow"/>
      <country name="China" capital="Beijing"/>
   </countries>
</document>
```

El siguiente documento XSLT generará un documento XML que muestre uno de esos países con su capital. Se selecciona un país introduciendo su nombre como valor del parámetro llamado **country** (resaltado en amarillo en el ejemplo siguiente).

```
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:param name="country" select="'USA'"/>
   <xsl:template match="countries">
      <xsl:for-each select="country[@name=$country]">
         <country>
            <name><xsl:value-of select="$country"/></name>
            <capital><xsl:value-of select="@capital"/></capital>
         </country>
      </xsl:for-each>
   </xsl:template>
</xsl:stylesheet>
```

Al ejecutar este documento XSLT en el documento XML mencionado más arriba, el resultado será el siguiente:

```
<country><name>USA</name><capital>Washington DC</capital></country>
```

Si se crea un parámetro llamado **country** en el cuadro de diálogo Parámetros de entrada XSLT/Variables externas XQuery y se le da un valor (*imagen anterior*), este valor pasará al parámetro **country** en la hoja de estilos para la transformación.De esta manera puede pasar distintos valores a diferentes parámetros en tiempo de ejecución.

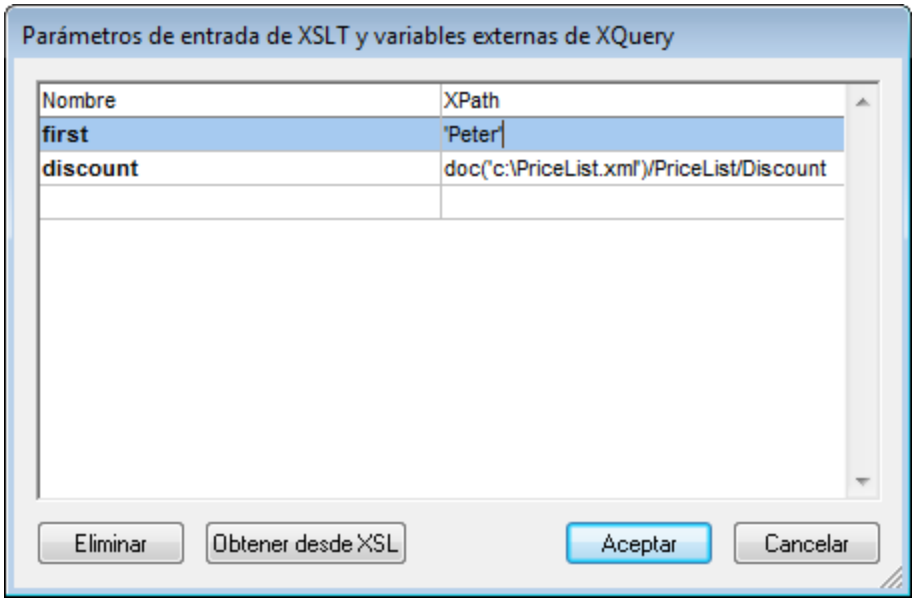
Nota:

- si usa el comando **XSL/XQuery | Transformación XSL-FO**, entonces los parámetros introducidos en el cuadro de diálogo "Parámetros de entrada de XSLT y variables externas de XQuery" **no se pasan** a la hoja de estilos. Para que estos parámetros se usen en el archivo PDF de salida, antes es necesario transformar el documento XML en FO con el comando **XSL/XQuery | Transformación XSL**) y después transformar el FO en PDF con el comando **XSL/XQuery | Transformación XSL-FO**.
- si usa un procesador XSLT que no sea uno de los motores XSLT integrados de Altova, los parámetros que introduzca en el diálogo "Parámetros de entrada de XSLT y variables externas de XQuery" no se pasan al procesador externo.

© 2016-2022 Altova GmbH

Cómo usar las variables externas de XQuery

El valor de las variables externas de XQuery puede ser una expresión XPath sin comillas o una cadena de texto delimitada por comillas. El tipo de datos de la variable externa se especifica en la declaración de variable en el documento XQuery.



Nota: una vez introducidas en este cuadro de diálogo, las variables externas de XQuery se utilizan para todas las ejecuciones subsiguientes hasta que se borren del cuadro de diálogo o se reinicie la aplicación. Las variables introducidas en este cuadro de diálogo se definen para toda la aplicación y se pasan al documento XQuery correspondiente en cada ejecución que se realice a partir de ese momento a través del entorno IDE. En otras palabras:

- las variables no están asociadas a ningún tipo de documento en particular
- las variables introducidas en el cuadro de diálogo "Parámetros de entrada de XSL y variables externas de XQuery" se borran cuando se cierra XMLSpy.

Ejemplo de uso de variables externas de XQuery

En este ejemplo, en el documento XQuery se declaró una variable `$first` que se utiliza en la cláusula `return` de la expresión FLWOR:

```
xquery version "1.0";
declare variable $first as xs:string external;
let $last := "Jones"
return concat($first, " ", $last )
```

Este XQuery devuelve `Peter Jones`, si el valor de la variable externa (introducida en el cuadro de diálogo "Parámetros de entrada de XSL y variables externas de XQuery") es `Peter`. Es necesario tener en cuenta que:

- La palabra clave `external` de la declaración de variable del documento XQuery indica que esta variable es una variable externa.

- Definir el tipo estático de la variable es opcional. Si no se indica un tipo de datos para la variable en la declaración de variable, entonces al valor de la variable se le asigna el tipo `xs:untypedAtomic`.
- Si se declara una variable externa en el documento XQuery, pero no se pasa al documento XQuery ninguna variable externa que tenga ese mismo nombre, entonces se emite un mensaje de error.
- Si se declara y se introduce una variable externa en el cuadro de diálogo "Parámetros de entrada de XSL y variables externas de XQuery", entonces se considera que está en el ámbito del documento XQuery que se está ejecutando. Si se declara una variable nueva con el mismo nombre dentro del documento XQuery, la variable nueva reemplaza temporalmente a la variable externa. Por ejemplo, el XQuery que aparece a continuación devuelve `Paul Jones` aunque la variable externa `$first` que está en el ámbito del documento tiene el valor `Peter`.

```
xquery version "1.0";
declare variable $first as xs:string external;
let $first := "Paul"
let $last := "Jones"
return concat($first, " ", $last )
```

28.8.5 Ejecutar XQuery/XQuery Update



El comando **XSL/XQuery | Ejecutar XQuery/XQuery Update** ejecuta un documento XQuery 1.0/3.1 o XQuery Update (1.0/3.0). Dependiendo de si el archivo seleccionado es un archivo XQuery o un archivo XQuery Update, se ejecuta XQuery o XQuery Update. XMLSpy reconoce el tipo de documento por la asociación de tipo de archivo 1587 (definida en la sección 1587 *Tipos de archivo* 1587 del cuadro de diálogo "Opciones" 1587).

XMLSpy selecciona automáticamente el motor XQuery adecuado en función de la declaración de versión que haya en el documento. Si el documento no incluye una declaración de versión, entonces se utiliza la versión predeterminada que se indicara en la sección 1616 *XQuery* 1616 del cuadro de diálogo "Opciones" 1616. El comando **Ejecutar XQuery/XQuery Update** se puede invocar tanto cuando el archivo activo es un documento XQuery como cuando es un XML. Cuando se invoca desde un archivo XML, aparece un cuadro de diálogo pidiendo que se asocie un archivo XQuery al archivo XML. También puede seleccionar un archivo a través de un recurso global o una URL (clic en el botón **Examinar** 1262) o seleccionar un archivo de las ventanas que están abiertas en XMLSpy (clic en el botón **Ventana**).

Nota: este comando también está disponible en el menú contextual de los elementos que muestra la ventana Proyecto 122.

Automatizar ejecuciones de XQuery con RaptorXML 2023

RaptorXML es una aplicación independiente de Altova para validar XML, transformar XSLT y ejecutar XQuery. Se puede usar desde la línea de comandos, por una interfaz COM, en programas Java y en aplicaciones .NET. Por tanto, puede automatizar las tareas de ejecución de XQuery con RaptorXML. Por ejemplo, puede crear un archivo de procesamiento por lotes que llame a RaptorXML para ejecutar un grupo de documentos XQuery. Para más información, consulte la documentación de RaptorXML.

© 2016-2022 Altova GmbH

28.8.6 Habilitar asignación inversa

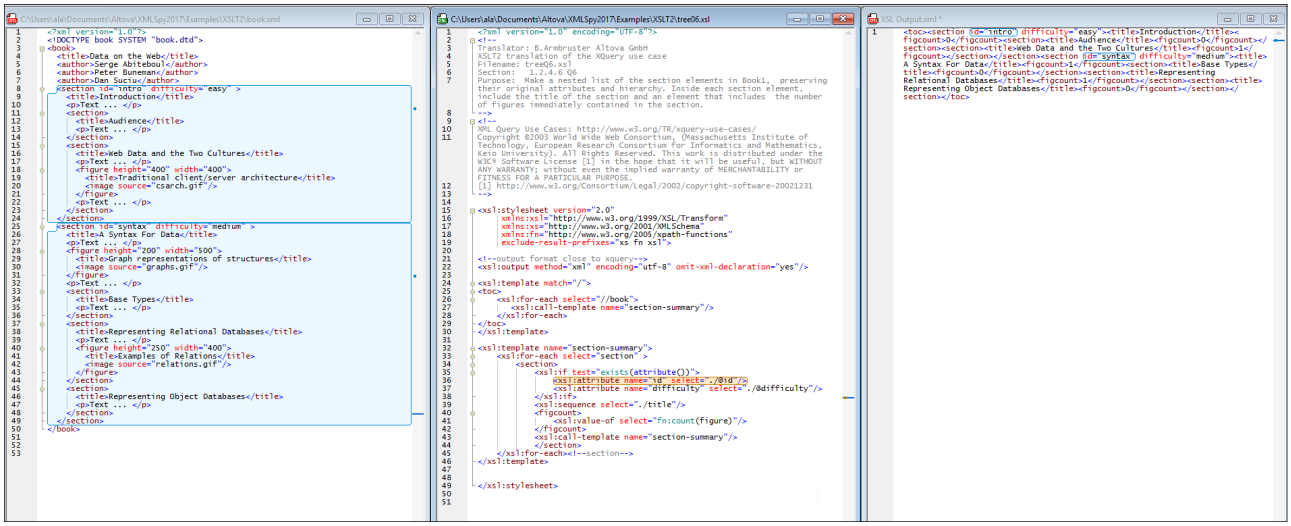
Este comando, también disponible en la barra de herramientas principal, activa la característica de asignación inversa.

Cuando se habilita esta característica (con ayuda de este comando), las transformaciones XSLT y ejecuciones XQuery se llevan a cabo de manera que el documento de salida se pueda relacionar con los documentos XSLT+XML o XQuery+XML originales. Esto significa que, si hace clic en un nodo del documento de salida, la aplicación resaltará la **instrucción XSLT** y los **datos de origen XML** que produjeron dicho nodo (*imagen siguiente*). Esto puede ser muy útil a la hora de comprobar cómo se crean exactamente las diferentes partes del documento de salida. También puede hacer clic en un nodo del documento XSLT/XQuery o XML para ver la parte correspondiente en los otros dos documentos.

Nota: en la vista Texto se abren todos los tipos de documentos de salida, excepto los de tipo HTML. Los documentos HTML de salida se abren en la vista Explorador, pero se pueden ver en la vista Texto también. Si el documento de salida se abre en la vista Explorador, la función de asignación inversa solamente estará disponible si se selecciona texto en el documento de salida y no en el documento XML o XSLT/XQuery.

Nota: la función de asignación inversa no es compatible con transformaciones realizadas desde la vista Authentic 597 ni con transformaciones que se ejecuten como transformaciones de proyecto 1328.

En la imagen siguiente puede ver la asignación inversa de una transformación XSLT. Los tres documentos se presentan en forma de mosaico vertical y en paralelo (`XML+XSLT+Resultado`). El usuario hizo clic en la instrucción XSLT que da lugar al atributo **`section/@id`** del documento de salida. La aplicación resaltan todos los nodos resultantes que genera la instrucción, así como los datos XML de origen que se usaron para generar el nodo. También se puede hacer clic en los nodos del documento de salida o del documento XML para resaltar los nodos correspondientes en los otros dos documentos.



Cuando se hace clic en **Habilitar asignación inversa XSLT/XQuery**, aparece un cuadro de diálogo donde puede indicar si desea ver los **documentos en mosaico** después de la transformación. Si elige esta opción, los tres documentos se presentarán en mosaico vertical y en paralelo (*imagen anterior*).

Debe tener en cuenta estos aspectos de la característica de asignación inversa XSLT/XQuery:

- Solamente se pueden presentar documentos XML que se carguen desde una ubicación del disco local. No se pueden mostrar archivos temporales.
- En algunos casos (p. ej. en ejecuciones XQuery), el documento resultante se crea sin obtener datos de ningún archivo XML de origen. Cuando esto ocurra, no se usa ningún archivo XML en la asignación inversa y, por tanto, no se muestra ningún archivo XML.
- Si se usaron varios archivos XML como origen de datos, el primero que se encuentre durante la operación de transformación o de ejecución será el que se presente en pantalla.
- Las transformaciones/ejecuciones que tienen habilitada la asignación inversa son más lentas y usan más memoria que las que no tienen habilitada esta característica. Esto se debe tener en cuenta en todo momento, pero sobre todo cuando se trabaje con archivos de gran tamaño.
- El menú contextual del documento de resultados (tanto si se presenta en mosaico como si no), contiene comandos que le llevarán hasta los nodos correspondientes en el documento XML (**Ir al nodo de contexto**) y en el documento XSLT/XQuery (**Ir a la instrucción de origen**).

Barra de herramientas de la sesión de asignación inversa

La barra de herramientas Asignación inversa (imagen siguiente) incluye estos dos botones:



- **Resaltar HTML en vista Explorador al pasar el ratón**: si el documento de salida se abre en la vista Explorador, la función de asignación inversa solamente estará disponible si se selecciona contenido en el documento de salida y no en el documento XML o XSLT/XQuery. En la vista Explorador puede seleccionar contenido para la sesión de asignación inversa de dos formas distintas: (i) haciendo clic en el contenido en la vista Explorador o (ii) pasando el puntero del ratón por encima del contenido. Utilice este botón de la barra de herramientas para alternar entre estos dos métodos de selección. La opción del puntero del ratón es práctica si al hacer clic en la vista Explorador se producen cambios en el documento de salida (p. ej. si supone hacer clic en un botón de opción o en un cuadro combinado).
- **Finalizar sesión de asignación inversa**: finaliza la sesión de asignación inversa.

Finalizar la sesión de asignación inversa

Mientras está en curso una sesión de asignación inversa, la barra de herramientas incluye el botón **Finalizar sesión de asignación inversa** (*imagen anterior*). Haga clic en este botón para finalizar la sesión de asignación inversa. Recuerde que la asignación inversa debe utilizarse para diagnosticar problemas solamente y lo mejor es cerrar la sesión antes de realizar cambios en los documentos.

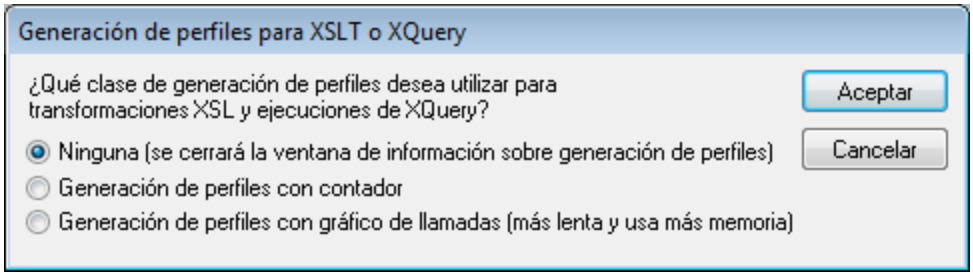
Color de resaltado de texto

Puede definir colores para la asignación inversa activa (contenido controlado por la asignación inversa que está seleccionado en un momento dado) y para la asignación inversa inactiva (contenido controlado por asignación inversa que no está seleccionado). Esto se hace en la sección *Fuentes y colores: Vista Texto* del cuadro de diálogo "Opciones" (seleccionando la opción **Otros** en la lista desplegable). Consulte la descripción de **Herramientas | Opciones | Fuentes y colores**[1603] para obtener más información.

© 2016-2022 Altova GmbH

28.8.7 Generación de perfiles para XSLT o XQuery

El comando **Generación de perfiles para XSLT o XQuery** abre el cuadro de diálogo del mismo nombre. Este cuadro de diálogo sirve para activar el Generador de perfiles 558, una herramienta que analiza la duración de las instrucciones en la transformación XSLT o ejecución de XQuery.



28.8.8 Asignar XSL



El comando **XSL/XQuery | Asignar XSL...** asigna una hoja de estilos XSLT a un documento XML. Al hacer clic en el comando se abre un cuadro de diálogo donde puede especificar qué archivo XSLT desea asignar. También puede seleccionar un archivo a través de un recurso global o una URL (clic en el botón **Examinar** 1262) o seleccionar un archivo de las ventanas que están abiertas en XMLSpy (clic en el botón **Ventana**).

Después de seleccionar el archivo XSLT se inserta una instrucción de procesamiento `xml-stylesheet` en el documento XML:

```
<?xml-stylesheet type="text/xsl" href="C:\workarea\recursion\recursion.xslt"?
   >
```

Nota: puede convertir la ruta de acceso del archivo asignado en relativa marcando la casilla Convertir ruta de acceso en relativa a... del cuadro de diálogo.

28.8.9 Asignar XSL-FO

El comando **XSL/XQuery | Asignar XSL-FO** asigna una hoja de estilos XSLT al documento XML para poder transformarlo en un documento FO. Al hacer clic en este comando aparece un cuadro de diálogo donde puede especificar qué archivo XSL o XSLT desea asignar e inserta la instrucción de procesamiento necesaria en el documento XML.

Puede convertir la ruta de acceso del archivo asignado en relativa marcando la casilla *Convertir ruta de acceso en relativa a...* del cuadro de diálogo.También puede seleccionar un archivo a través de un recurso global o una

URL (clic en el botón **Examinar**[1262]) o seleccionar un archivo de las ventanas que están abiertas en XMLSpy (clic en el botón **Ventana**).

Nota: a un documento XML se le pueden asignar dos archivos XSLT: uno para la transformación XSLT estándar y otro para la transformación XSL-FO.

28.8.10 Asignar archivo XML de muestra



El comando **XSL/XQuery | Asignar archivo XML de muestra** asigna un archivo XML a un documento XSLT. Este comando inserta una instrucción de procesamiento que designa el archivo XML que debe procesar el archivo XSLT cuando se ejecuta el comando **Transformación XSL** en el archivo XSLT:

```
<?altova_samplexml C:\workarea\html2xml\article.xml?>
```

Nota: puede convertir la ruta de acceso del archivo asignado en relativa marcando la casilla *Convertir ruta de acceso en relativa a...* del cuadro de diálogo. También puede seleccionar un archivo a través de un recurso global o una URL (clic en el botón **Examinar**[1262]) o seleccionar un archivo de las ventanas que están abiertas en XMLSpy (clic en el botón **Ventana**).

28.8.11 Ir al archivo XSL



El comando **XSL/XQuery | Ir al archivo XSL** abre el documento XSLT asociado. Si el documento XML activo contiene una instrucción de procesamiento de hoja de estilos (es decir, si tiene asignado un archivo XSLT) como por ejemplo esta

```
<?xml-stylesheet type="text/xsl" href="Company.xsl"?>
```

entonces, al hacer clic en el comando **Ir al archivo XSL** se abre el documento XSLT en XMLSpy.

28.8.12 Ir a la instrucción de origen

Si selecciona un nodo en el documento resultante de una asignación inversa[1401] y hace clic en **Ir a la instrucción de origen**, el cursor del ratón le llevará directamente a la instrucción de origen en el documento XSLT o XQuery que generó el nodo seleccionado en el documento de salida, donde puede, si quiere, editar esa instrucción.

Para saber más sobre asignaciones inversas consulte la descripción del comando de menú **XSL | Habilitar asignación inversa**[1401].

© 2016-2022 Altova GmbH

28.8.13 Ir al nodo de contexto

Si selecciona un nodo en el documento resultante de una asignación inversa[1401] y hace clic en **Ir al nodo de contexto**, el cursor del ratón le llevará al nodo de datos del documento XML que generó el contenido del nodo seleccionado en el documento de salida, donde puede, si quiere, editar ese nodo.

Para saber más sobre asignaciones inversas consulte la descripción del comando de menú **XSL | Habilitar asignación inversa**[1401].

28.8.14 Iniciar el depurador



Alt+F11

El comando **XSL/XQuery | Iniciar el depurador** inicia el procesamiento del documento XSLT/XQuery o continúa procesándolo hasta el final. Si se establecieron puntos de interrupción, el procesamiento se detiene en ellos, uno por uno. Si se establecieron puntos de seguimiento, al llegar al nodo de cierre de la instrucción que tiene el punto de seguimiento aparece en la ventana Seguimiento el resultado de la instrucción. Si la sesión del depurador no se inició todavía, este comando inicia la sesión y se detiene en el primer nodo que deba procesarse. Si la sesión ya está en ejecución y hace clic en este comando, el procesador continúa procesando el documento XSLT/XQuery hasta el final o hasta que encuentre un punto de interrupción.

28.8.15 Detener el depurador



El comando **XSL/XQuery | Detener el depurador** detiene el depurador. Esto no equivale a detener la **sesión** de depuración. Este comando es muy útil si desea editar un documento en medio de una sesión de depuración o si prefiere usar otros archivos dentro de la misma sesión de depuración. Tras detener el depurador, es necesario reiniciarlo para empezar a procesar el documento XSLT/XQuery desde el principio.

28.8.16 Reiniciar el depurador



El comando **XSL/XQuery | Reiniciar el depurador** borra la información de la ventana de resultados y reinicia la sesión de depuración con los archivos que estén activos en ese momento.

28.8.17 Finalizar la sesión de depurador



El comando **XSL/XQuery | Finalizar la sesión de depurador** termina la sesión de depuración y le devuelve a la vista normal de XMLSpy que estaba activa antes de que iniciara la sesión de depuración. En el cuadro de diálogo Configurar depuración 537 hay una opción para dejar abiertos todos los documentos de resultados de la sesión de depuración o cerrarlos al terminar la sesión de depuración.

28.8.18 Paso a paso por instrucciones



F11

El comando **XSL/XQuery | Paso a paso por instrucciones** recorre paso a paso todos los nodos y expresiones XPath de la hoja de estilos. Si el depurador se detuvo, puede usar este comando para reiniciarlo.

28.8.19 Paso a paso para salir



Mayús+F11

El comando **XSL/XQuery | Paso a paso para salir** sale del nodo actual y pasa al siguiente nodo del mismo nivel del nodo primario o pasa al siguiente nodo del nivel inmediatamente superior al nodo primario.

28.8.20 Paso a paso por procedimientos



Ctrl+F11

El comando **XSL/XQuery | Paso a paso por procedimientos** se salta el nodo actual y pasa al siguiente nodo del mismo nivel o al siguiente nodo del nivel inmediatamente superior al del nodo actual. Si el depurador se detuvo, puede usar este comando para reiniciarlo.

© 2016-2022 Altova GmbH

28.8.21 Mostrar nodo en ejecución



El comando **XSL/XQuery | Mostrar nodo en ejecución** muestra/selecciona en el documento XSLT/XQuery el nodo que está en ejecución en ese momento, así como el nodo de contexto correspondiente en el documento XML. Esto puede ser muy útil si, por ejemplo, hizo clic en otras pestañas que muestran una parte del código de la hoja de estilos XSLT o del archivo XML y ahora desea volver exactamente donde estaba.

28.8.22 Insertar o quitar punto de interrupción

 **F9**

El comando **XSL/XQuery | Insertar o quitar punto de interrupción** inserta o quita un punto de interrupción en la posición del cursor. Puede definir puntos de interrupción en línea en nodos de documentos XML, XSLT y XQuery para determinar donde se debe detener el procesamiento. Al establecer un punto de interrupción aparece una línea discontinua roja encima del nodo. No está permitido definir puntos de interrupción en nodos de cierre y se pasan por alto todos los puntos de interrupción definidos en atributos de documentos XSLT. Este comando también está disponible si hace clic con el botón derecho en el nodo del punto de interrupción.

28.8.23 Insertar o quitar punto de seguimiento

 **Mayús+F9**

El comando **XSL/XQuery | Insertar o quitar punto de seguimiento** inserta o quita un punto de seguimiento en la posición del cursor en el documento XSLT/XQuery. Durante la depuración, en la ventana Seguimiento aparece el valor de las instrucciones que tiene un punto de seguimiento cuando se llega al nodo de cierre de la instrucción. Al establecer un punto de seguimiento aparece una línea discontinua azul encima del nodo. No está permitido definir puntos de seguimiento en nodos de cierre. Este comando también está disponible si hace clic con el botón derecho en el nodo del punto de seguimiento.

28.8.24 Habilitar o deshabilitar punto de interrupción

 **Ctrl+F9**

El comando **XSL/XQuery | Habilitar o deshabilitar punto de interrupción** habilita o deshabilita puntos de interrupción ya definidos. La línea discontinua roja que señaliza el punto de interrupción se vuelve gris cuando se deshabilita el punto de interrupción. El depurador no se detiene en los puntos de interrupción que están

deshabilitados. Para deshabilitar/habilitar un punto de interrupción, ponga el cursor en el nombre del nodo y haga clic en el comando **Habilitar o deshabilitar punto de interrupción**. Este comando también está disponible si hace clic con el botón derecho en el nodo del punto de interrupción.

Nota: en la barra de herramientas no hay ningún icono para este comando.

28.8.25 Habilitar o deshabilitar punto de seguimiento



Ctrl+Mayús+F9

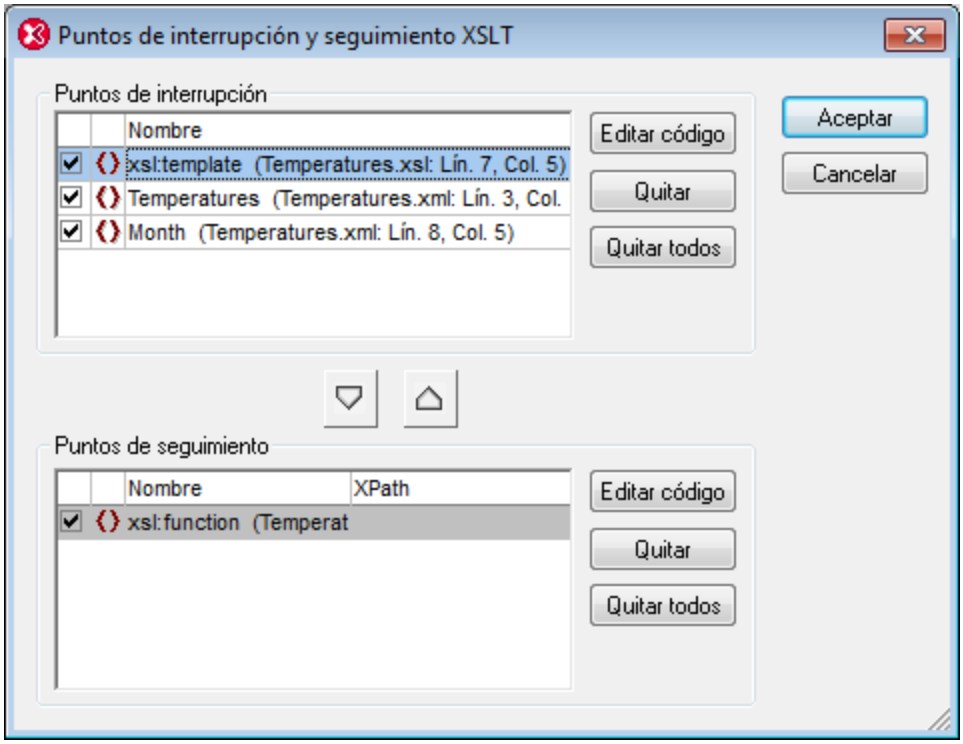
El comando **XSL/XQuery | Habilitar o deshabilitar punto de seguimiento** habilita o deshabilita puntos de seguimiento ya definidos. La línea discontinua azul que señaliza el punto de seguimiento se vuelve gris cuando se deshabilita el punto de seguimiento. La ventana de resultados no muestra información sobre los puntos de seguimiento que están deshabilitados. Para deshabilitar/habilitar un punto de seguimiento, ponga el cursor en el nombre del nodo y haga clic en el comando **Habilitar o deshabilitar punto de seguimiento**. Este comando también está disponible si hace clic con el botón derecho en el nodo del punto de seguimiento.

28.8.26 Puntos de interrupción y seguimiento



El comando **XSL/XQuery | Puntos de interrupción y seguimiento...** abre el cuadro de diálogo "Puntos de interrupción y seguimiento XSLT". Este cuadro de diálogo enumera todos los puntos de interrupción y seguimiento (incluso los que están deshabilitados) que se definieron en todos los archivos de la sesión de depuración actual.

© 2016-2022 Altova GmbH



Las casillas indican si el punto de interrupción o seguimiento está habilitado (si la casilla está marcada) o deshabilitado. Para quitar un punto de interrupción o seguimiento, selecciónelo y pulse el botón **Quitar**. Para quitar todos los puntos de interrupción o seguimiento, pulse el botón **Quitar todos**. El botón **Editar código** cierra este cuadro de diálogo y le lleva al punto de interrupción o seguimiento correspondiente en el archivo.

Con el botón  puede mudar el punto de interrupción seleccionado al panel *Puntos de seguimiento*. Con el botón  puede mudar el punto de seguimiento seleccionado al panel *Puntos de interrupción*.
En la columna **XPath** del panel *Puntos de seguimiento* puede establecer una expresión XPath para cada punto de seguimiento.

28.8.27 Ventanas del depurador

Al pasar el puntero por encima del comando **XSL/XQuery | Ventanas del depurador** aparece un submenú que enumera las ventanas de información del depurador XSLT/XQuery. En este menú puede activar/desactivar (mostrar/ocultar) las ventanas de información en la interfaz del depurador. Este comando se puede usar solamente si está activa una sesión de depuración.

28.8.28 Configurar depuración



El comando **XSL/XQuery | Configurar depuración** abre el cuadro de diálogo Configurar depuración [537], donde puede configurar todas las opciones del depurador. Para más información consulte la sección Depurador XSLT y depurador XQuery [531] del *Manual del usuario*.

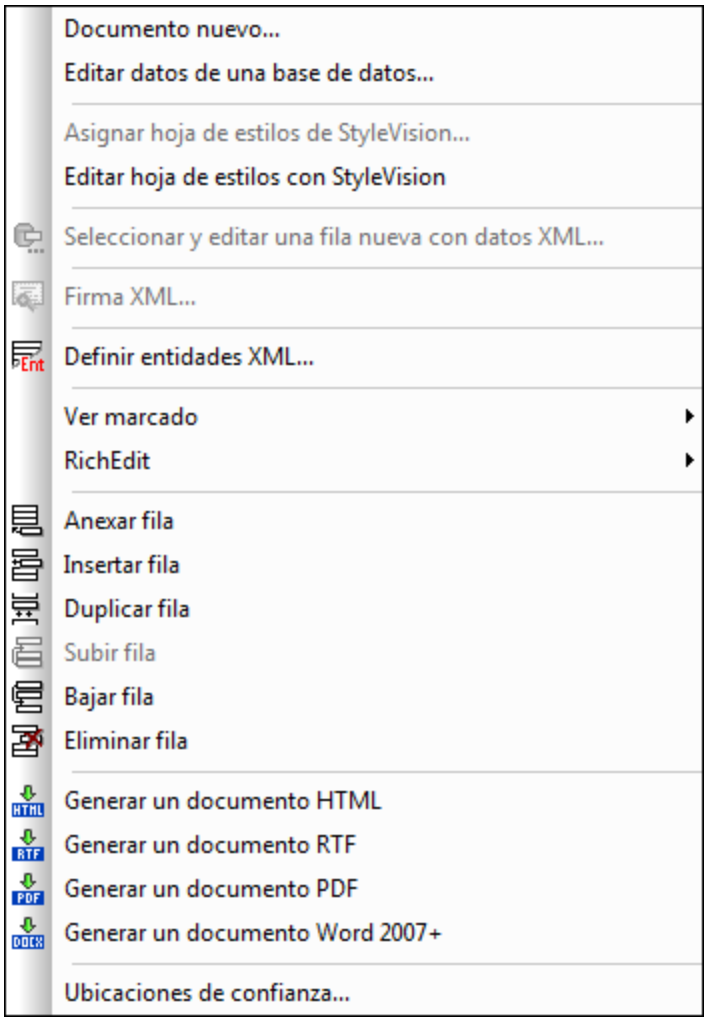
© 2016-2022 Altova GmbH

28.9 Menú Authentic

En la vista Authentic puede editar documentos XML basados en **archivos SPS creados con Altova StyleVision.** Los archivos SPS son hojas de estilos que permiten mostrar el archivo XML de forma gráfica en la vista Authentic. Además de contener información sobre la presentación, el archivo SPS también permite escribir datos en el archivo XML. Estos datos se procesan de forma dinámica usando las funciones de las hojas de estilos XSLT y su resultado se reproduce instantáneamente en la vista Authentic.

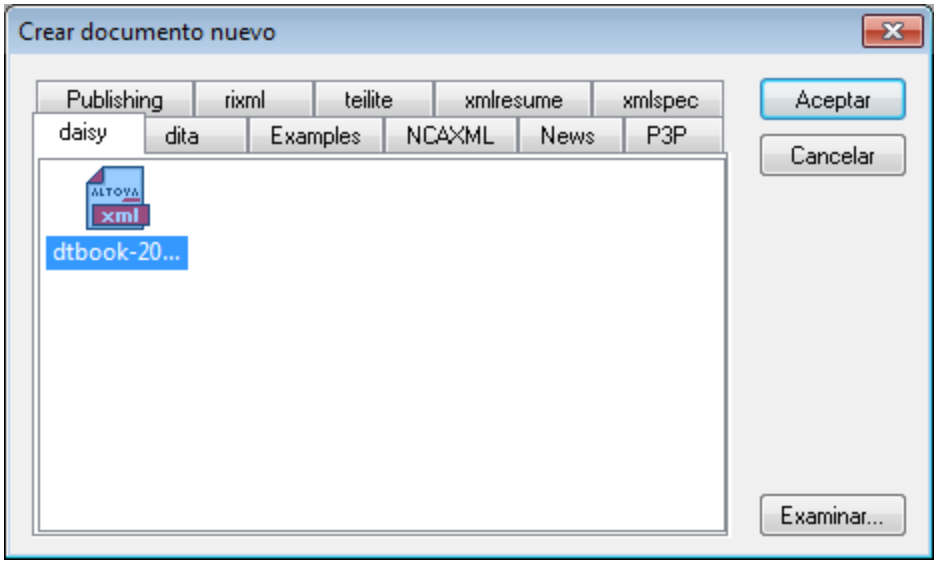
También puede usar archivos SPS para mostrar una vista XML editable de una base de datos. El archivo SPS contiene información para conectarse a la base de datos, mostrar los datos de la base de datos en la vista Authentic y volver a escribir datos en la base de datos.

El menú **Authentic** incluye todos los comandos necesarios para editar documentos XML en la vista Authentic. Para aprender a usar la vista Authentic, consulte este tutorial 599.



28.9.1 Documento nuevo

Este comando sirve para abrir una plantilla de documento XML nuevo en la vista Authentic. La plantilla de documento XML está basada en un archivo SPS (StyleVision Power Stylesheet) y se abre seleccionando el archivo SPS en el cuadro de diálogo "Crear documento nuevo" (*imagen siguiente*). Tras seleccionar el archivo SPS haga clic en **Aceptar**. La plantilla de documento XML definida para el archivo SPS elegido se abre en la vista Authentic.



El cuadro de diálogo "Crear documento nuevo" ofrece varias plantillas de documento XML basadas en esquemas y archivos DTD frecuentes. Si lo prefiere, también puede buscar archivos SPS personalizados que tengan asignado un archivo XML. Los archivos SPS se crean con Altova StyleVision, una herramienta con la que puede diseñar plantillas de documento XML a partir de un esquema XML o un archivo DTD. Tras diseñar el archivo SPS en StyleVision puede asignarle un archivo XML como plantilla XML. Los datos de esta plantilla XML aportan los datos de partida de la plantilla de documento que se abre en la vista Authentic de XMLSpy.

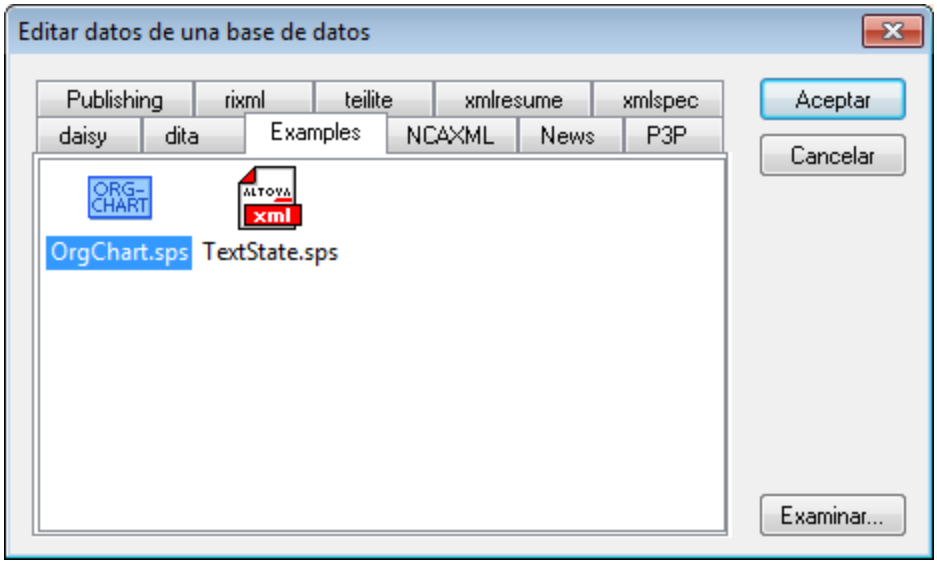
La nueva plantilla de documento XML las propiedades de presentación definidas en el archivo SPS y los datos del archivo XML que seleccionó como plantilla XML. El usuario de la vista Authentic puede editar la plantilla de documento XML en una interfaz gráfica WYSIWYG y guardar los cambios en un archivo XML.

28.9.2 Editar datos de una base de datos

El comando **Authentic** | **Editar datos de una base de datos** sirve para abrir una vista editable de una base de datos (BD) en la vista Authentic. El archivo SPS contiene la información necesaria para conectarse a la BD y sobre cómo presentar la BD y aceptar los cambios realizados en ella en la vista Authentic. El archivo SPS basado en la BD se abre con el comando **Editar datos de una base de datos...**. Al abrirlo se establece una conexión a la BD y sus datos se representan (a través de la lente de XML) en la vista Authentic.

Al hacer clic en el comando **Editar datos de una base de datos...** aparece un cuadro de diálogo del mismo nombre.

© 2016-2022 Altova GmbH



Busque el archivo SPS que desea usar, selecciónelo y haga clic en **Aceptar**. A continuación se establece la conexión con la BD y en la vista Authentic se abre una vista editable de la BD. El diseño de la vista de la BD que aparece en la vista Authentic se define dentro del archivo SPS.

Nota: si con el comando **Editar datos de una base de datos** intenta abrir un archivo SPS que no está basado en una BD o si intenta abrir un archivo SPS basado en una BD que se creó con una versión de StyleVision anterior a StyleVision 2005, entonces recibirá un error.

Nota: los archivos SPS se crean con Altova StyleVision.

28.9.3 Asignar una hoja de estilos de StyleVision

Este comando asigna un archivo SPS a un documento XML para que este documento se pueda ver y editar en la vista Authentic. El archivo SPS y el archivo XML al que se asigna deben estar basados en el mismo esquema XML.

Siga estos pasos para asignar un archivo SPS a un archivo XML:

1. Con el archivo XML activo en la ventana principal, seleccione el comando **Authentic | Asignar una hoja de estilos de StyleVision...**.
2. Aparece un cuadro de diálogo donde puede seleccionar el archivo SPS que desea asignarle al archivo XML.
3. Haga clic en **Aceptar** para insertar la instrucción necesaria en el documento XML. Recuerde que la ruta de acceso al archivo asignado se puede convertir en relativa marcando la casilla *Convertir ruta de acceso en relativa a...*. También puede seleccionar un archivo a través de un recurso global o una URL (clic en el botón **Examinar**[1262]) o seleccionar un archivo de las ventanas que están abiertas en XMLSpy (clic en el botón **Ventana**).

```
<?xml version="1.0" encoding="UTF-8"?>
<?altova_sps HTML-Orgchart.sps?>
```

En el ejemplo anterior, el archivo SPS se llama `HTML_Orgchart.sps` y está en el mismo directorio que el archivo XML.

Nota: las versiones anteriores de los productos de Altova usaban una instrucción de procesamiento con el destino o nombre xmlspysps, por lo que la instrucción de procesamiento tendría este aspecto: <? xmlspysps HTML-Orgchart.sps?>. Estas instrucciones de procesamiento siguen siendo válidas en la versión actual de los productos de Altova.

28.9.4 Editar una hoja de estilos de StyleVision

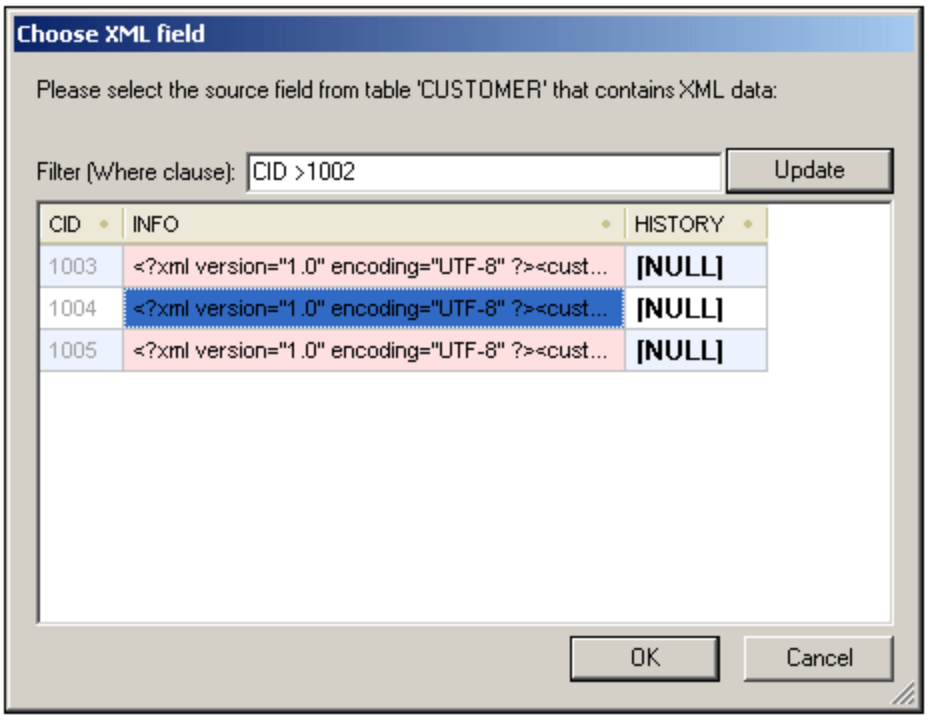
El comando **Authentic | Editar una hoja de estilos de StyleVision** solo está disponible en la Vista Authentic. Es decir, solo si se ha asignado al documento una hoja de estilos de StyleVision. Este comando inicia la aplicación StyleVision para que pueda editar el archivo SPS seleccionado.

28.9.5 Seleccionar y editar una fila nueva con datos XML

El comando **Seleccionar y editar una fila nueva con datos XML** sirve para seleccionar una fila nueva de la tabla pertinente en una BD XML, como IBM DB2. Esta fila aparece en la vista Authentic, donde se puede editar y guardar con los cambios realizados.

Cuando se usa una BD XML como origen de datos XML, los datos XML que se presentan en la vista Authentic corresponden al documento XML que está en una de las celdas de la columna de datos XML. El comando **Seleccionar y editar una fila nueva con datos XML** permite seleccionar un documento XML de otra celda (o fila) de esa columna XML. Al seleccionar este comando aparece el cuadro de diálogo "Elegir campo XML" (*imagen siguiente*), que muestra la tabla en la que está la columna XML.

© 2016-2022 Altova GmbH



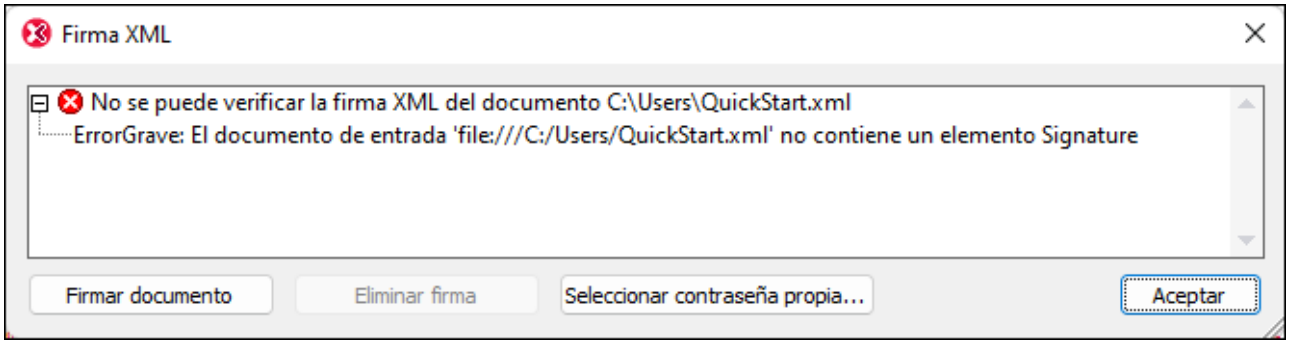
En este cuadro de diálogo puede introducir un filtro para filtrar la tabla. Este filtro debe ser una cláusula `WHERE` SQL (solamente la condición, sin la palabra clave `WHERE`. Por ejemplo `CID>1002`). Haga clic en **Actualizar** para actualizar el cuadro de diálogo. En la imagen anterior puede ver la tabla después de aplicar el filtro. El siguiente paso es seleccionar la celda que contiene el documento XML. Después haga clic en **Aceptar** y el documento XML de la celda (fila) seleccionada se carga en la vista Authentic.

28.9.6 Firma XML

El comando **Firma XML** está disponible en la vista Authentic cuando el archivo SPS asociado tiene firmas XML. El comando **Firma XML** también está disponible en forma de icono en la barra de herramientas de Authentic.

Verificación de la firma y contraseña/certificado propios

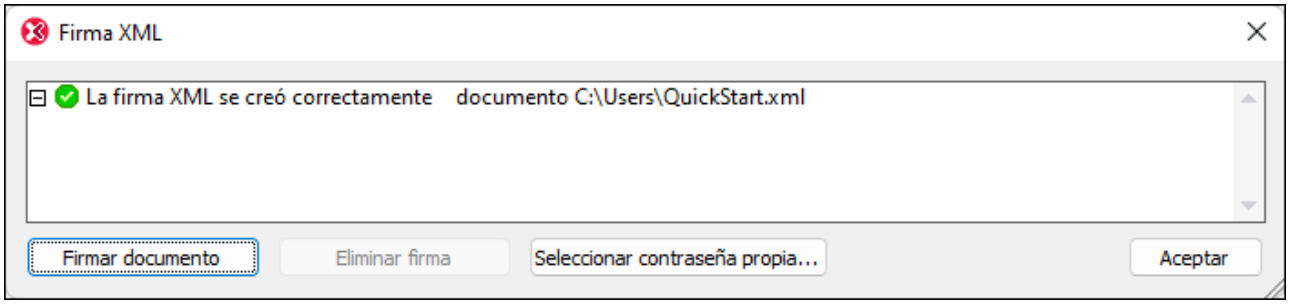
Al hacer clic en el comando **Firma XML** se inicia el proceso de verificación de la firma. Si el documento no tiene una firma, aparece el cuadro de diálogo "Firma XML" con un mensaje a tal efecto (*ver imagen*) y el cuadro de diálogo incluye un botón con el que el usuario de la vista Authentic puede firmar el documento.



Si en este cuadro de diálogo aparece el botón **Seleccionar certificado propio** o **Seleccionar contraseña propia**, significa que se definió la posibilidad de seleccionar un certificado o una contraseña propios en la vista Authentic. (Cuando configura la firma, el diseñador del archivo SPS decide si se puede elegir un certificado o contraseña). La selección del usuario de la vista Authentic se almacena en memoria y solamente es válida para la sesión actual. Si tras seleccionar un certificado o contraseña se cierra el documento o la aplicación, el documento vuelve a su configuración original definida en el SPS.

Verificación e información de autenticación

Si el proceso de verificación se ejecuta en un documento firmado, hay dos posibilidades. Primero, que esté disponible la información de autenticación (en la firma o en el SPS). En este caso se ejecuta el proceso de verificación y el resultado aparece en el cuadro de diálogo "Firma XML" (*imagen siguiente*).

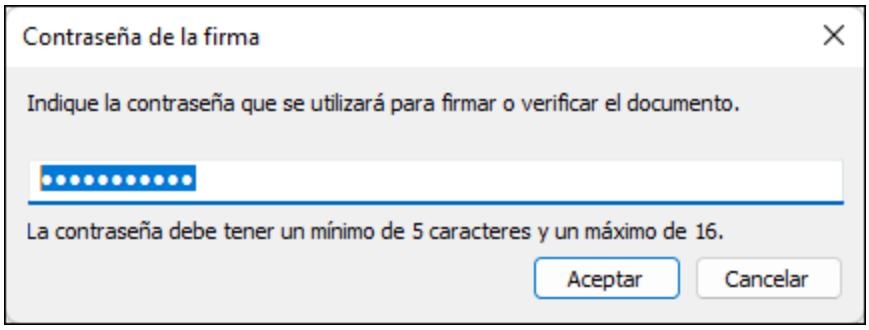


Cuando hablamos de información de autenticación nos referimos a la información de clave del certificado o a la contraseña. El diseñador del archivo SPS especifica si la información de clave del certificado se guarda en la firma cuando se firma el documento XML o, si se trata de una firma con contraseña, si la contraseña se guarda en el SPS. En ambos casos, la información de autenticación está disponible. Por consiguiente, el proceso de verificación se ejecuta directamente, sin solicitar más información al usuario de la vista Authentic.

La segunda posibilidad es que la información de autenticación no esté disponible en la firma (falta la información de clave del certificado) ni en el archivo SPS (falta la contraseña). En este caso, se le pide al

© 2016-2022 Altova GmbH

usuario de la vista Authentic que ofrezca más información: una contraseña (*como en la imagen siguiente*) o la ubicación del certificado.



28.9.7 Definir entidades XML

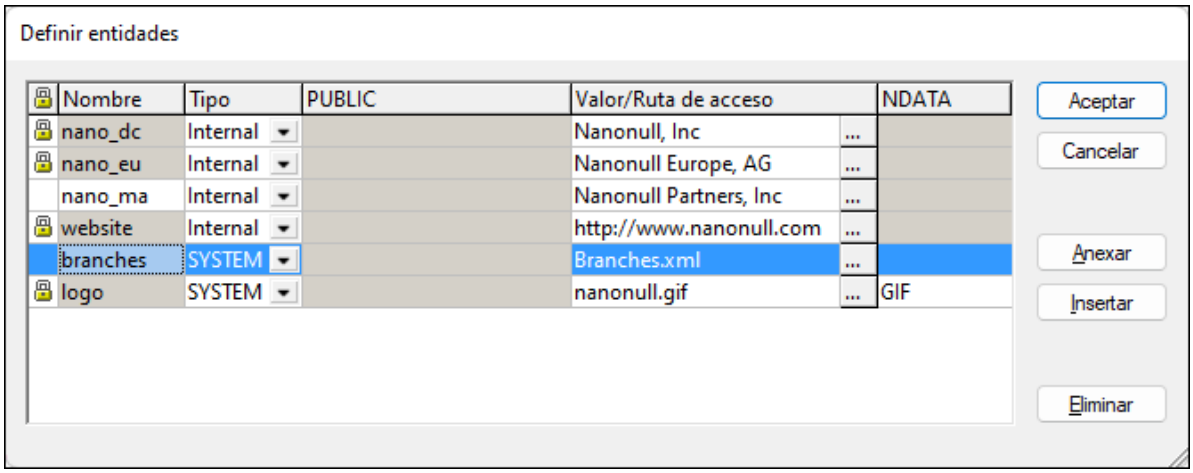
Tanto si su documento está basado en una DTD como si se basa en un esquema XML, puede definir las entidades que se deben usar en la vista Authentic. Una vez definidas, estas entidades aparecen en el ayudante de entrada Entidades y en el submenú **Insertar entidad** del menú contextual. Al hacer doble clic en una entidad del ayudante de entrada Entidades, esta entidad se inserta en el punto de inserción del cursor.

Las entidades son útiles cuando hay una cadena de texto, un fragmento XML o un recurso externo que se va a utilizar en varios puntos del documento. La entidad (básicamente un nombre corto que representa los datos) se define en el cuadro de diálogo "Definir entidades". Después de definirla, puede usar la entidad en varios puntos del documento. Así puede ahorrar tiempo y mantener el documento más fácilmente.

En un documento puede usar dos tipos de entidades: **entidades analizadas**, es decir, datos XML (una cadena de texto o un fragmento de un documento XML) o **entidades sin analizar**, es decir, datos no XML como un archivo binario (por lo general, objetos gráficos, sonoros o multimedia). Cada entidad tiene un nombre y un valor. En el caso de las entidades analizadas, la entidad es un marcador de posición para los datos XML. El valor de la entidad es los datos XML propiamente dichos o un URI que apunta a un archivo XML que contiene los datos XML. En el caso de las entidades sin analizar, el valor de la entidad es un URI que apunta al archivo de datos no XML.

Siga estos pasos para definir una entidad:

1. Haga clic en **Authentic | Definir entidades XML...** y aparece el cuadro de diálogo "Definir entidades".



2. Escriba el nombre de su entidad en el campo **Nombre**. Se trata del nombre que aparecerá en el ayudante de entrada Entidades.
3. En la lista desplegable del campo **Tipo** seleccione el tipo de entidad. Hay tres tipos. Una entidad interna (`Internal`) es una cuyo texto se almacena en el documento XML propiamente dicho. Las opciones **`PUBLIC`** y **`SYSTEM`** indican que el recurso está fuera del XML y que se localizará usando un identificador público o de sistema. Un identificador de sistema es un URI que ofrece la ubicación del recurso. Un identificador público es un identificador que no depende de la ruta de acceso y permite a algunos procesadores identificar el recurso. Si se indica tanto un identificador público como de sistema, el identificador público se resuelve como identificador de sistema y se usa el identificador de sistema.
4. Si selecciona el tipo `PUBLIC`, escriba el identificador público de su recurso en el campo `PUBLIC`. Si selecciona el tipo `Internal` o `SYSTEM`, se deshabilita el campo `PUBLIC`.
5. En el campo **Valor/Ruta de acceso** tiene estas opciones:

   - Si el tipo de entidad es **`Internal`**, escriba la cadena de texto de la entidad. La cadena de texto no puede estar delimitada por comillas. Si escribe comillas, se entenderá que forman parte de la cadena de texto.
   - Si el tipo de entidad es **`SYSTEM`**, escriba el URI del recurso o seleccione un recurso de su red local con el botón **Examinar**. Si el recurso contiene datos analizados, este debe ser un archivo XML (es decir, debe tener la extensión `.xml`). El recurso también puede ser un archivo binario, como un archivo GIF.
   - Si el tipo de entidad es **`PUBLIC`**, debe introducir también un identificador de sistema en este campo.

6. El campo **NDATA** le dice al procesador que la entidad no se debe analizar sino que se debe enviar al procesador adecuado. Por tanto, rellene el campo NDATA solamente si la entidad es una entidad sin analizar.

Características del cuadro de diálogo

En este cuadro de diálogo puede anexar, insertar y eliminar entidades haciendo clic en los botones correspondientes. También puede ordenar las entidades por orden alfabético en base al valor de cualquiera de las columnas. Al hacer clic una vez en el encabezado de la columna, la columna se ordena alfabéticamente en orden ascendente. Si hace clic dos veces en el encabezado, la columna se ordena por orden descendiente. También puede ajustar el tamaño del cuadro de diálogo y el ancho de las columnas.

© 2016-2022 Altova GmbH

Una vez utilizadas en el documento XML, las entidades se bloquean y no se pueden editar en el cuadro de diálogo "Definir entidades". Las entidades bloqueadas se señalizan con un símbolo en forma de candado junto a la primera columna del diálogo. Así se garantiza la validez del documento (el documento no es válido si se hace referencia a una entidad que no se definió).

También recibirá una advertencia si define la misma entidad dos veces.

Limitaciones de las entidades

- Una entidad que está dentro de otra entidad no se resuelve, ni en el cuadro de diálogo, ni en la vista Authentic ni en el resultado XSLT. Además, el carácter & de dichas entidades se presenta con caracteres de escape, es decir, `&`.
- Las entidades externas no se resuelven en la vista Authentic, excepto cuando la entidad es un archivo de imagen y se introduce como valor de un atributo que está definido en el esquema como atributo de tipo `ENTITY` o `ENTITIES`. Este tipo de entidades se resuelven cuando el documento se procesa con un XSLT generado a partir del archivo SPS.

28.9.8 Ver marcado

El comando **Ver marcado** tiene un submenú con opciones para controlar el marcado del documento XML en la vista Authentic. A continuación describimos estas opciones.



El comando **Ocultar marcado** oculta los símbolos de marcado del documento en la vista Authentic.



El comando **Mostrar marcado en tamaño pequeño** muestra símbolos pequeños de marcado en la vista Authentic.



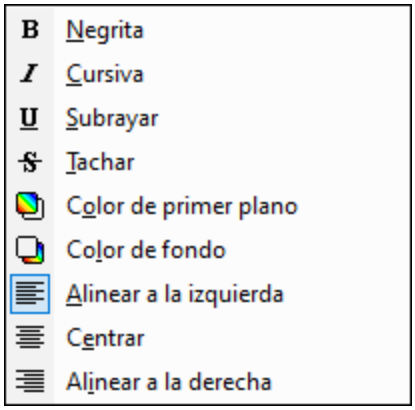
El comando **Mostrar marcado en tamaño grande** muestra símbolos grandes de marcado en la vista Authentic.



El comando **Mostrar marcado mixto** muestra símbolos de marcado de tamaño mixto en la vista Authentic. El diseñador del archivo SPS puede definir diferentes tamaños de marcado para cada elemento y atributo del documento. En el modo de marcado mixto, el usuario de la vista Authentic utiliza este tipo personalizado de marcado.

28.9.9 RichEdit

Al pasar el puntero del ratón por el comando **RichEdit** aparece un submenú con comandos de marcado RichEdit (*imagen siguiente*). Los comandos de este submenú están habilitados solamente en la vista Authentic y solamente cuando se coloca el cursor dentro de un elemento que se creó como componente RichEdit en el diseño SPS.



Las propiedades de estilo del menú **RichEdit** se aplican al texto seleccionado al hacer clic en el uno de los comandos de marcado RichEdit. Además de la fuente y el tamaño de fuente especificado en la barra de herramientas Authentic, el usuario de la vista Authentic puede especificar el peso de la fuente, su estilo, decoración, color, color de fondo y la alineación del texto seleccionado.

28.9.10 Anexar/insertar/duplicar/eliminar fila



El comando **Anexar fila** anexa una fila a la tabla seleccionada en la vista Authentic.



El comando **Insertar fila** inserta una fila en la tabla seleccionada en la vista Authentic.



El comando **Duplicar fila** crea una copia de la fila seleccionada en la tabla de la vista Authentic.



El comando **Eliminar fila** elimina la fila seleccionada en la tabla de la vista Authentic.

© 2016-2022 Altova GmbH

28.9.11 Mover fila, Eliminar fila

En una tabla de la vista Authentic el comando **Subir fila** sube la fila seleccionada una fila más arriba.

En una tabla de la vista Authentic el comando **Bajar fila** baja la fila seleccionada una fila más abajo.

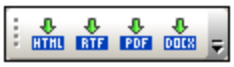
En una tabla de la vista Authentic el comando **Eliminar fila** borra la fila seleccionada.

28.9.12 Generar un documento HTML, RTF, PDF o Word 2007+

Estos cuatro comandos sirven para generar documentos de salida a partir del documento XML de un archivo PXF en la vista Authentic:

- **Generar un documento HTML**
- **Generar un documento RTF**
- **Generar un documento PDF**
- **Generar un documento Word 2007+**
- **Generar un documento de texto**

Estos cinco comandos también están disponibles en la barra de herramientas "Formulario XML Portátil" (*imagen siguiente*).

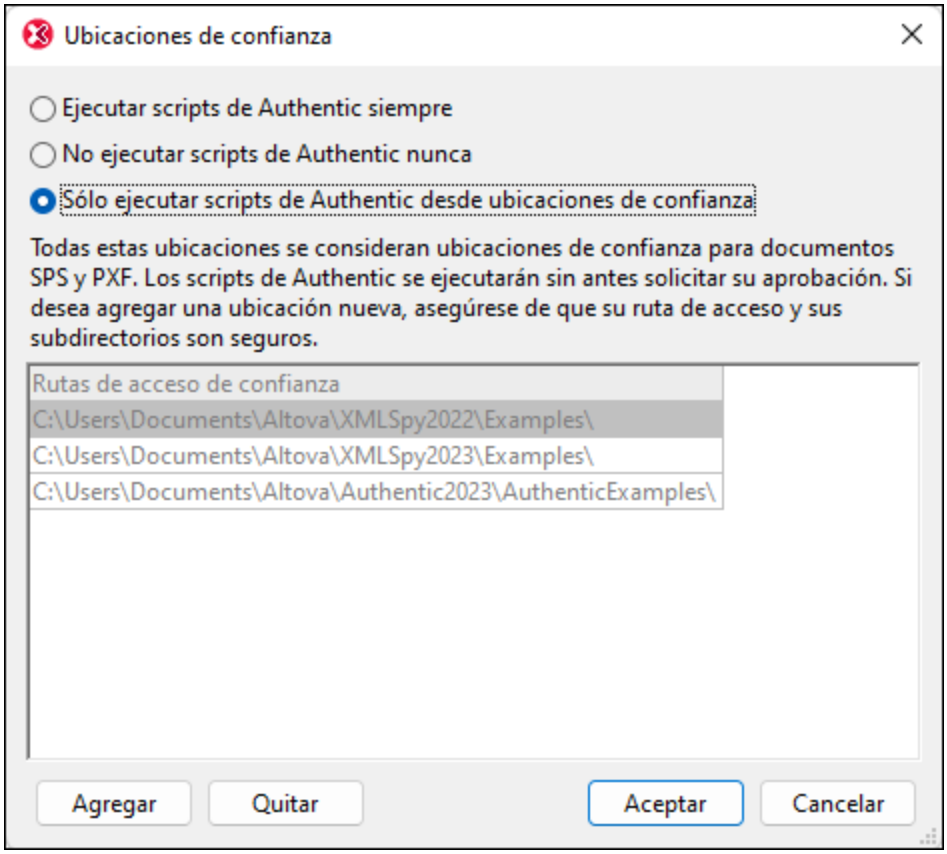


Al hacer clic en estos comandos o botones se genera un documento HTML, RTF, PDF o DocX, dependiendo del comando seleccionado.

Estos comandos se habilitan solamente si el archivo abierto en la vista Authentic es un archivo PXF. Además, si el archivo PXF contiene una hoja de estilos para un formato de salida concreto, solamente se habilita el comando correspondiente. Por ejemplo, si el archivo PXF contiene hojas de estilos XSLT para HTML y RTF, entonces solamente se habilitan los comandos para generar documentos HTML y RTF, mientras que los comandos para PDF y DocX se deshabilitan.

28.9.13 Ubicaciones de confianza

El comando **Ubicaciones de confianza** abre el cuadro de diálogo del mismo nombre (*imagen siguiente*). En este cuadro de diálogo puede configurar opciones de seguridad para los scripts del archivo SPS. Cuando se abre en la vista Authentic un archivo XML basado en un SPS que contiene scripts, los scripts se ejecutan o no dependiendo de las opciones elegidas en este cuadro de diálogo.



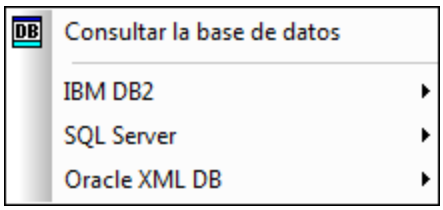
Hay tres posibilidades:

- Los scripts de Authentic se ejecutan siempre
- Los scripts de Authentic no se ejecutan nunca
- Se ejecutan solamente los scripts de Authentic que están en ubicaciones de confianza. La lista de ubicaciones (carpetas) de confianza aparece en el panel inferior del cuadro de diálogo. Use el botón **Agregar** para buscar una carpeta y añadirla a la lista. Para quitar una carpeta de la lista, selecciónela y después haga clic en **Quitar**. Para confirmar haga clic en **Aceptar**.

© 2016-2022 Altova GmbH

28.10 Menú Base de datos

El menú **Base de datos** (*imagen siguiente*) contiene todos los comandos necesarios para realizar operaciones de base de datos (BD). Los comandos de sus submenús se describen en los apartados y subapartados de esta sección.

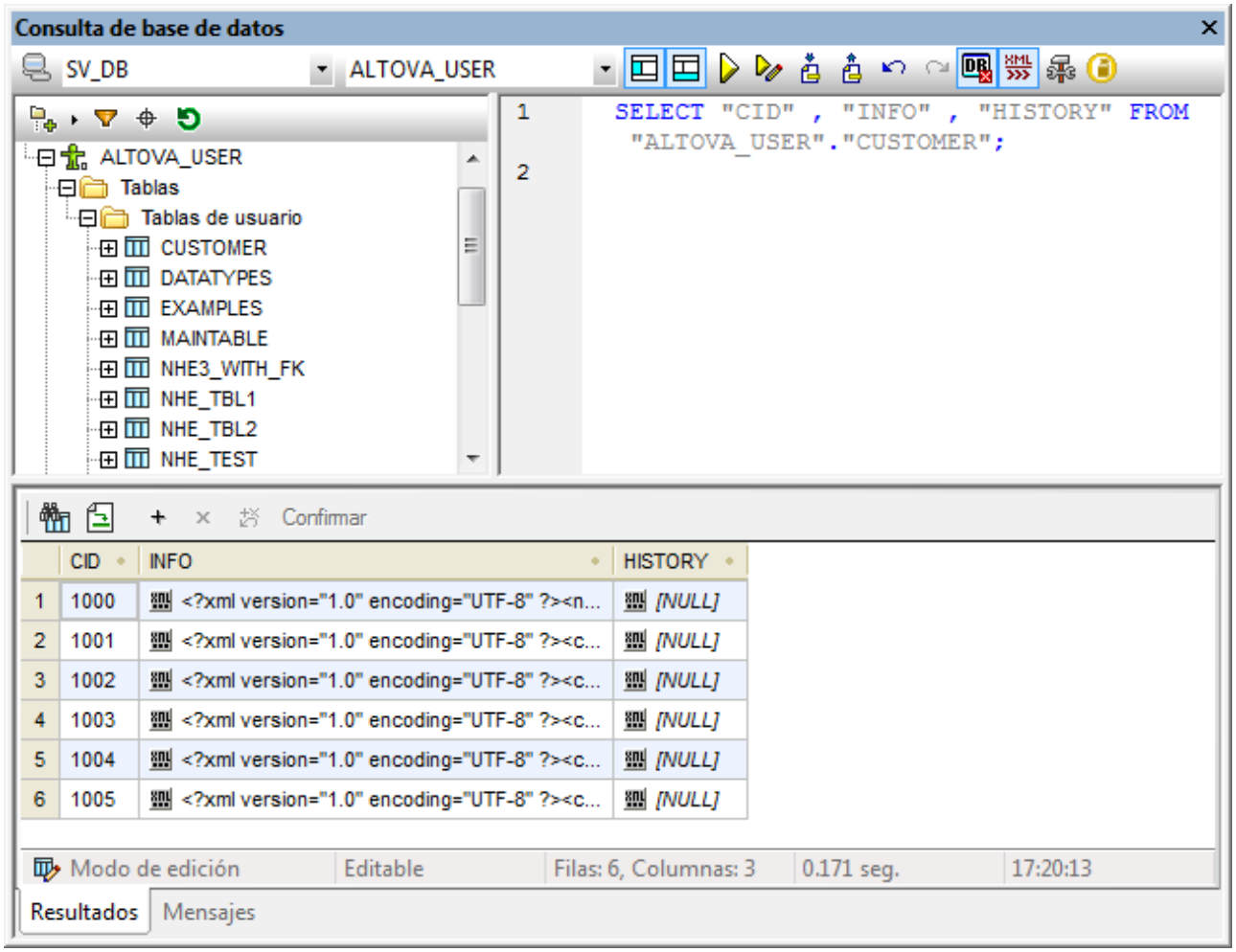


- Consultar la base de datos 1423, que sirve para consultar las bases de datos compatibles.
- IBM DB2 1439, que incluye comandos para trabajar con funciones propias de IBM DB2.
- SQL Server 1445, que contiene comandos para gestionar bases de datos SQL Server.
- Oracle XML DB 1447, que incluye comandos para trabajar con bases de datos de Oracle.

Todas las operaciones descritas en esta sección exigen conectarse a una base de datos. En la sección Conectarse a una base de datos 952 se explica en detalle cómo establecer conexiones con bases de datos.

28.10.1 Consultar la base de datos

El comando **Consultar la base de datos** abre la ventana "Consulta de base de datos" (*imagen siguiente*). Esta ventana se puede abrir o cerrar con el comando **Base de datos | Consultar la base de datos** o haciendo clic en el icono .



Información general sobre la ventana "Consulta de base de datos"

La ventana "Consulta de base de datos" se divide en tres paneles:

- El panel **Explorador**[1428] situado en la parte superior izquierda, que muestra información sobre la conexión y las tablas de la BD.
- El panel de **Consulta**[1432] situado en la parte superior derecha, donde se introduce la consulta propiamente dicha.
- El panel **Resultados/Mensajes**[1436] situado en la parte inferior y formado por dos pestañas. En la pestaña *Resultados* se muestran los resultados de las consultas, en lo que se denomina *cuadrícula de resultados*. En la pestaña *Mensajes* se muestran mensajes sobre la ejecución de la consulta, como advertencias y errores.

En la parte superior de la ventana "Consulta de base de datos" hay una barra de herramientas. Los iconos de esta barra de herramientas se describen en el apartado Panel de consulta: descripción y características[1432], pero es importante conocer cuanto antes estos dos iconos:

© 2016-2022 Altova GmbH

 Este icono activa/desactiva (muestra/oculta) el panel Explorador.

 Este icono activa/desactiva (muestra/oculta) el panel Resultados/Mensajes.

Información general sobre el mecanismo de consulta de la base de datos

En los subapartados siguientes se describe en detalle el mecanismo de consulta de la BD, pero la idea general es esta:

1. Se establece una conexión a la BD[1425] desde la ventana "Consulta de la base de datos". Algunas de las BD compatibles son MS Access 2000 y 2003, Microsoft SQL Server, Oracle, MySQL, Sybase y IBM DB2.
2. La BD conectada o partes de la BD aparecen en el panel Explorador[1428], cuya vista se puede configurar dependiendo de lo que necesite.
3. En el panel de Consulta[1435] se introduce una consulta[1435] en la sintaxis de la BD que se está consultando y se ejecuta la consulta.
4. Los resultados de la consulta[1436] se pueden ver y filtrar, editar y guardar otra vez en la BD.

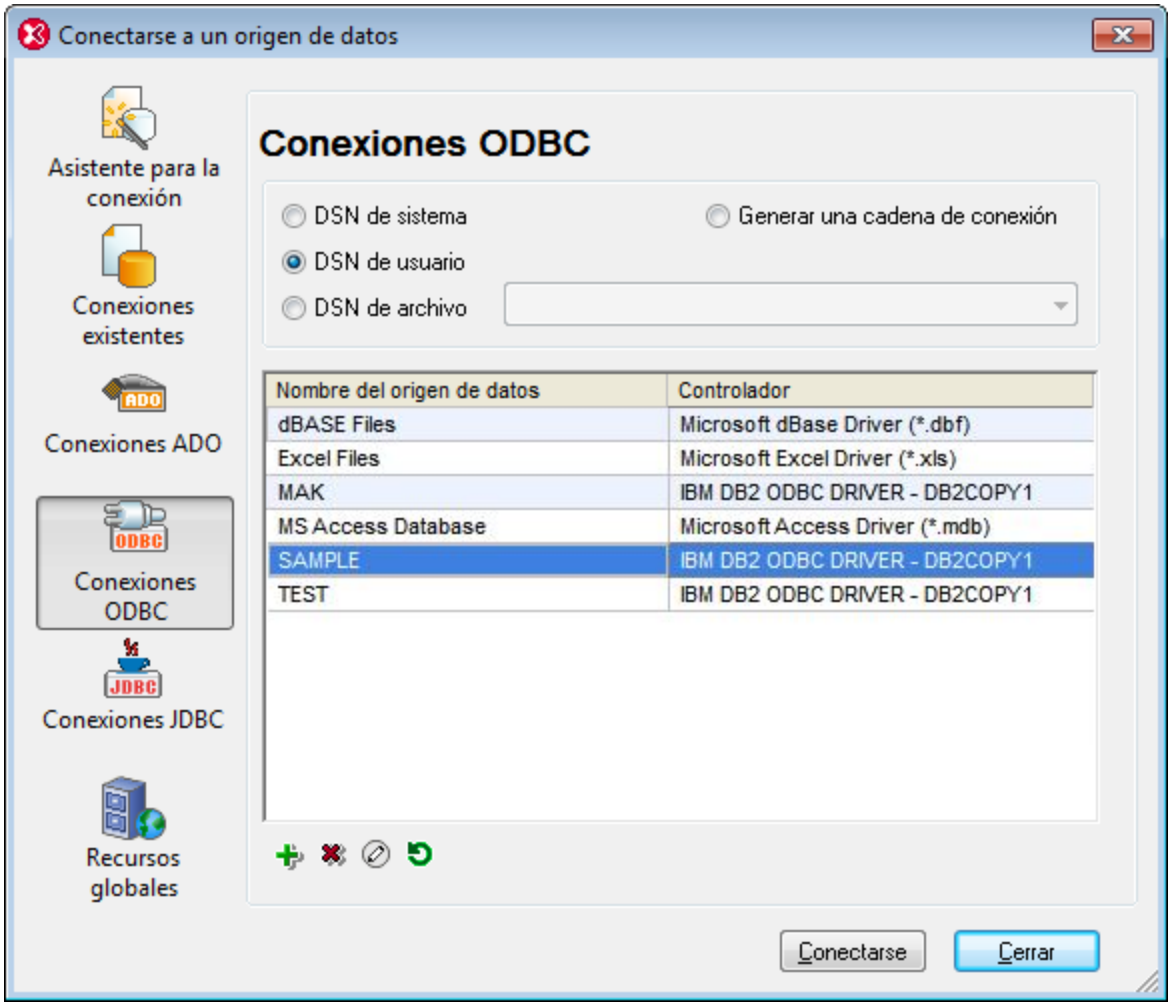
28.10.1.1 Orígenes de datos

Para poder consultar una base de datos es necesario:

- conectarse a la base de datos y
- seleccionar el origen de datos y el objeto raíz que desea utilizar.

Conectarse a una base de datos

Al hacer clic en el comando **Consultar la base de datos** por primera vez en la sesión de trabajo (o cuando no existe una conexión a una BD), aparece el cuadro de diálogo "Conectarse a un origen de datos" (*imagen siguiente*) desde donde puede establecer la conexión a la BD. Más tarde podrá crear otras conexiones a BD con el icono **Conexión rápida**  de la ventana "Consulta de base de datos".



En la sección Conectarse a un origen de datos 952 se describe en detalle cómo establecer la conexión con una base de datos desde este cuadro de diálogo.

Las bases de datos compatibles y sus objetos raíz aparecen a continuación. Altova procura ofrecer compatibilidad con otras bases de datos, pero sólo se garantiza una conexión y un procesamiento correctos con las bases de datos de la lista. Si usa la versión de 64 bits de XMLSpy, compruebe que tiene acceso a los controladores de BD de 64 bits de la BD a la que quiere conectarse.

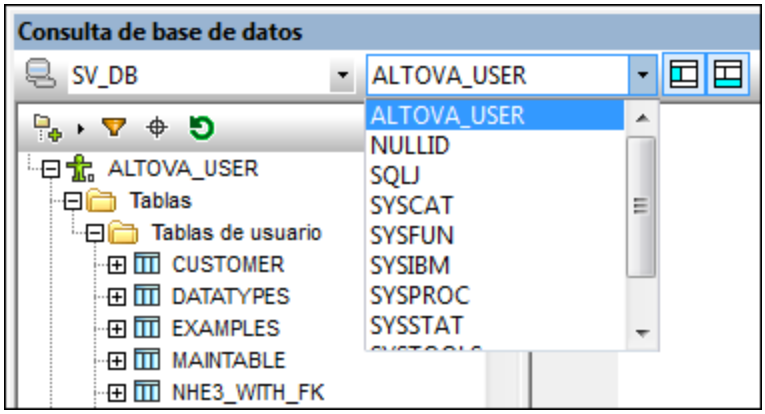
Base de datos	Observaciones
Firebird 2.x, 3.x	
IBM DB2 8.x, 9.x, 10.x, 11.x	
IBM Db2 for i 6.x, 7.4	Los archivos lógicos son compatibles y se muestran en vistas.
IBM Informix 11.70 y superior	
MariaDB 10 y superior	

© 2016-2022 Altova GmbH

Microsoft Access 2003 y superior	En el momento de escribir esta documentación (principios de septiembre de 2019) no hay ningún Microsoft Access Runtime disponible para Access 2019. Solo puede conectarse a la BD de Access 2019 con productos de Altova si tiene instalado Microsoft Access 2016 Runtime y solamente si la BD no usa el tipo de datos "Large Number" (número grande).
Microsoft Azure SQL Database	SQL Server 2016 codebase
Microsoft SQL Server 2005 y superior Microsoft SQL Server para Linux	
MySQL 5 y superior	
Oracle 9i y superior	
PostgreSQL 8 y superior	Son compatibles todas las conexiones PostgreSQL, tanto nativas como basadas en controladores, a través de interfaces como ODBC o JDBC. Las conexiones nativas no necesitan controladores.
Progress OpenEdge 11.6	
SQLite 3.x	Las conexiones SQLite son conexiones nativas y directas compatibles con el archivo de base de datos de SQLite. No se precisan controladores separados. En la vista Authentic, los datos que provienen de una base de datos SQLite no se pueden editar. Cuando intente guardar los datos de SQLite desde la vista Authentic, un cuadro de mensaje le informará de esta limitación conocida.
Sybase ASE 15, 16	
Teradata 16	

Seleccionar el origen de datos

Las conexiones existentes y los objetos raíz de cada conexión se enumeran en dos cuadros combinados de la barra de herramientas de la ventana "Consulta de base de datos" (*imagen siguiente*).



En el ejemplo de la imagen anterior, seleccionamos la BD llamada `SV_DB`. De los objetos raíz disponibles en esta BD, seleccionamos objeto raíz `ALTOVA_USER`. Observe que la BD y el objeto raíz se muestran en el panel Explorador.

28.10.1.2 Panel Explorador: visualizar los objetos de la BD

El panel Explorador ofrece una vista general de los objetos de la BD seleccionada. Esta vista incluye información sobre restricciones de la BD, por ejemplo, si alguna columna es una clave primaria o foránea. En las bases de datos IBM DB2 versión 9, el Explorador también muestra una carpeta con todos los esquemas XML registrados.

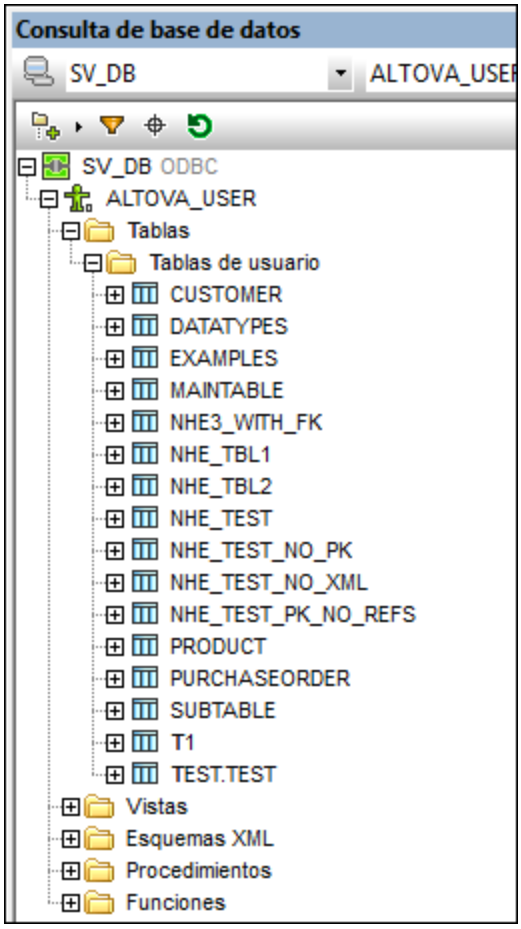
En este apartado describimos:

- Cómo organizar[1428] la información en el panel Explorador.
- Cómo filtrar[1430] los objetos de la base de datos.
- Cómo buscar[1431] objetos de la base de datos.

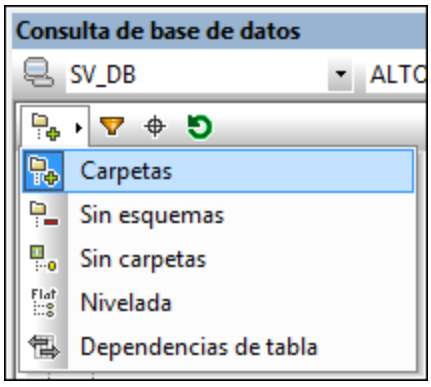
Cómo organizar la información en el panel Explorador

El Explorador se organiza de manera predeterminada por carpetas, mostrando los objetos de la BD por orden jerárquico. Al hacer clic con el botón derecho en un objeto aparece un menú contextual. Las opciones del menú contextual dependen del tipo de objeto seleccionado.

© 2016-2022 Altova GmbH



Para cambiar la organización del panel Explorador, haga clic en el icono **Disposición** de la barra de herramientas y en la lista desplegable seleccione cómo desea organizar los objetos (*imagen siguiente*). Observe que el icono cambia dependiendo de la opción seleccionada.



Las opciones disponibles son:

- *Carpetas:* organiza los objetos de la BD por carpetas, en base al tipo de objeto, en forma de árbol jerárquico. Esta es la disposición predeterminada.

- *Sin esquemas:* similar a la disposición en carpetas, pero ya no hay carpetas para los esquemas de BD. Por tanto, las tablas no se organizan por esquemas de BD.
- *Sin carpetas:* muestra todos los objetos de la BD en orden jerárquico, pero sin usar carpetas.
- Nivelada: divide los objetos de la BD por tipo en el primer nivel jerárquico. Por ejemplo, en lugar de mostrar una tabla y dentro de la tabla sus columnas, el Explorador muestra todas las columnas en una carpeta *Columnas*.
- *Dependencias de tabla:* categoriza las tablas en base a su relación con otras tablas. Las categorías son: tablas con claves foráneas, tablas a las que hacen referencia claves foráneas y tablas que no tienen relación con otras tablas.

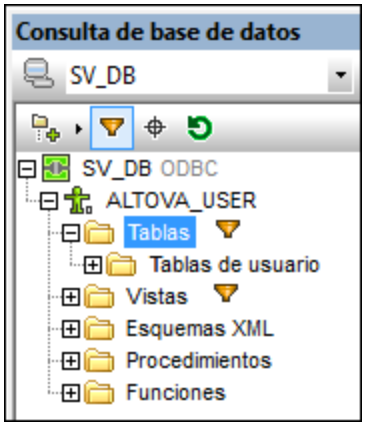
Para ordenar las tablas por tablas de usuario y del sistema, elija la opción **Carpetas**, **Sin esquemas** o **Nivelada**, haga clic con el botón derecho en la carpeta *Tablas* y seleccione **Ordenar por tablas de usuario y del sistema**. Las tablas se ordenan por orden alfabético en las carpetas *Tablas de usuario* y *Tablas del sistema*.

Cómo filtrar los objetos de la base de datos

En el panel Explorador (con cualquiera de las opciones de disposición, excepto *Sin carpetas* y *Dependencias de tabla*) puede filtrar los esquemas, las tablas y las vistas según su nombre o parte de su nombre. Los objetos se filtran a medida que teclea y el filtro distingue entre mayúsculas y minúsculas por defecto.

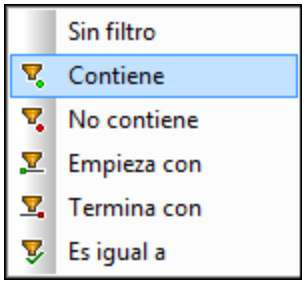
Siga estos pasos para filtrar los objetos en el explorador:

1. Haga clic en el icono **Filtrar contenido de carpeta** de la barra de herramientas del panel Explorador. Junto a las carpetas *Tablas* y *Vistas* aparecen iconos en forma de embudo (*imagen siguiente*).

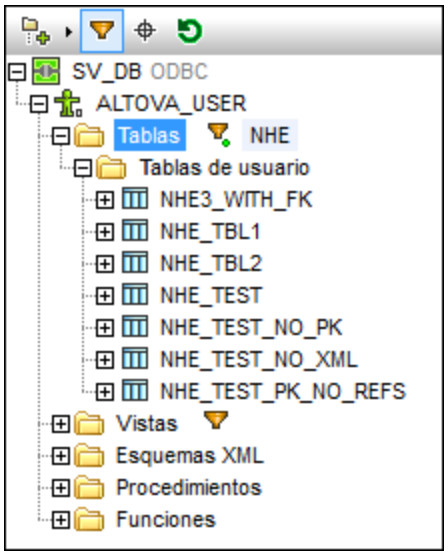


2. Haga clic en el icono en forma de embudo de la carpeta que desea filtrar y seleccione una opción del menú emergente, por ejemplo, *Contiene*.

© 2016-2022 Altova GmbH



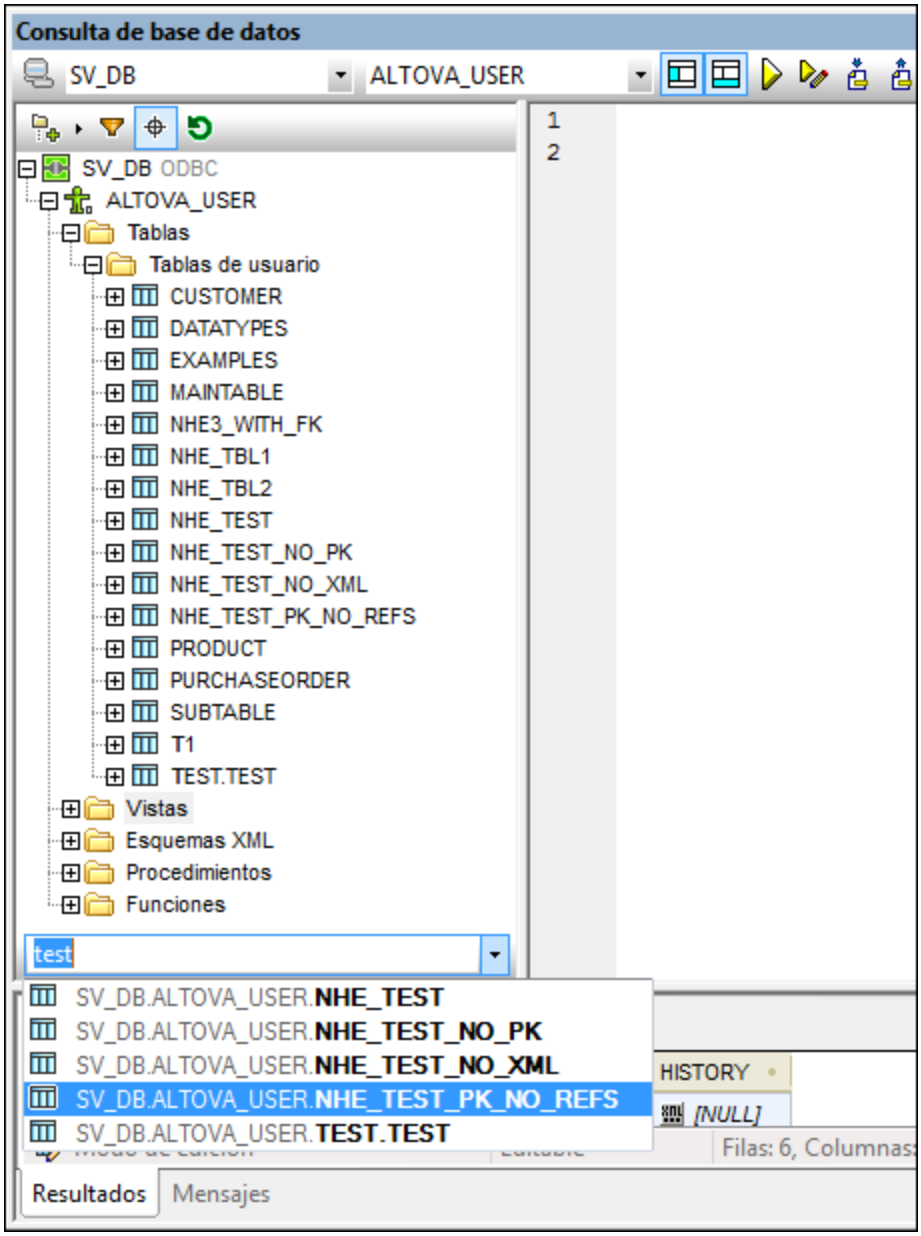
3. En el campo de entrada que aparece, escriba una cadena de texto (en el ejemplo de la imagen siguiente usamos el texto `NHE` para filtrar los objetos de la carpeta *Tablas*). El filtro se aplica en tiempo real mientras se teclea.



Cómo buscar objetos de la tabla

Para buscar un componente concreto de la tabla por su nombre, puede usar el localizador de objetos del panel Explorador. Así funciona esta característica:

1. En la barra de herramientas del panel Explorador, haga clic en el icono **Localizador de objetos**. Aparece una lista desplegable en la parte inferior del Explorador.
2. Escriba la cadena de búsqueda en el campo de entrada de esta lista, por ejemplo, `test` (*imagen siguiente*). Si hace clic en la flecha desplegable, aparecen todos los objetos que contienen la cadena de búsqueda.



3. Haga clic en un objeto de la lista para verlo en el Explorador.

28.10.1.3 Panel de consulta: descripción y características

El panel de consulta es un editor SQL inteligente donde puede escribir consultas para la BD seleccionada. Tras escribir la consulta, haga clic en el comando **Ejecutar** de la ventaja "Consulta de base de datos". La consulta se ejecuta y su resultado y los mensajes resultantes aparecen en el panel Resultados/Mensajes[1436]. En el apartado siguiente, Panel de consulta: trabajar con las consultas[1435], explicamos cómo se trabaja con las consultas. En este apartado nos centramos en las características principales del panel de consulta:

© 2016-2022 Altova GmbH

- Los iconos del editor SQL de la barra de herramientas de la ventana "Consulta de base de datos"
- Las opciones del editor SQL
- La función de finalización automática de instrucciones SQL
- La definición de regiones en los scripts SQL
- La inserción de comentarios en los scripts SQL
- El uso de los marcadores

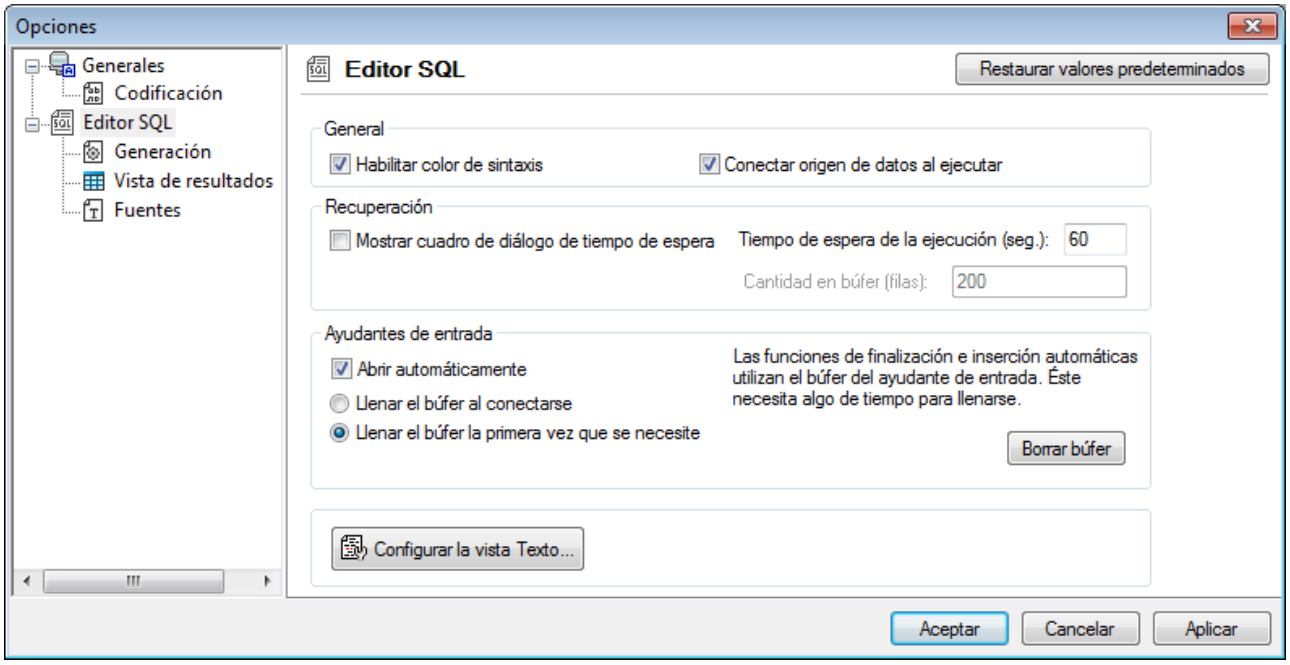
Iconos del editor SQL en la barra de herramientas de la ventana "Consulta de base de datos"

Estos iconos de la barra de herramientas de la ventana "Consulta de base de datos" se utilizan para trabajar con el editor SQL:

	**Ejecutar**	Ejecuta la instrucción SQL seleccionada. Si el script contiene varias instrucciones y no se selecciona ninguna, se ejecutan todas las instrucciones
	**Ejecutar para editar datos**	Igual que el comando **Ejecutar**, pero los resultados se pueden editar (en la pestaña Resultados)
	**Importar archivo SQL**	Abre un archivo SQL en el editor SQL
	**Exportar archivo SQL**	Guarda las consultas SQL en un archivo SQL
	**Deshacer**	Se deshacen los cambios realizados en el editor SQL (sin límite)
	**Rehacer**	Se rehacen los cambios deshechos en el editor SQL (sin límite)
	**Ocultar ventana de consulta de BD al abrir documentos XML**	Determina si la ventana "Consulta de base de datos" se oculta al abrir un documento XML
	**Confirmar cambios automáticamente al guardar el documento XML**	Cuando se guarda un documento XML en XMLSpy, los cambios se confirman en la BD si se activa esta opción. De lo contrario, es necesario confirmar los cambios en el panel Resultados.
	**Opciones**	Abre el cuadro de diálogo "Opciones" del editor SQL
	**Abrir el script SQL en DatabaseSpy**	Abre el script SQL en Altova DatabaseSpy

Opciones

Al hacer clic en el icono **Opciones** de la barra de herramientas de la ventana "Consulta de base de datos", aparece el cuadro de diálogo "Opciones" (*imagen siguiente*). En el panel izquierdo del diálogo puede elegir la página de opciones y editar las opciones en el panel derecho. Para restaurar las opciones predeterminadas de una página, pulse el botón **Restaurar valores predeterminados**.



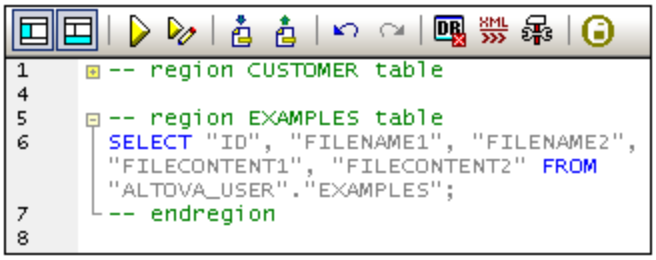
Estas son las principales opciones de configuración:

- ***Generales* | *Codificación:*** opciones para configurar la codificación de archivos SQL nuevos, archivos SQL existentes cuya codificación no se puede detectar y para establecer la marca BOM. (Si la codificación de los archivos SQL existentes se puede detectar, los archivos se abren y se guardan sin cambiar la codificación.)
- ***Editor SQL:*** opciones para habilitar/deshabilitar el color de la sintaxis y la conexión al origen de datos al ejecutar la consulta. Puede definir un tiempo de espera de la ejecución y si aparece un mensaje cuando se sobrepasa el tiempo de espera. El grupo de opciones *Ayudantes de entrada* afecta a los ayudantes de entrada que ofrece la función de finalización automática. Cuando el usuario teclea en una instrucción SQL, el editor muestra una lista de sugerencias de finalización automática contextuales. Puede elegir si estas sugerencias aparecen automáticamente o si prefiere invocarlas a mano. Si desactiva la presentación automática de sugerencias, podrá invocarlas a mano con **Ctrl+Barra espaciadora**. El búfer para la información del ayudante de entrada se puede llenar al conectarse al origen de datos o cuando se necesite por primera vez. El botón **Configurar la vista Texto** 1489 abre el cuadro de diálogo del mismo nombre.
- ***Editor SQL* | *Generación:*** la aplicación genera instrucciones SQL cuando se arrastran objetos desde el panel Explorador hasta al panel de consulta. En esta página puede configurar cómo se generan estas instrucciones SQL. Puede seleccionar opciones de configuración para cada tipo de BD. Si marca la casilla *Aplicar a todas las bases de datos*, las opciones seleccionadas en ese momento se definen para todos los tipos de BD. Hay una opción para anexar un punto y coma al final de cada instrucción y para envolver los identificadores con caracteres de escape. **Nota:** los datos de bases de datos Oracle y IBM iSeries y IBM DB2 usando una conexión JDBC solamente se pueden editar si esta casilla está desactivada.
- ***Editor SQL* | *Vista de resultados:*** opciones para configurar la pestaña *Resultados*.
- ***Editor SQL* | *Fuentes:*** opciones para configurar el estilo de fuente del texto del editor de texto y de la pestaña Resultados.

© 2016-2022 Altova GmbH

Definición de regiones en los scripts SQL

Una región es una parte de un script SQL que se marca y se declara como unidad. Las regiones se pueden contraer y expandir para ocultar o mostrar partes del script y también se pueden anidar dentro de otras regiones. Las regiones se delimitan con los comentarios `--region` y `--endregion`. También puede asignar un nombre a cada región. Este nombre se introduce después del delimitador inicial`-- region` (*imagen siguiente*).



Para insertar una región, seleccione la instrucción o las instrucciones que desea convertir en región, haga clic con el botón derecho y seleccione **Insertar región**. Si lo desea, añada un nombre para la región. Para quitar una región, elimine los dos delimitadores `--region` y `--endregion`.

Inserción de comentarios en los scripts SQL

Puede convertir el texto de scripts SQL en comentarios. El texto de los comentarios se ignora cuando se ejecuta la instrucción.

- Para convertir un bloque de texto en comentario, selecciónelo, haga clic con el botón derecho y seleccione **Convertir el bloque en comentario/quitar comentario**.
- Para convertir una línea de texto en comentario, ponga el cursor en el punto de la línea donde debe comenzar el comentario, haga clic con el botón derecho y seleccione **Convertir la línea en comentario/quitar comentario**. Para quitar el comentario, márquelo, haga clic con el botón derecho y seleccione **Convertir la línea en comentario/quitar comentario**.

Uso de los marcadores

Puede insertar marcadores y navegar por el documento de marcador en marcador. Para insertar un marcador, ponga el cursor en la línea en la que desea insertar el marcador, haga clic con el botón derecho y seleccione **Insertar o quitar marcador**. Para ir al marcador siguiente o anterior, haga clic con el botón derecho y seleccione **Ir al siguiente marcador** o **Ir al marcador anterior** respectivamente. Para quitar un marcador, ponga el cursor en la línea donde está el marcador que desea eliminar, haga clic con el botón derecho y seleccione **Insertar o quitar marcador**. Para quitar todos los marcadores, haga clic con el botón derecho y seleccione **Quitar todos los marcadores**.

28.10.1.4 Panel de consulta: trabajar con las consultas

Tras conectarse a la base de datos, puede introducir un script SQL en el editor SQL y ejecutarlo. En este apartado explicamos:

- cómo introducir un script SQL en el editor SQL y
- cómo ejecutar el script en la ventana "Consulta de base de datos"

En este apartado se hace referencia a estos iconos:

	**Ejecutar**	Ejecuta la instrucción SQL seleccionada. Si el script contiene varias instrucciones y no se selecciona ninguna, se ejecutan todas las instrucciones
	**Ejecutar para editar datos**	Igual que el comando **Ejecutar**, pero los resultados se pueden editar (en la pestaña Resultados)
	**Importar archivo SQL**	Abre un archivo SQL en el editor SQL

Cómo crear instrucciones y scripts SQL en el editor SQL

Puede crear instrucciones y scripts SQL de varias maneras:

- *Arrastrar y colocar:* arrastre un objeto del panel Explorador hasta el editor SQL. Automáticamente se genera una instrucción SQL para consultar el objeto.
- *Menú contextual*: haga clic con el botón derecho en un objeto del panel Explorador y seleccione **Mostrar en editor SQL | SELECT**.
- *Manualmente:* escriba las instrucciones SQL en el editor SQL. La función de finalización automática le ayudará durante la edición.
- *Importar un script SQL*: haga clic en el icono **Importar archivo SQL** de la barra de herramientas de la ventana "Consulta de base de datos".

Ejecución de instrucciones SQL

Si el script SQL del editor SQL tiene más de una instrucción SQL, seleccione la instrucción que desea ejecutar y haga clic en el icono **Ejecutar** o en **Ejecutar para editar datos** de la barra de herramientas. Si no selecciona ninguna instrucción del script SQL, se ejecutan todas las instrucciones del script. Los datos de la BD se recuperan y se muestran en forma de cuadrícula en la pestaña Resultados[1436]. Si selecciona la opción **Ejecutar para editar datos**, los datos recuperados de la cuadrícula de resultados se pueden editar[1436]. Los mensajes relacionados con la ejecución aparecen en la pestaña Mensajes[1436].

28.10.1.5 Resultados y mensajes

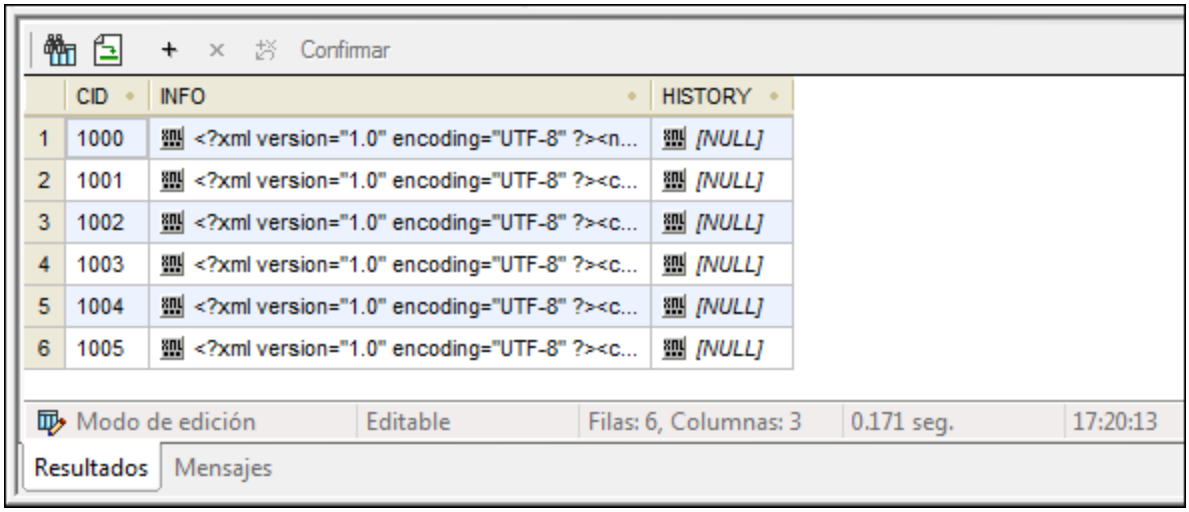
El panel de resultados y mensajes se divide en dos pestañas:

- La pestaña *Resultados*[1436] muestra los datos que recupera la consulta.
- La pestaña *Mensajes*[1438] muestra mensajes relacionados con la ejecución de la consulta.

Pestaña Resultados

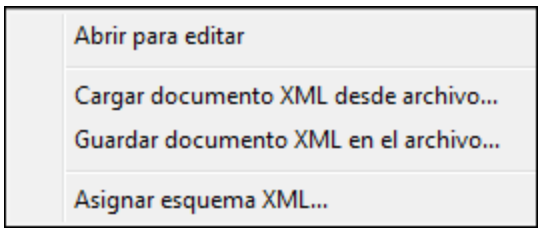
Los datos recuperados por la consulta se muestran en forma de cuadrícula en la pestaña *Resultados* (*imagen siguiente*). Cuando los resultados de la consulta se generan con el comando **Ejecutar consulta**, los documentos XML de la pestaña *Resultados* aparecen señalados con el icono XML  (*imagen siguiente*). Si usó el comando **Ejecutar para editar datos**, los documentos XML aparecen señalados con el icono .

© 2016-2022 Altova GmbH



A continuación describimos las operaciones que se pueden llevar a cabo en la pestaña *Resultados*, a través del menú contextual que aparece al hacer clic con el botón derecho:

- *Ordenar una columna:* haga clic con el botón derecho en cualquier parte de una columna y seleccione **Orden | Ascendente/Descendente/Restaurar valor predeterminado**.
- *Copiar en el portapapeles:* para ello es necesario (i) seleccionar el rango de datos y (ii) copiar la selección. Puede seleccionar los datos de varias formas: (i) haciendo clic en el encabezado de una columna o en el número de fila para seleccionar una columna o una fila respectivamente; (ii) seleccionando distintas celdas por separado (use la tecla **Mayús** o **Ctrl** para seleccionar varias celdas), (iii) haciendo clic con el botón derecho en una celda y seleccionando la opción **Seleccionar | Fila/Columna/Todo**. Después de seleccionar los datos, haga clic con el botón derecho y seleccione **Copiar las celdas seleccionadas**. Los datos se copian en el portapapeles para que los pueda pegar después en otra aplicación.
- *Anexar una fila nueva:* si la consulta se editó para ejecutar datos, haga clic con el botón derecho en cualquier parte y seleccione **Anexar una fila nueva**.
- *Eliminar una fila:* si la consulta se editó para ejecutar datos, haga clic con el botón derecho en cualquier fila y seleccione **Eliminar fila**.
- *Editar registros:* si la consulta se editó para ejecutar datos puede editar los diferentes campos de la cuadrícula. Para confirmar los cambios, pulse el botón **Confirmar** de la barra de herramientas de la pestaña *Resultados*.
- *Editar registros XML:* esta característica es compatible con bases de datos IBM DB2, SQL Server, PostgreSQL (8.3 y superior) y Oracle (9 y superior) y solamente con las tablas de BD que tienen una clave principal. Si la consulta se editó para ejecutar datos y uno de los campos es un campo XML, haga clic en el icono de la cuadrícula de resultados y aparece el menú de edición XML (*imagen siguiente*). También puede abrir un campo XML para editarlo haciendo clic con el botón derecho en el XML en el panel Explorador y seleccionando el comando **Editar datos**.



El comando **Abrir para editar** abre el documento XML en una ventana de XMLSpy y el icono del campo XML cambia a este otro icono, que incluye tres puntos rojos. Si el icono **Confirmar cambios XML automáticamente** de la barra de herramientas de la ventana de consulta estaba pulsado cuando abrió el documento XML, entonces los cambios realizados en el documento XML se confirman automáticamente en la BD. De lo contrario, los cambios guardados deben confirmarse con el botón **Confirmar** del panel *Resultados*. (Recuerde que para volver a la ventana de consulta de base de datos debe hacer clic en **Base de datos | Consultar la base de datos**.) El comando **Cargar documento XML desde archivo** carga un documento XML externo en el campo de la BD seleccionado. El comando **Guardar documento XML en el archivo** guarda el documento XML del campo de la BD seleccionado en el archivo que usted elija. El comando **Asignar esquema XML** abre el cuadro de diálogo Elegir esquema XML[1443], donde puede seleccionar un esquema XML y asignarlo al documento XML. Esta asignación se guarda en la BD. En el apartado IBM DB2 | Asignar esquema XML[1443] puede encontrar más información.

- *Establecer valor nulo; Establecer valor predeterminado; Deshacer los cambios realizados en esta celda:* si la consulta se editó para ejecutar datos, haciendo clic con el botón derecho en una celda puede establecer su valor como `NULL` o darle su valor predeterminado (si se definió un valor predeterminado). Los cambios realizados en la celda se pueden deshacer con el comando **Deshacer los cambios realizados en esta celda**. El valor de la celda se sustituye con el valor que existe en la BD.

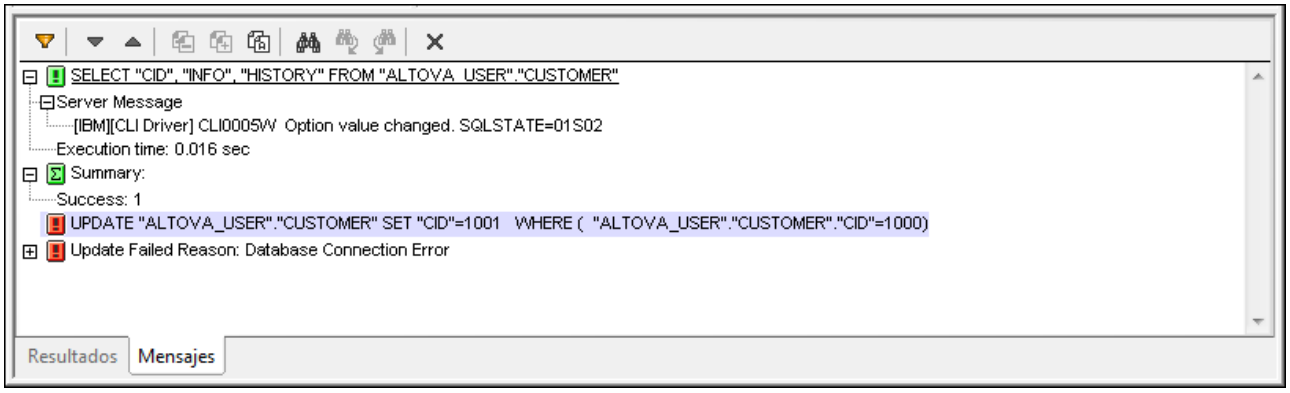
En la barra de herramientas de la pestaña **Resultados** puede encontrar estos iconos:

	Ir a la instrucción	Resalta en el editor SQL la instrucción que generó el resultado actual
	Buscar	Busca texto en el panel de resultados. También se busca en el contenido del documento XML
	Anexar una fila nueva	Añade una fila nueva a la cuadrícula de resultados
	Eliminar fila	Elimina la fila actual de la cuadrícula de resultados
	Deshacer todos los cambios	Deshace todos los cambios realizados en la cuadrícula de resultados
Commit	**Confirmar**	Confirma en la BD los cambios realizados en la cuadrícula de resultados

Pestaña Mensajes

La pestaña *Mensajes* ofrece información sobre la instrucción SQL recién ejecutada y le informa de errores o mensajes de advertencia.

© 2016-2022 Altova GmbH



La barra de herramientas de la pestaña *Mensajes* incluye iconos con los que podrá personalizar la vista, navegar por ella y copiar mensajes en el portapapeles. El icono **Filtro** permite mostrar/ocultar cierto tipo de mensajes. Los iconos **Siguiente** y **Anterior** sirven para recorrer la lista de mensajes. Los mensajes se pueden copiar en el portapapeles con o sin sus componentes secundarios. La función **Buscar** permite buscar un término en la lista de mensajes. Por último, el icono **Borrar** borra el contenido del panel *Mensajes*.

Nota: estos comandos de la barra de herramientas también están disponibles en el menú contextual que aparece al hacer clic con el botón derecho en el panel *Mensajes*.

28.10.2 IBM DB2

La opción de menú **IBM DB2** incluye un submenú con comandos para (i) registrar esquemas en una base de datos IBM DB2 y eliminarlos del registro de la BD (**Administrar esquemas XML** 1439) y (ii) para asignar esquemas y usarlos en la validación de archivos XML (**Asignar esquema XML** 1443).

Para utilizar estos dos comandos es necesario establecer una conexión a la base de datos IBM DB2 correspondiente. Para obtener ver un ejemplo de conexión consulte el apartado Conectarse a IBM DB2 (ODBC) 991. En el presente apartado nos centramos en cómo administrar los esquemas de una base de datos IBM DB2 y cómo asignar esquemas XML a un archivo XML de BD.

Nota: la cuadrícula de resultados de la ventana Consulta de base de datos 1423 incluye funciones importantes para trabajar con archivos XML en bases de datos IBM DB2. Entre ellas, destaca la función para abrir archivos y editarlos, cargar archivos XML en celdad de BD como archivos XML, guardar estos archivos XML en un directorio local y asignar esquemas XML a archivos XML de la BD.

28.10.2.1 Administrar esquemas XML

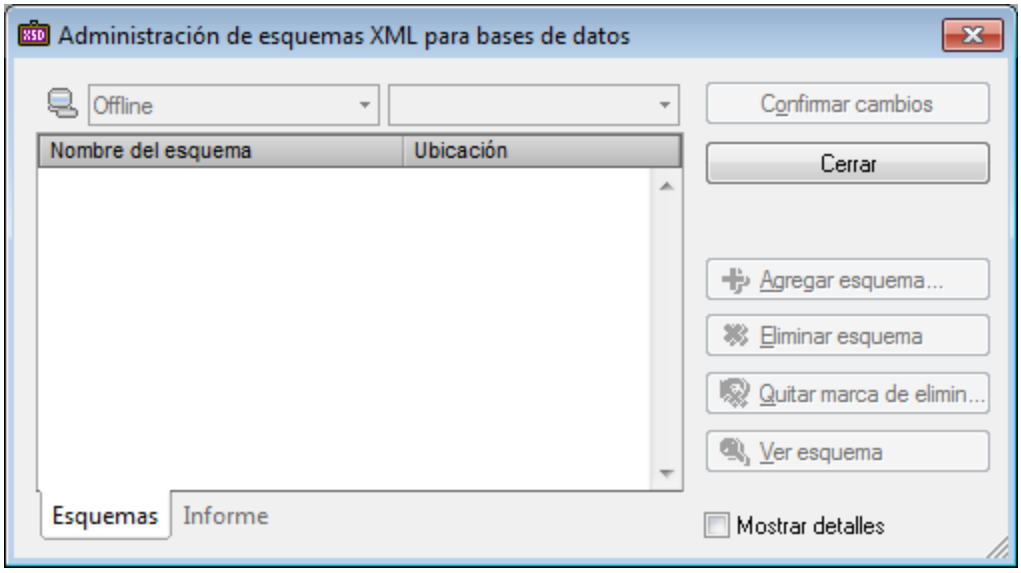
El comando **Administrar esquemas XML** sirve para añadir esquemas a una base de datos IBM DB2 o para eliminarlos del registro de la BD. Para poder administrar esquemas es necesario:

- Conectarse a la base de datos IBM DB2
- Seleccionar el esquema de BD en el que desea añadir o eliminar esquemas
- Llevar a cabo las tareas de administración de esquemas XML

En este apartado describimos todos estos pasos.

Conectarse a la base de datos IBM DB2

Al hacer clic en el comando **Administrar esquemas XML** se abre el diálogo "Administración de esquemas XML para bases de datos" (*imagen siguiente*).

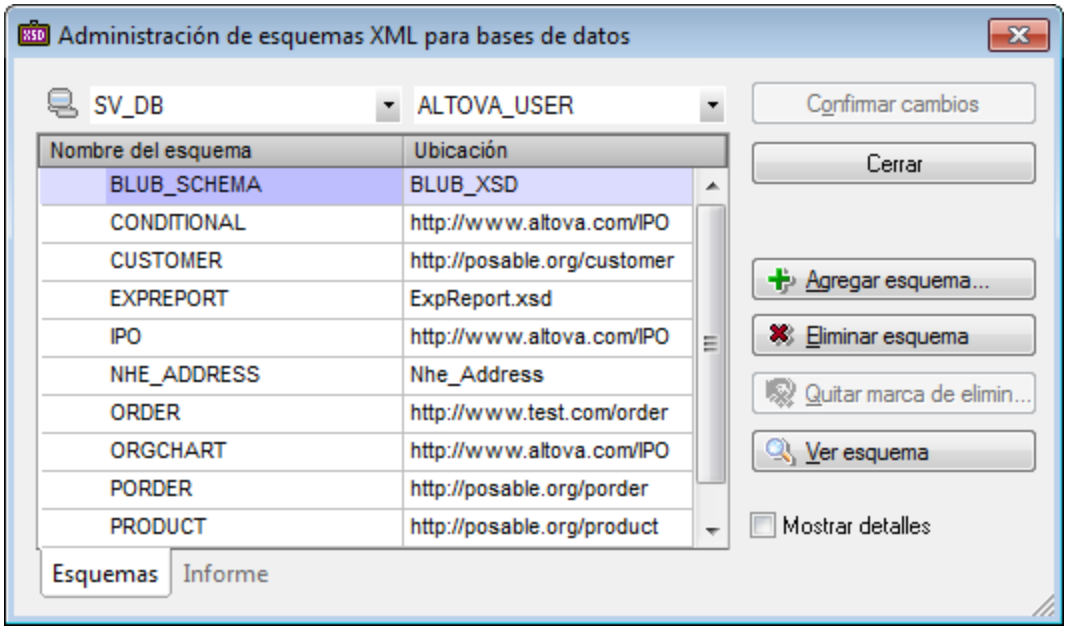


Lo primero que debe hacer es conectarse a la BD. Si la conexión ya existe, la BD aparece en el cuadro combinado de BD. Para iniciar el proceso de conexión, haga clic en el icono de conexión rápida del cuadro de diálogo. Aparece el cuadro de diálogo de conexión, donde puede establecer la conexión a la BD (consulte las instrucciones de la sección *Conectarse a una base de datos* 952).

Ver la lista de esquemas XML

Tras conectarse a la base de datos IBM DB2, la BD aparece en el cuadro combinado situado a la izquierda (*imagen siguiente*). Si está conectado a más de una BD, puede seleccionar la que desea usar en el cuadro combinado. En el ejemplo de la imagen siguiente seleccionamos la base de datos `SV_DB`.

© 2016-2022 Altova GmbH



El cuadro combinado de la derecha muestra todos los esquemas de BD de la base de datos IBM DB2 seleccionada. Cuando seleccione un esquema de la BD en este cuadro combinado, todos los esquemas XML registrados en él aparecen en el panel principal. En la imagen anterior, se enumeran todos los esquemas XML registrados en el esquema de la BD `ALTOVA_USER` y su ubicación. Si marca la casilla *Mostrar detalles*, en el panel principal aparecen más columnas con información adicional sobre los esquemas.

Administrar los esquemas XML

La lista de esquemas XML del panel principal enumera los esquemas registrados en el esquema de BD seleccionado. Ahora puede añadir o eliminar esquemas en la lista.

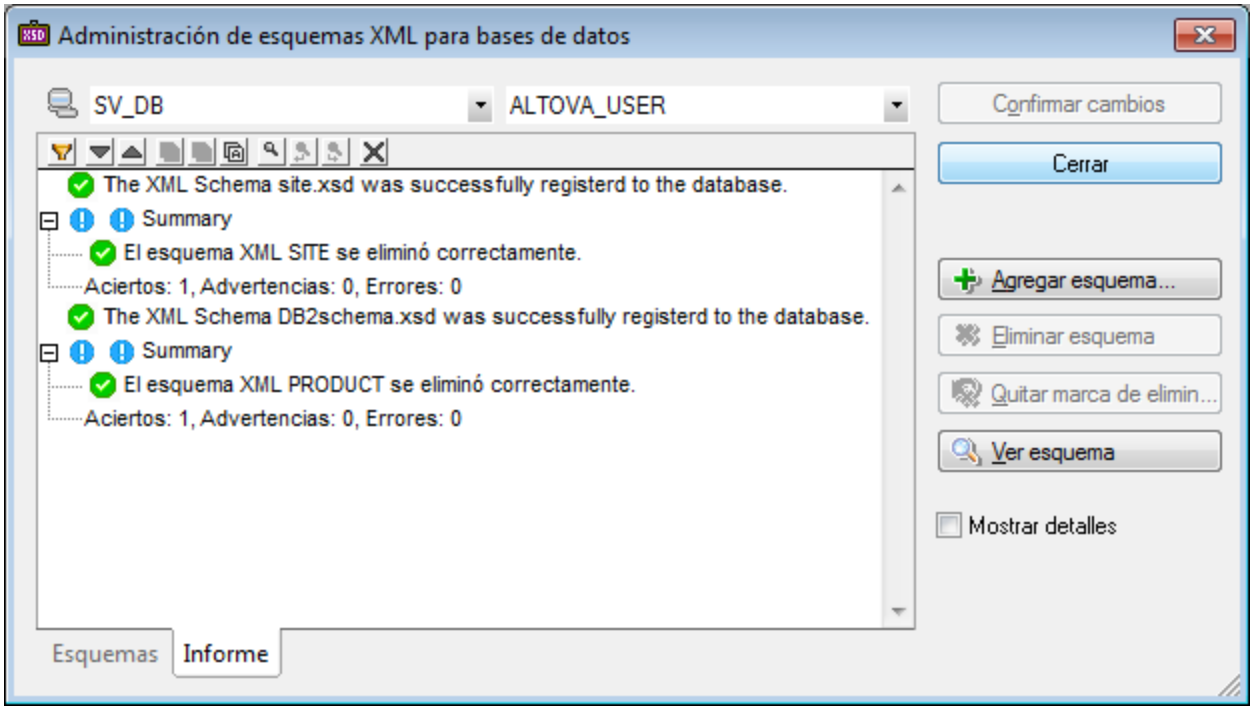
Para añadir un esquema, haga clic en el botón **Agregar esquema**, busque el archivo de esquema y selecciónelo. El archivo de esquema se añade a la lista del panel principal. Al hacer clic en el botón **Confirmar cambios**, el esquema recién añadido se registra en el esquema de la BD.

Para eliminar un esquema, seleccione el esquema de la lista y haga clic en el botón **Eliminar esquema**. Junto al esquema aparece ahora una marca de eliminación, que indica que el esquema será eliminado cuando se confirmen los cambios. Si desea eliminar esta marca de eliminación, seleccione el esquema y haga clic en el botón **Quitar marca de eliminación**. Cuando haga clic en el botón **Confirmar cambios**, todos los esquemas que tengan una marca de eliminación serán eliminados del registro del esquema de la BD.

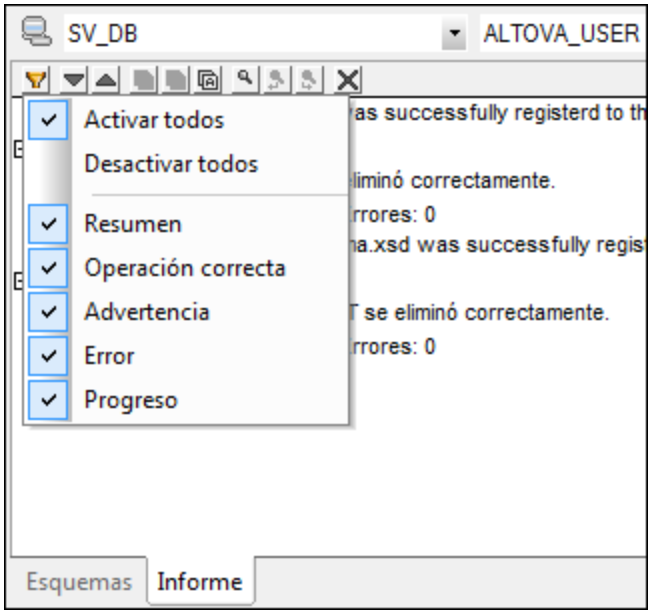
Si hace clic en el botón **Ver esquema**, el esquema se abre en XMLSpy. Para cerrar el cuadro de diálogo "Administración de esquemas XML" haga clic en el botón **Cerrar**.

Informes

Al hacer clic en el botón **Confirmar cambios**, la BD se modifica con los cambios realizados. En la pestaña *Informe* aparece un informe sobre la acción de confirmación de cambios (*imagen siguiente*). Así puede ver si la acción se completó correctamente y corregir cualquier error. Los informes aparecen uno después del otro en este panel.



La pestaña *Informes* tiene una barra de herramientas con varios iconos con los que podrá personalizar la vista, navegar por el informe, copiar mensajes, buscar texto y borrar los mensajes (*imagen siguiente*).



Con el icono **Filtro** puede ver cierto tipo de mensajes y ocultar otros. Los iconos **Siguiente** y **Anterior** sirven para recorrer la lista de mensajes. También puede copiar los mensajes en el portapapeles, con o sin sus componentes secundarios. La función **Buscar** sirve para buscar un término de búsqueda en el informe. Por último, con el icono **Borrar** puede borrar el informe.

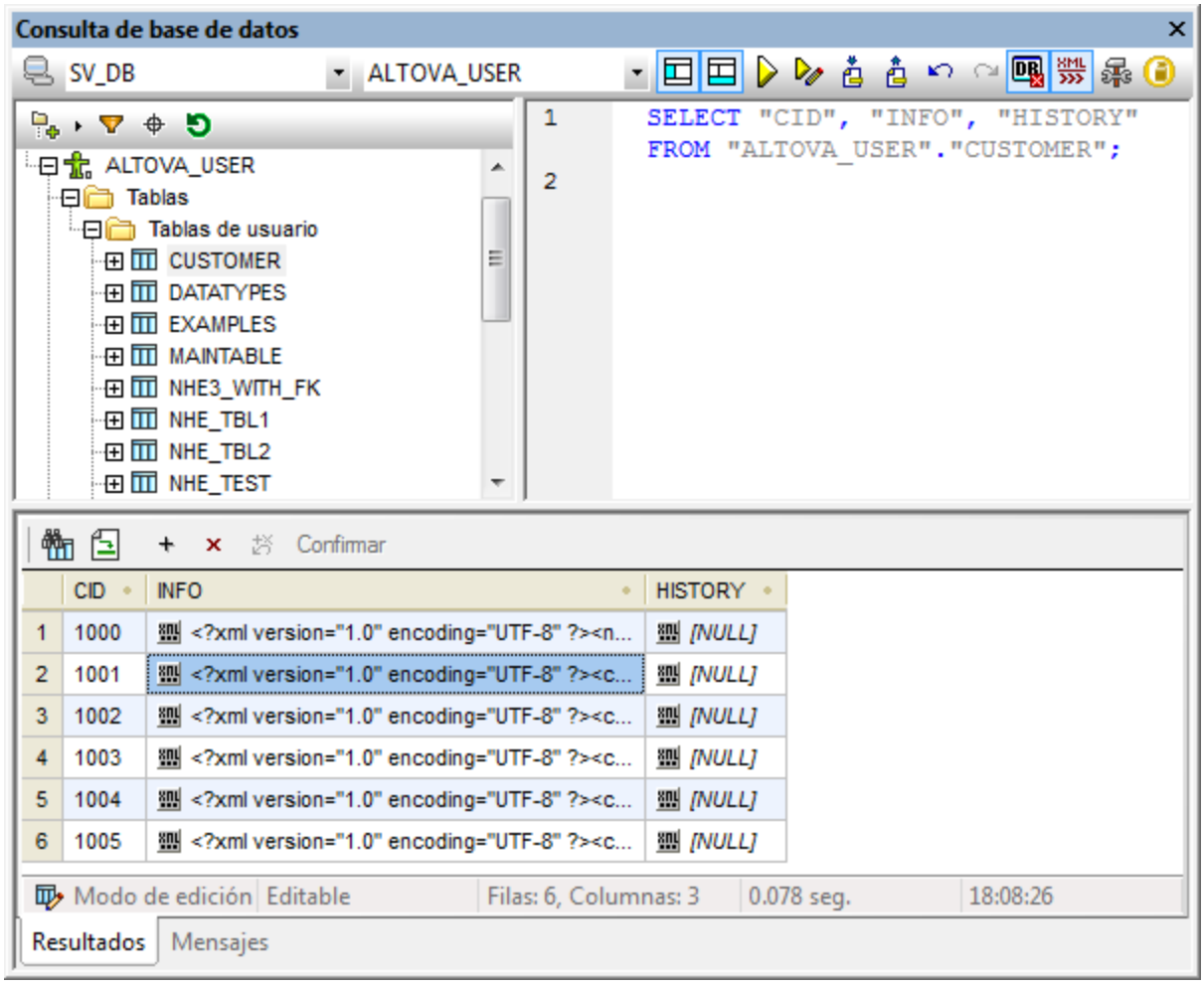
© 2016-2022 Altova GmbH

28.10.2.2 Asignar esquema XML

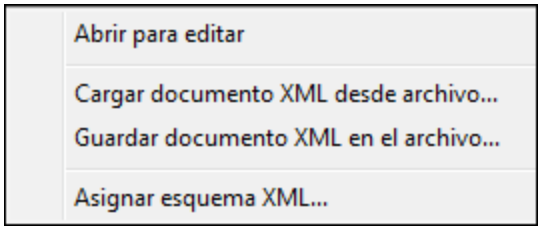
El comando **Asignar esquema XML** sirve para asignar un esquema a un archivo XML cuando el archivo XML se abre desde la cuadrícula de resultados de la ventana "Consulta de base de datos". Después de asignarle el esquema, el archivo se puede validar con el esquema asignado. La asignación de esquema se escribe en la BD al guardar el archivo XML en XMLSpy.

Abrir un archivo XML de BD para editarlo

En la ventana "Consulta de base de datos", cuando se ejecuta una consulta para una BD XML con el comando **Ejecutar para editar datos**, la cuadrícula de resultados del panel Resultados permite acceder a los archivos XML de la BD y editarlos (*imagen siguiente*).



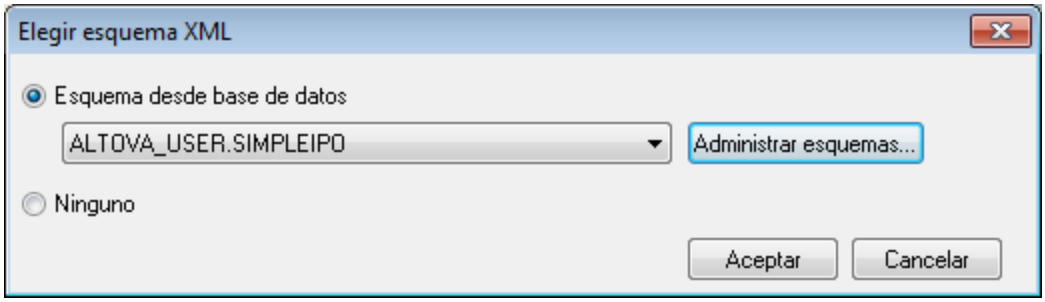
Al hacer clic en el icono aparece este menú.



Si selecciona el comando **Abrir para editar**, el documento XML se abre en XMLSpy, donde puede editarlo.

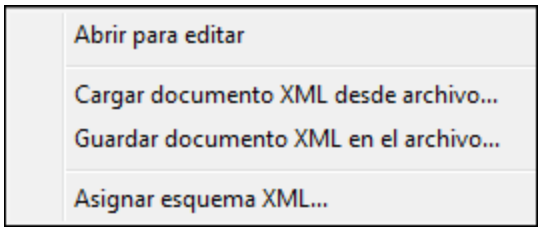
Asignar un esquema al archivo XML de la BD

Cuando el archivo XML de la BD está abierto en XMLSpy se habilita el comando **IBM DB2 | Asignar esquema XML**. Al hacer clic en este comando, aparece el cuadro de diálogo "Elegir esquema XML" (*imagen siguiente*).



Puede seleccionar un esquema de los que están almacenados en la BD (de la lista desplegable del cuadro combinado *Esquema desde base de datos*) o un archivo externo. Al hacer clic en **Aceptar**, el esquema se asigna al archivo XML. Recuerde que la asignación no se escribe en el archivo XML. Cuando el archivo XML se guarda en XMLSpy, la asignación de esquema se guarda en la BD (siempre que estuviera activado el icono **Confirmar cambios XML automáticamente** en la barra de herramientas de la ventana de consulta cuando se abrió el archivo XML) y no en el archivo XML.

Nota: el menú de edición XML de la cuadrícula de resultados de la ventana "Consulta de base de datos" ofrece el comando **Asignar esquema XML** (*imagen siguiente*), que también sirve para asignar un esquema al archivo XML de la BD.



La diferencia entre estos dos comandos es que el comando **Asignar esquema XML** del menú **BD | IBM DB2** permite asignar un esquema XML mientras se edita el archivo XML, es decir, permite cambiar las asignaciones de esquema mientras edita el documento XML y validar el documento XML inmediatamente.

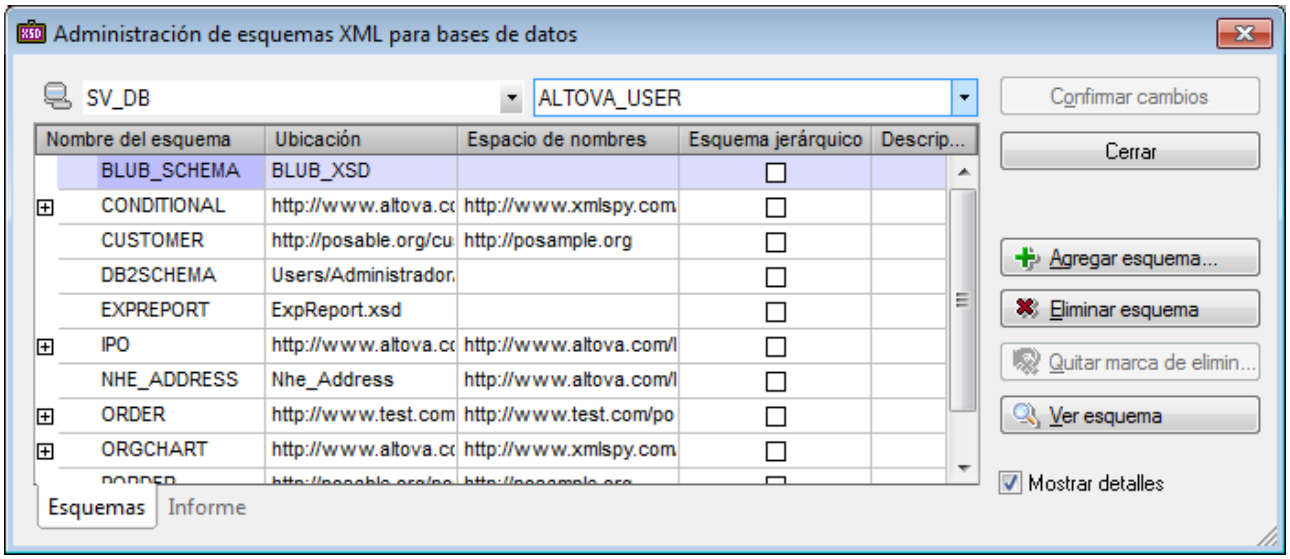
© 2016-2022 Altova GmbH

28.10.3 SQL Server

La opción de menú **SQL Server** incluye un submenú con el comando **Administrar esquemas XML**.

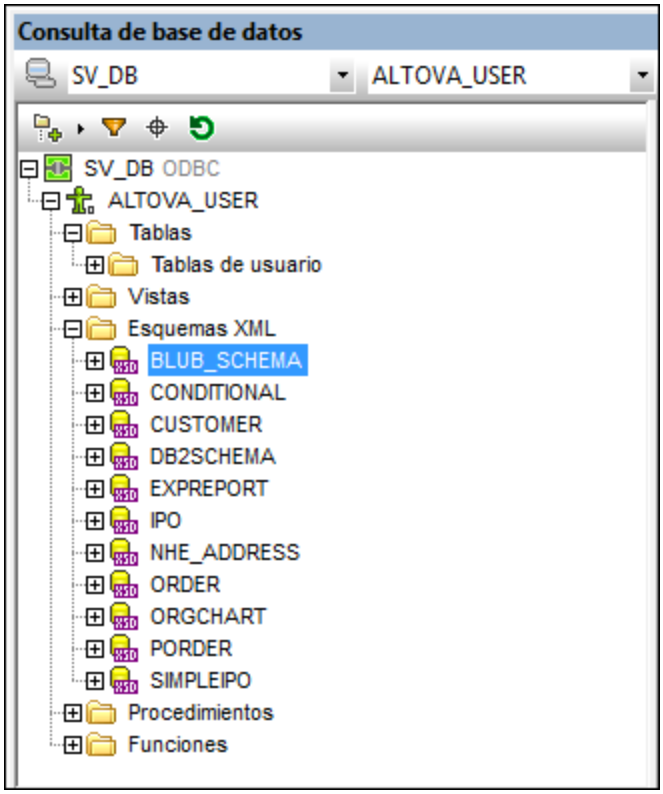
28.10.3.1 Administrar esquemas XML

La función de administración de esquemas XML para BD sirve para añadir y eliminar esquemas XML del repositorio de esquemas de una BD XML. Tras conectarse a la BD, XMLSpy le ofrece la opción de abrir el cuadro de diálogo "Administración de esquemas XML para bases de datos", donde puede gestionar sus esquemas.



El cuadro de diálogo ofrece un icono de conexión rápida que llama al Asistente para la conexión a bases de datos, que le ayudará a conectarse a un origen de datos. Si está conectado a más de una BD, puede seleccionar una conexión en el cuadro combinado situado a la izquierda. El objeto raíz de la BD se selecciona en el cuadro combinado de la derecha. Tras seleccionar la BD y el objeto raíz, aparecen en el panel todos los esquemas XML registrados en el repositorio seleccionado. En el panel se enumera el nombre, la ubicación y el espacio de nombres de cada esquema.

Recuerde que los esquemas también se pueden ver en la ventana "Consulta de base de datos" (*imagen siguiente*), aunque en esa ventana no puede administrarlos. Los esquemas XML solamente se pueden administrar en el diálogo "Administración de esquemas XML para bases de datos".

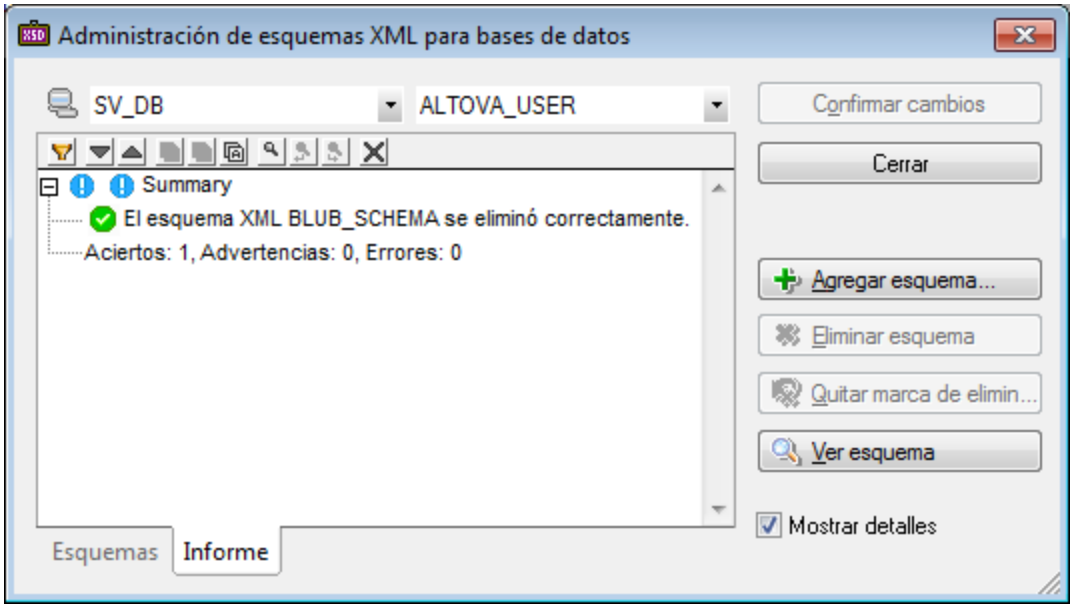


En el cuadro de diálogo "Administración de esquemas XML para bases de datos" puede:

- Añadir un esquema con el botón **Agregar esquema**. El esquema seleccionado se añade a la lista.
- Añadir una marca de eliminación a cualquier esquema de la lista haciendo clic en el botón **Eliminar esquema**. Para quitar la marca de eliminación utilice el botón **Quitar marca de eliminación**.
- Abrir el esquema seleccionado en la vista Esquema haciendo clic en el botón **Ver Esquema**.
- Confirmar en la BD los cambios realizados con el botón **Confirmar cambios**.

Tras confirmar los cambios, aparece un informe en la pestaña *Informe* (*imagen siguiente*).

© 2016-2022 Altova GmbH



28.10.4 Oracle XML DB

En XMLSpy puede conectarse a bases de datos Oracle XML DB y consultarlas.

Estas son las funciones compatibles con este tipo de BD:

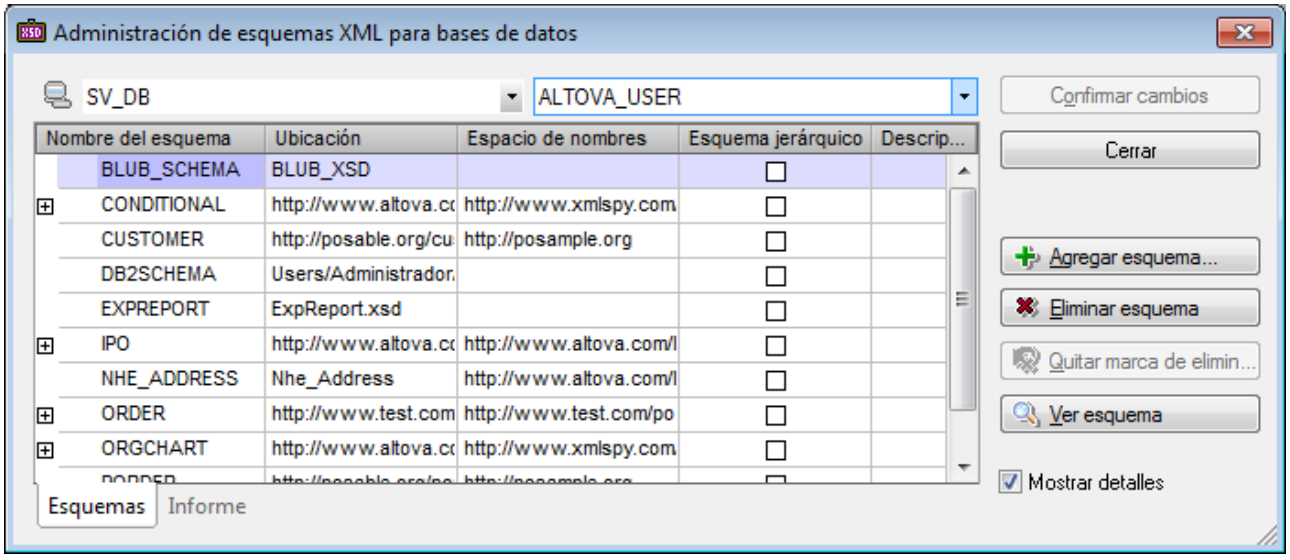
- Añadir (y registrar) un esquema XML en la base de datos Oracle XML DB. Para ello es necesario tener instalado el cliente de Oracle XML DB.
- Abrir y eliminar esquemas
- Consultar la BD usando instrucciones XPath (DBUri)
- Examinar documentos XML (usando WebDAV)
- Crear un documento XML basado en un esquema guardado en la BD

Proceso general de instalación

- Descargue e instale XMLSpy
- Instale el servidor de Oracle (si es necesario)
- Cree una base de datos Oracle

28.10.4.1 Administrar esquemas XML

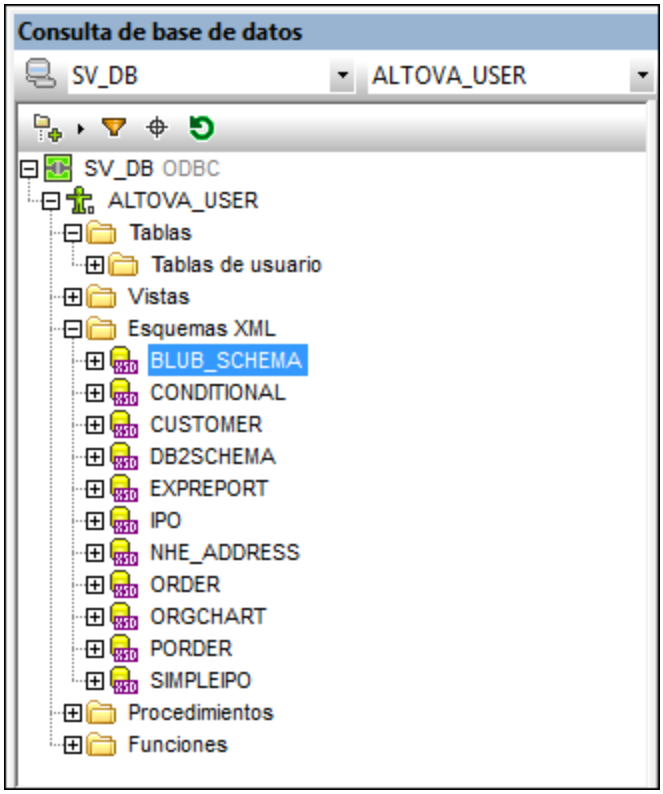
La función de administración de esquemas XML para BD sirve para añadir y eliminar esquemas XML del repositorio de esquemas de una BD XML. Tras conectarse a la BD, XMLSpy le ofrece la opción de abrir el cuadro de diálogo "Administración de esquemas XML para bases de datos", donde puede gestionar sus esquemas.



El cuadro de diálogo ofrece un icono de conexión rápida  que llama al Asistente para la conexión a bases de datos, que le ayudará a conectarse a un origen de datos. Si está conectado a más de una BD, puede seleccionar una conexión en el cuadro combinado situado a la izquierda. El objeto raíz de la BD se selecciona en el cuadro combinado de la derecha. Tras seleccionar la BD y el objeto raíz, aparecen en el panel todos los esquemas XML registrados en el repositorio seleccionado. En el panel se enumera el nombre, la ubicación y el espacio de nombres de cada esquema.

Recuerde que los esquemas también se pueden ver en la ventana "Consulta de base de datos" (*imagen siguiente*), aunque en esa ventana no puede administrarlos. Los esquemas XML solamente se pueden administrar en el diálogo "Administración de esquemas XML para bases de datos".

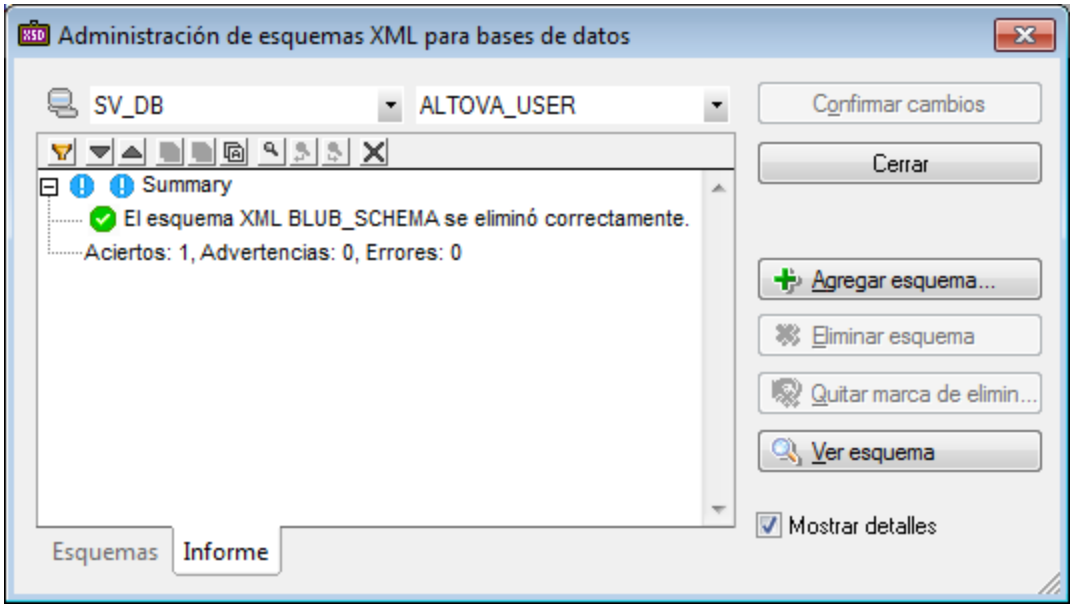
© 2016-2022 Altova GmbH



En el cuadro de diálogo "Administración de esquemas XML para bases de datos" puede:

- Añadir un esquema con el botón **Agregar esquema**. El esquema seleccionado se añade a la lista.
- Añadir una marca de eliminación a cualquier esquema de la lista haciendo clic en el botón **Eliminar esquema**. Para quitar la marca de eliminación utilice el botón **Quitar marca de eliminación**.
- Abrir el esquema seleccionado en la vista Esquema haciendo clic en el botón **Ver Esquema**.
- Confirmar en la BD los cambios realizados con el botón **Confirmar cambios**.

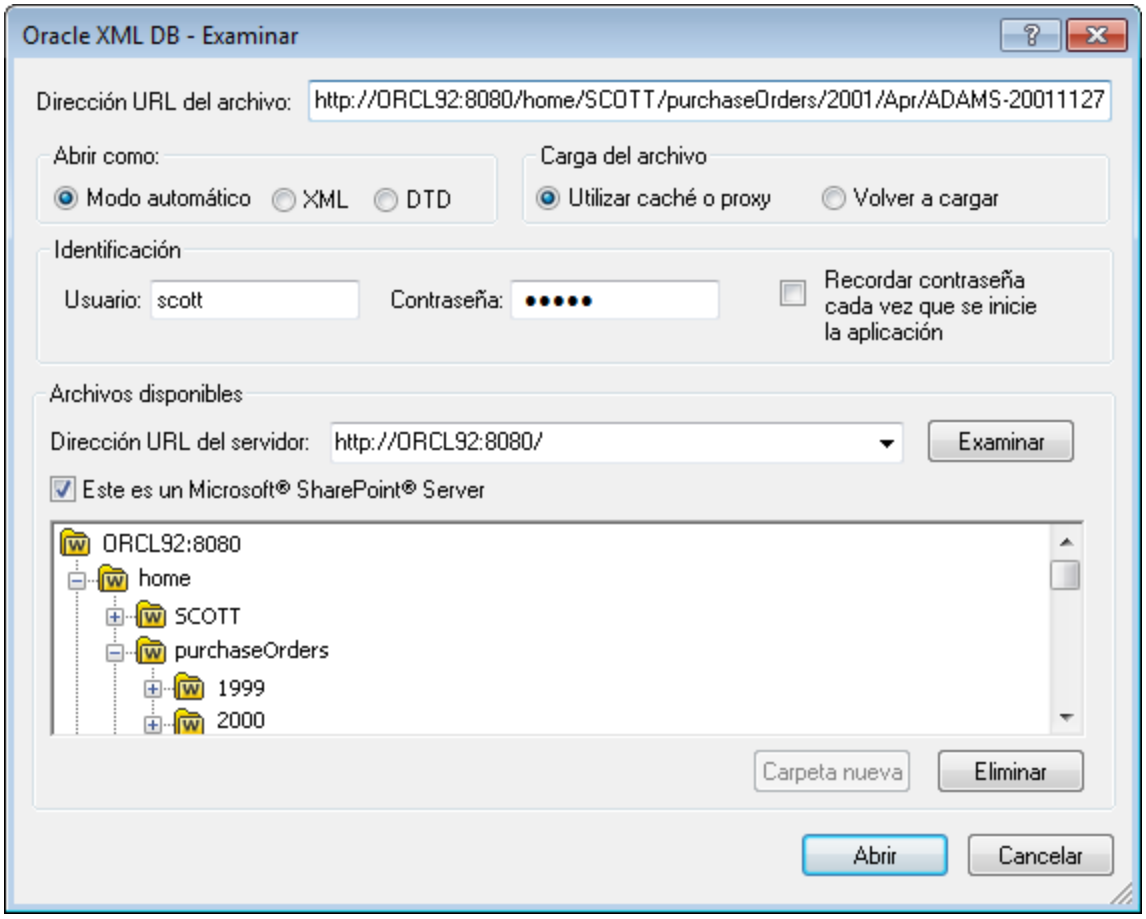
Tras confirmar los cambios, aparece un informe en la pestaña *Informe* (*imagen siguiente*).



28.10.4.2 Examinar documentos XML de Oracle

Con este comando puede examinar los documentos XML disponibles en el servidor. Los datos del servidor ya están disponibles si previamente consultó la BD o administró sus esquemas. De lo contrario, es necesario introducir los datos a mano.

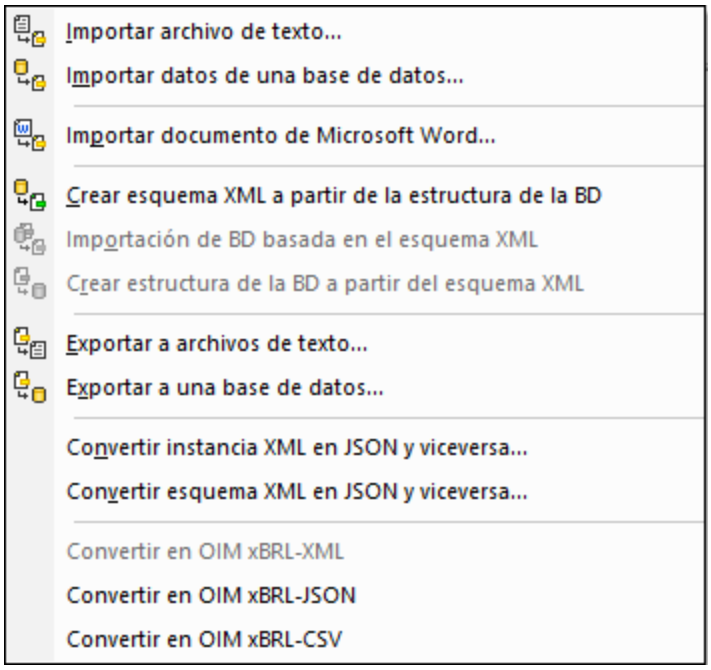
© 2016-2022 Altova GmbH



Use la vista en forma de árbol para buscar sus archivos XML. Al hacer doble clic en un archivo en la vista en forma de árbol, el archivo se abre en la aplicación. Otra opción es hacer clic en un archivo y pulsar el botón **Abrir**. El botón **Carpeta nueva** añade una carpeta nueva y el botón **Eliminar** elimina el archivo XML que está seleccionado en ese momento.

28.11 Menú Conversión

El menú **Conversión** (*imagen siguiente*) ofrece potentes funciones de conversión de datos para:



- Importar y exportar archivos de texto, Word, archivos de base de datos y archivos XML.
- Importar datos de BD[1466] basados en un esquema XML.
- Crear un esquema XML[1461] a partir de la estructura de una base de datos.
- Crear una **estructura de base de datos**[1467] a partir de un esquema XML.
- Convertir instancias en XML en instancias JSON y viceversa[1476] o esquemas XML en esquemas JSON y viceversa[1480].
- Convertir datos XBRL en las representaciones OIM de xBRL-JSON y xBRL-CSV, y convertir cualquier representación OIM (xBRL-XML, xBRL-JSON y xBRL-CSV) en otra.

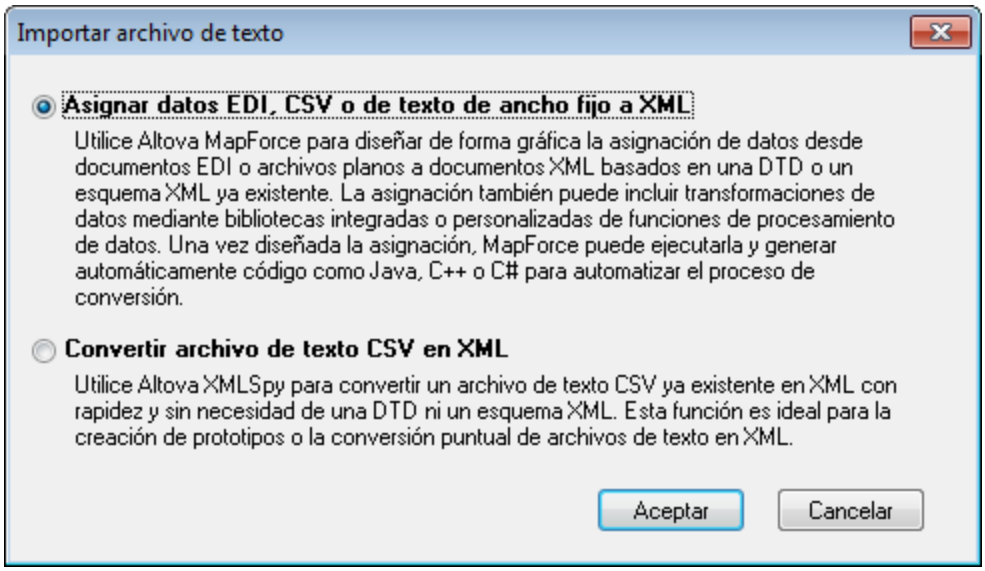
28.11.1 Importar archivo de texto



Este comando sirve para importar archivos de texto estructurado a XMLSpy y pasarlos a formato XML inmediatamente. Es una función muy útil si necesita importar datos heredados de sistemas antiguos. A continuación explicamos paso a paso cómo importar datos de un archivo de texto como documento XML.

1. Seleccione la opción de menú **Conversión | Importar archivo de texto**. Aparece este cuadro de diálogo:

© 2016-2022 Altova GmbH



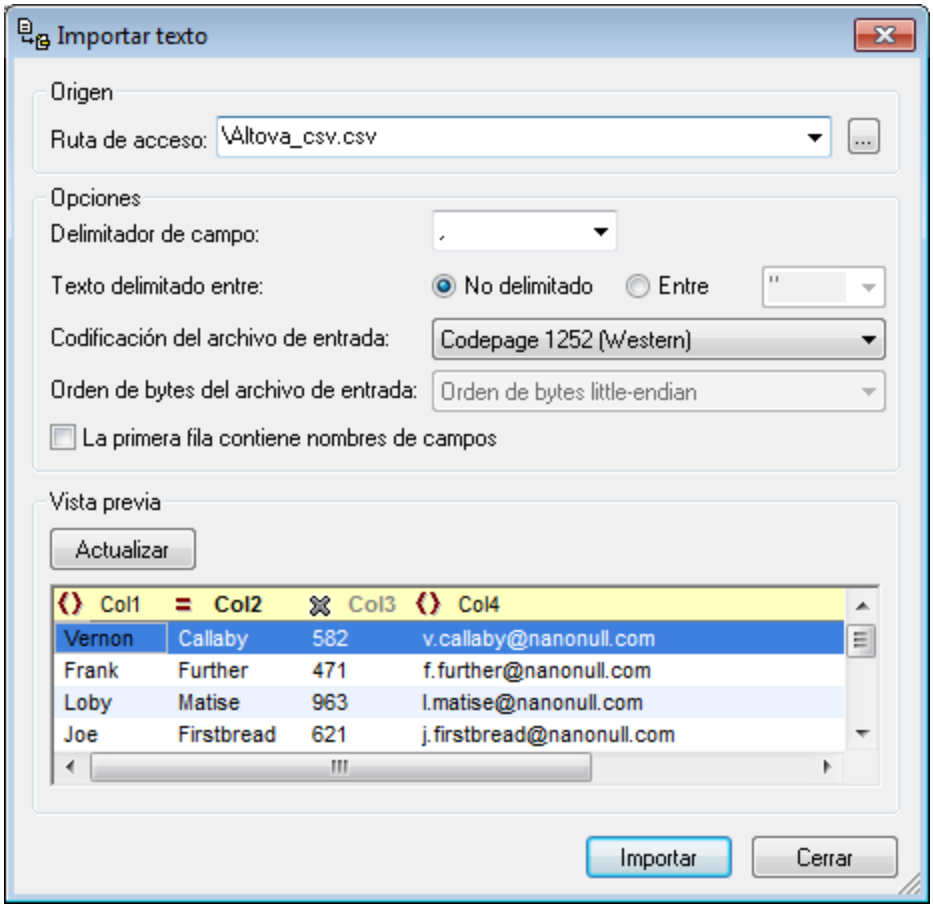
2. Seleccione una de estas dos opciones:

   - *Asignar datos EDI, CSV o de texto de ancho fijo a XML* (para poder seleccionar esta opción es necesario tener instalado Altova MapForce)
   - *Convertir archivo de texto CSV en XML*

3. Haga clic en **Aceptar**. Aparece el cuadro de diálogo "Importar texto". Seleccione las opciones que prefiera (*consulte el punto siguiente*). Cuando termine de configurar todas estas opciones, haga clic en **Importar**. Los datos importados se convierten en un documento XML que se abre en la vista Cuadrícula.

Opciones de importación

En el cuadro de diálogo "Importar texto" (*imagen siguiente*) puede elegir las opciones que quiera para importar texto. Estas opciones se describen más abajo.



Ruta de acceso
Escriba la ruta de acceso del archivo que desea importar en el cuadro de texto *Ruta de acceso*. También puede buscar el archivo haciendo clic en el botón *Examinar*, situado a la derecha del cuadro de texto. Una vez seleccionado el archivo, en el panel *Vista previa* aparece el archivo XML resultante en forma de cuadrícula. A partir de ahora, las opciones que seleccione en este cuadro de diálogo se reflejan inmediatamente en esta vista previa del archivo XML.

Delimitador
Para importar un archivo de texto correctamente, es necesario especificar qué delimitador de campo se usa en el archivo para separar las columnas o los campos. XMLSpy detecta automáticamente los separadores de filas más comunes (CR, LF y CR+LF).

Comillas
Los archivos de texto exportados de sistemas heredados a veces llevan valores de texto delimitados entre comillas, para distinguirlos mejor de los valores numéricos. Si este es el caso, puede especificar qué tipo de comillas delimitan el texto para que se eliminen automáticamente cuando se importen los datos.

Codificación
Los datos se pasan a Unicode[2249] (que es la base de todos los documentos XML), de modo que debe especificar en qué juego de caracteres está codificado el archivo que desea importar. Codepage 1252 (también conocida como ANSI) es la codificación más frecuente de los sistemas Windows occidentales.

© 2016-2022 Altova GmbH

Orden de bytes
Si el archivo que desea importar es Unicode de 16 o 32 bits (UCS-2, UTF-16 o UCS-4), también puede elegir entre un orden de bytes little-endian o big-endian.

La primera fila contiene nombres de campos
Los archivos de texto suelen incluir el nombre del campo en la primera fila del archivo. Si este es el caso, marque esta casilla.

Vista previa
En el panel *Vista previa* puede cambiar el nombre de los encabezados de columna (haga clic en el encabezado y edítelo). Los encabezados de columna equivalen al nombre de elemento o atributo en el documento XML. También puede elegir si la columna es un elemento o atributo en el documento XML o si se debe importar o no al documento XML. Para seleccionar estas opciones haga clic en el icono del tipo de columna hasta que aparezca la opción deseada. En la imagen anterior, la columna `Col1` es un elemento, `Col2` es un atributo y `Col3` no se importará al documento XML.

28.11.2 Importar datos de una base de datos



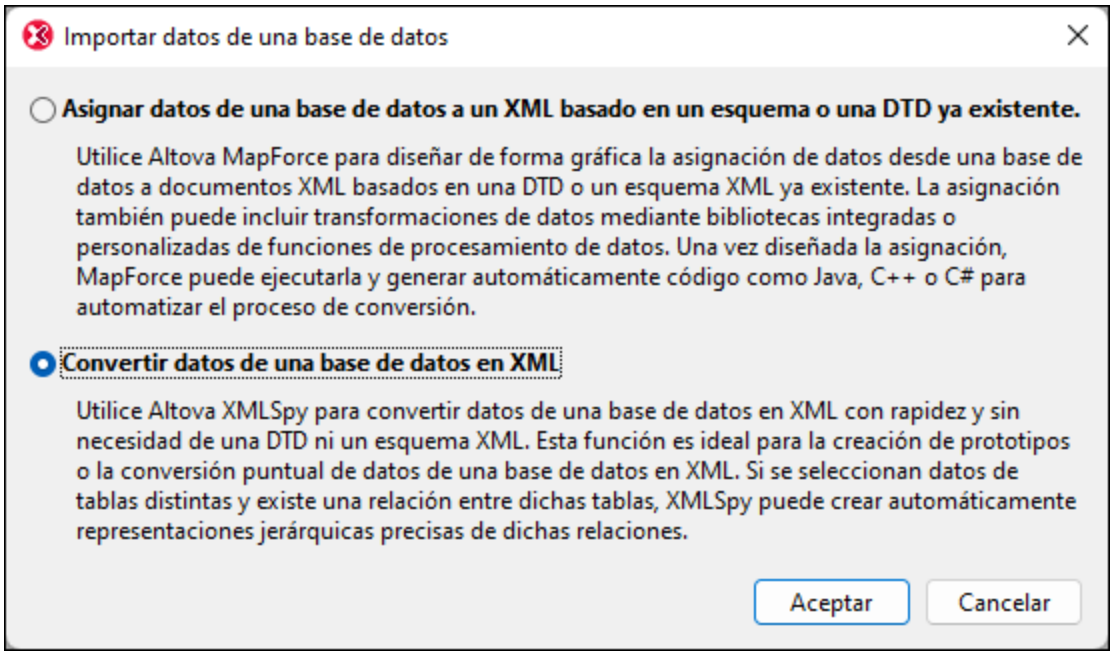
El comando **Importar datos de una base de datos** sirve para importar datos de una BD y convertirlos en un archivo XML. El mecanismo de importación tiene dos fases:

1. Primero se establece la conexión a la base de datos (consulte las instrucciones de la sección *Conectarse a una base de datos* 952).
2. Después se seleccionan los datos que se deben importar 1456.

Seleccionar el comando de importación

Siga estos pasos para importar los datos de la BD:

1. Haga clic en el comando **Importar datos de una base de datos**. Aparece el cuadro de diálogo "Importar datos de una base de datos" (*imagen siguiente*).



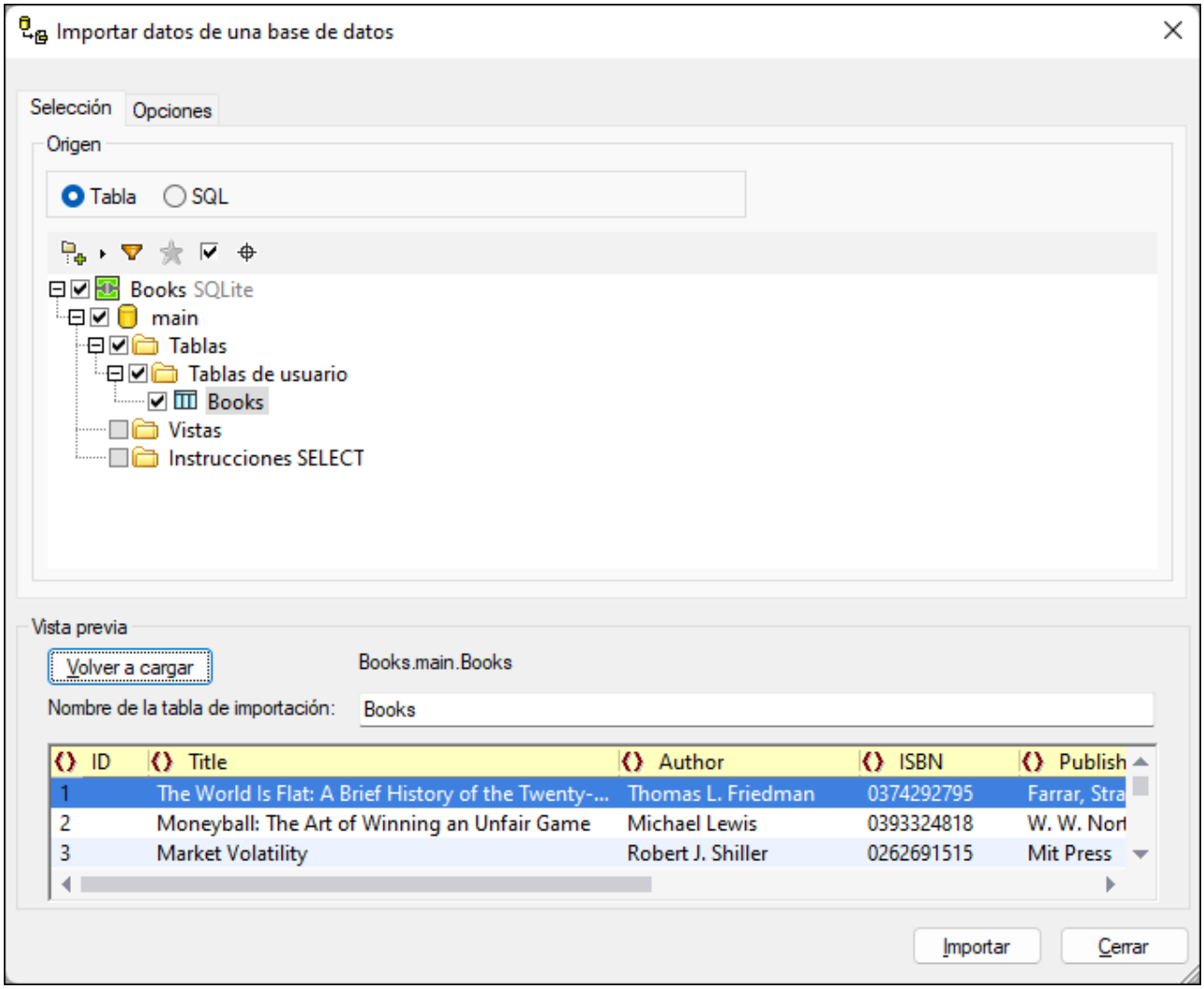
2. Seleccione la opción *Convertir datos de una base de datos en XML* y haga clic en **Aceptar**. Aparece el cuadro de diálogo "Conectarse a un origen de datos". (Para la opción *Asignar datos de una base de datos a un XML basado en un esquema o una DTD ya existente*, es necesario diseñar y ejecutar la asignación de datos con Altova MapForce.)
3. En el cuadro de diálogo "Conectarse a un origen de datos" seleccione la BD a la que se desea conectar. (consulte las instrucciones de la sección *Conectarse a una base de datos* 952).
4. Una vez establecida la conexión, aparece el diálogo "Importar datos de una base de datos" que incluye paneles y pestañas donde puede seleccionar los datos que desea importar. Más abajo describimos las opciones de este cuadro de diálogo.
   Cuando termine de configurar estas opciones, haga clic en **Importar** para importar los datos de la BD.

Selección de datos y opciones de importación

El cuadro de diálogo "Importar datos de una base de datos" se divide en dos partes (*ver imágenes siguientes*):

- la parte superior, que a su vez se divide en dos pestañas: (i) *Selección* y (ii) *Opciones*.
- la parte inferior, una ventana de vista previa en la que se muestran los datos en base a las opciones de importación y selección de datos seleccionadas.

© 2016-2022 Altova GmbH



Pestaña Selección

En la pestaña *Selección* (*imagen anterior*), el panel *Origen* (*imagen siguiente*) muestra una representación en forma de árbol de las tablas de la BD (botón de opción *Tabla*) o una instrucción SQL editable para seleccionar las tablas de la BD (botón de opción *SQL*).



Opciones de selección de la vista Tabla

Si selecciona la vista *Tabla*, puede seleccionar las tablas de la BD que desea importar (*marcando sus casillas como en la imagen anterior*). Si hace clic en el nombre de una tabla, su contenido se muestra en el panel *Vista previa*, donde puede seguir seleccionando y filtrando los datos de la tabla (*véase más abajo*).

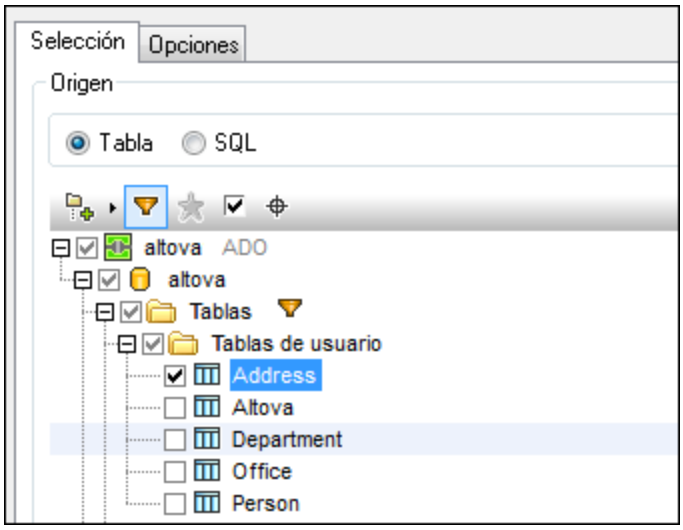
La vista de selección de tabla permite seleccionar comandos con los iconos de la barra de herramientas.



Estos son los iconos, de izquierda a derecha:

- *El icono Disposición:* sirve para organizar los objetos de la BD: (i) por *carpetas*, una por cada tipo de objeto; (ii) por carpetas pero *sin esquemas*; (iii) de forma jerárquica pero *sin carpetas* y (iv) según el tipo de tabla, en función de sus *dependencias con otras tablas*.
- *Filtrar contenido de carpeta:* aparece un icono en forma de embudo junto a las carpetas (*véase la carpeta Tablas de la imagen siguiente*). Al hacer clic en el icono en forma de embudo, aparece un menú con varias opciones de filtrado. En la imagen siguiente, por ejemplo, configuramos el filtro para ver solamente los objetos cuyo nombre contiene el texto `Sys`. Si hace clic en el icono aparece un menú que permite filtrar la información de distintas maneras.

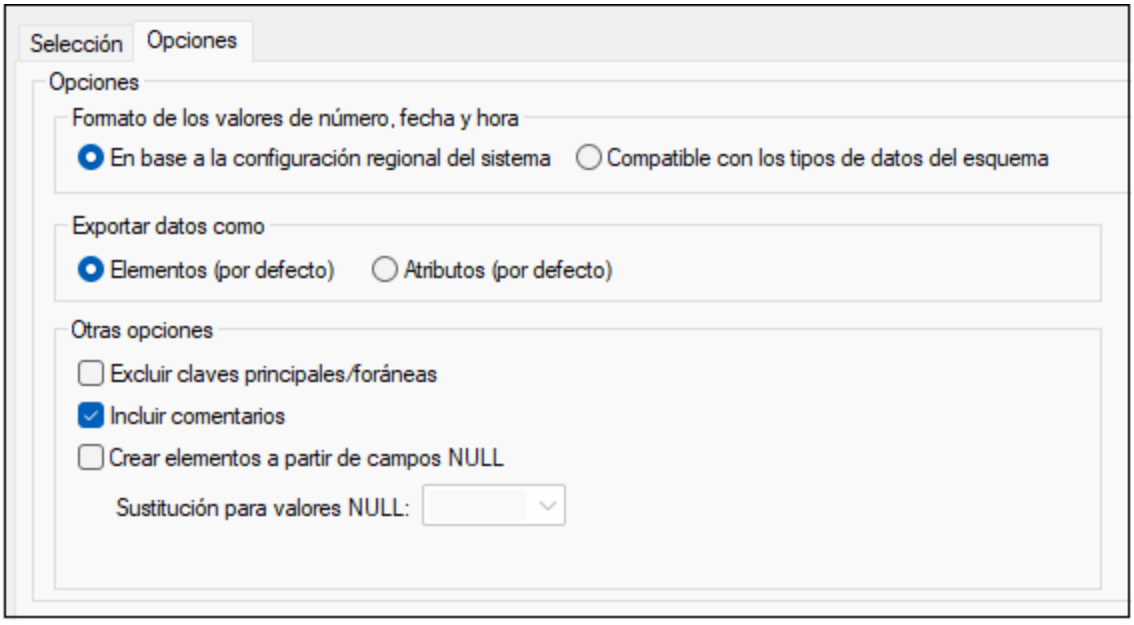
© 2016-2022 Altova GmbH



- *Ver favoritos:* muestra solo los objetos marcados como favoritos.
- *Mostrar únicamente los objetos seleccionados*: muestra solamente los objetos seleccionados.
- *Localizador de objetos:* aparece un campo de texto que funciona como una función de búsqueda. Puede escribir una cadena de texto y aparece una lista desplegable con todos los objetos de la tabla cuyo nombre contiene dicha cadena de texto. Al seleccionar uno de los objetos en la lista desplegable, el objeto aparece resaltado en la vista en forma de árbol.

Pestaña Opciones

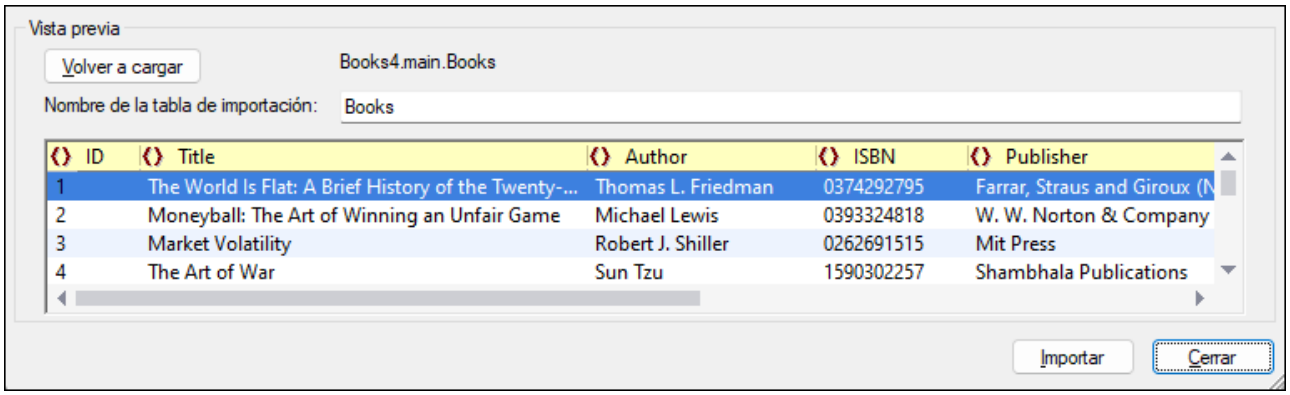
En la pestaña *Opciones* (*imagen siguiente*) puede especificar: (i) cómo se importan los valores de número, fecha y hora, (ii) si los datos se importan como elementos o atributos y (iii) si los comentarios y los campos `NULL` se incluyen en la importación.



Si decide importar los campos `NULL`, puede introducir un valor XML de sustitución para ellos.

Panel Vista previa

El panel *Vista previa* (*imagen siguiente*) muestra la estructura de la tabla que está seleccionada en la pestaña *Selección*. Cuando seleccione una tabla nueva en la pestaña *Selección*, haga clic en el botón **Volver a cargar** de la vista previa.



Por cada campo de la tabla puede elegir si el campo se importa como elemento o como atributo o si no se importa en absoluto. Para ello haga clic en el símbolo que aparece a la izquierda del nombre de la columna hasta que aparezca la opción deseada. Por ejemplo, en la imagen anterior el campo `city` se importará como elemento y el campo `zip` se importará como atributo.

Conversión de tipos de datos

En los Anexos 2233 de este manual puede encontrar información sobre la conversión de tipos de datos de BD en tipos de datos XML Schema.

28.11.3 Importar documento de Microsoft Word



Si usó estilos de párrafo en su **documento de Microsoft Word**, puede usar este comando para **importarlo** y pasarlo a formato XML de forma automática. Al hacer clic en este comando, aparece el cuadro de diálogo Abrir. Busque y seleccione el documento Word que desea importar. XMLSpy genera automáticamente un documento XML con estilos CSS incluidos. Cada párrafo de Word genera un elemento XML, cuyo nombre se toma del estilo de párrafo de Microsoft Word.

© 2016-2022 Altova GmbH

28.11.4 Crear esquema XML a partir de la estructura de la BD

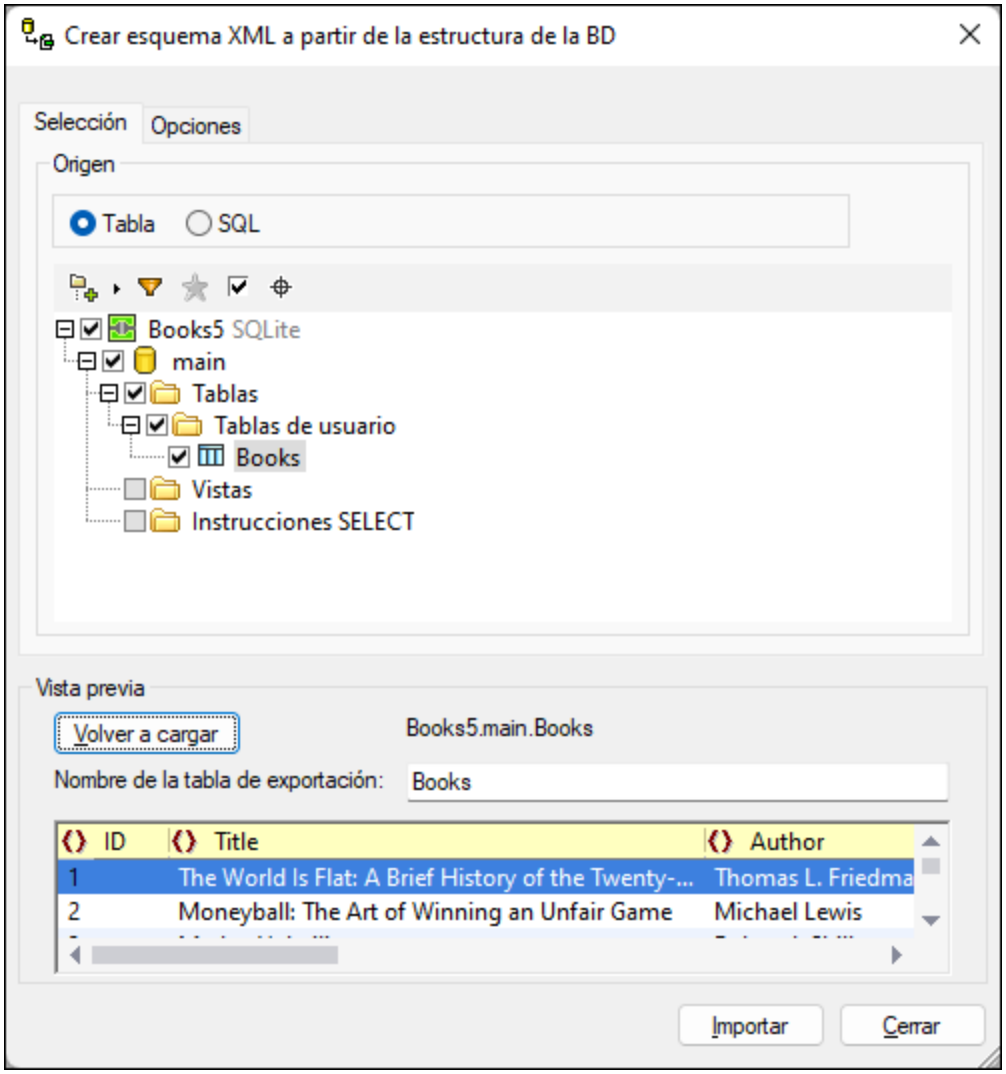


El comando **Crear esquema XML a partir de la estructura de la BD** sirve para crear un esquema XML basándose en la estructura de una BD. El mecanismo de creación del esquema XML tiene dos fases:

1. Establecer la conexión a la base de datos.
2. Configurar las opciones de selección de datos de la BD y seleccionar el esquema XML.

Seleccionar la estructura de datos que quiere importar

Una vez establecida la conexión, aparece el cuadro de diálogo "Crear esquema XML a partir de la estructura de la BD" (*imagen siguiente*), que incluye dos pestañas y varios paneles con opciones para seleccionar la estructura que desea importar. Estas opciones se describen más abajo. Cuando termine de configurar la importación, haga clic en el botón **Importar**.



Pestaña Selección

En la pestaña *Selección* (*imagen anterior*), el panel *Origen* (*imagen siguiente*) muestra una representación en forma de árbol de las tablas de la BD (botón de opción *Tabla*) o una instrucción SQL editable para seleccionar las tablas de la BD (botón de opción *SQL*).

© 2016-2022 Altova GmbH



Opciones de selección de la vista Tabla

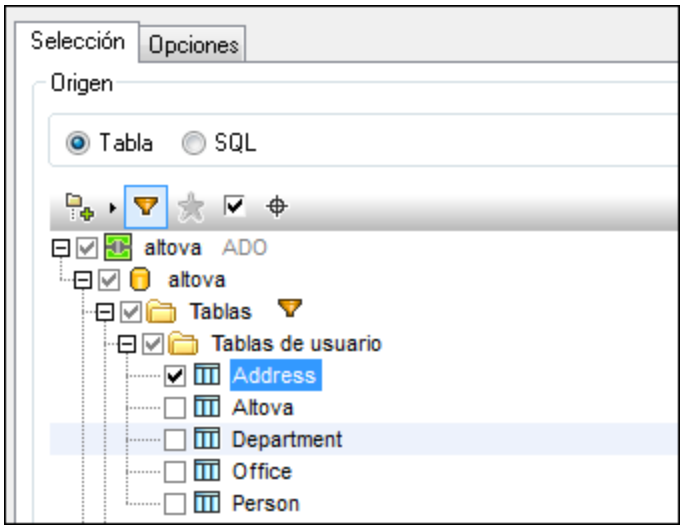
Si selecciona la vista *Tabla*, puede seleccionar las tablas de la BD que desea importar (*marcando sus casillas como en la imagen anterior*). Si hace clic en el nombre de una tabla, su contenido se muestra en el panel *Vista previa*, donde puede seguir seleccionando y filtrando los datos de la tabla (*véase más abajo*).

La vista de selección de tabla permite seleccionar comandos con los iconos de la barra de herramientas.



Estos son los iconos, de izquierda a derecha:

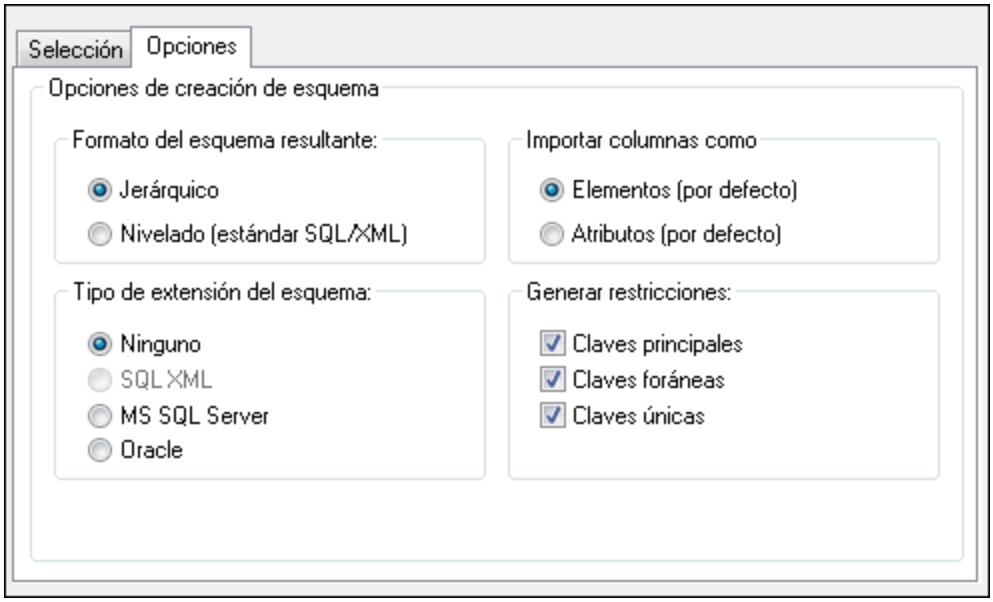
- *El icono Disposición:* sirve para organizar los objetos de la BD: (i) por *carpetas*, una por cada tipo de objeto; (ii) por carpetas pero *sin esquemas*; (iii) de forma jerárquica pero *sin carpetas* y (iv) según el tipo de tabla, en función de sus *dependencias con otras tablas*.
- *Filtrar contenido de carpeta:* aparece un icono en forma de embudo junto a las carpetas (*véase la carpeta Tablas de la imagen siguiente*). Al hacer clic en el icono en forma de embudo, aparece un menú con varias opciones de filtrado. En la imagen siguiente, por ejemplo, configuramos el filtro para ver solamente los objetos cuyo nombre contiene el texto `Sys`. Si hace clic en el icono aparece un menú que permite filtrar la información de distintas maneras.



- *Ver favoritos:* muestra solo los objetos marcados como favoritos.
- *Mostrar únicamente los objetos seleccionados*: muestra solamente los objetos seleccionados.
- *Localizador de objetos:* aparece un campo de texto que funciona como una función de búsqueda. Puede escribir una cadena de texto y aparece una lista desplegable con todos los objetos de la tabla cuyo nombre contiene dicha cadena de texto. Al seleccionar uno de los objetos en la lista desplegable, el objeto aparece resaltado en la vista en forma de árbol.

Pestaña Opciones

En la pestaña *Opciones* (*imagen siguiente*) puede especificar: (i) el formato del esquema resultante, (ii) su tipo de extensión, (iii) si las columnas se importan como elementos o como atributos y (iv) qué restricciones de la BD se generan en el esquema resultante.



Formato del esquema: puede elegir entre un formato nivelado (estándar SQL/XML) o un formato jerárquico:

 © 2016-2022 Altova GmbH

- El modelo de esquema **nivelado** se basa en la especificación INCITS/ISO/IEC 9075-14-2008. La especificación SQL/XML define cómo se asignan datos de BD a XML. Las relaciones se definen en esquemas por medio de restricciones de identidad y no hay referencias a los elementos. Por tanto, el esquema tiene una estructura plana o nivelada, similar a la vista en forma de árbol de la BD. Puede comprar la especificación en la tienda web de ANSI. Para más información visite la página www.iso.org.
- El modelo de esquema **jerárquico** representa de forma gráfica las dependencias entre las tablas en una vista en forma de árbol. En esta vista, las tablas con dependencias se muestran como elementos secundarios en el modelo de contenido. Las dependencias entre las tablas también se muestran en la pestaña *Restricciones de identidad*.
  En el esquema las tablas aparecen como elementos globales y las columnas son sus elementos o atributos. (El usuario decide si las columnas se importan como elementos o atributos).
  Las relaciones se crean de forma jerárquica, de modo que el campo de clave foránea de una tabla es de hecho una referencia al elemento global que representa dicha tabla.

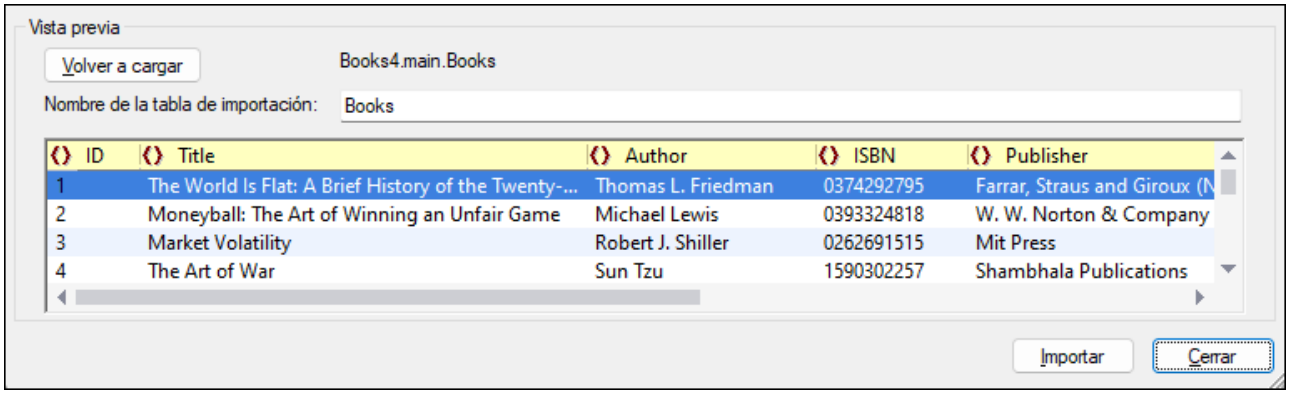
Tipo de extensión del esquema: la información de extensión del esquema es información adicional que se lee en la BD y que se incrusta en el esquema como datos de anotación o como atributos. Puede elegir entre cuatro opciones: (i) sin información de la extensión, (ii) extensiones SQL/XML, (iii) extensiones MS SQL Server y (iv) extensiones Oracle.

- **Ninguno**: la base de datos no aporta información adicional.
- **SQL XML:** las extensiones SQL/XML solamente se insertan cuando se generan esquemas en formato **nivelado**. La información de la extensión se almacena en anotaciones y se describe en la especificación SQL/XML (INCITS/ISO/IEC 9075-14-2008).
- **MS SQL Server:** se generan extensiones SQL Server. Consulte SQL Server Books Online y la página de MSDN sobre la anotación de esquemas XSD. En el esquema se generan estos elementos de anotación: `sql:relation`, `sql:field`, `sql:datatype`, `sql:mapped`.
- **Oracle:** esta opción está seleccionada por defecto cuando se trabaja con una base de datos Oracle. La información de la BD se almacena como atributos. Para más información consulte la documentación en línea de Oracle. En el esquema se genera este subconjunto de atributos: `SQLName`, `SQLType`, `SQLSchema`.

Nota: las extensiones SQL Server y Oracle se pueden generar para bases de datos SQL Server y Oracle, pero también para otras BD. Esto es muy práctico si está trabajando con otro tipo de BD y quiere generar un esquema para usarlo más tarde con SQL Server u Oracle.

Panel Vista previa

El panel *Vista previa* (*imagen siguiente*) muestra la estructura de la tabla que está seleccionada en la pestaña *Selección*. Cuando seleccione una tabla nueva en la pestaña *Selección*, haga clic en el botón **Volver a cargar** de la vista previa.



Por cada campo de la tabla puede elegir si el campo se importa como elemento o como atributo o si no se importa en absoluto. Para ello haga clic en el símbolo que aparece a la izquierda del nombre de la columna hasta que aparezca la opción deseada. Por ejemplo, en la imagen anterior el campo `city` se importará como elemento y el campo `zip` se importará como atributo.

Conversión de tipo de datos

En los Anexos[2233] de este manual puede encontrar información sobre la conversión de tipos de datos de BD en tipos de datos XML Schema.

28.11.5 Importación de BD basada en el esquema XML



El comando **Importación de BD basada en el esquema XML** sirve para crear un documento XML a partir de datos de BD que sea válido con respecto a un esquema XML determinado. Esta función es compatible con estas bases de datos:

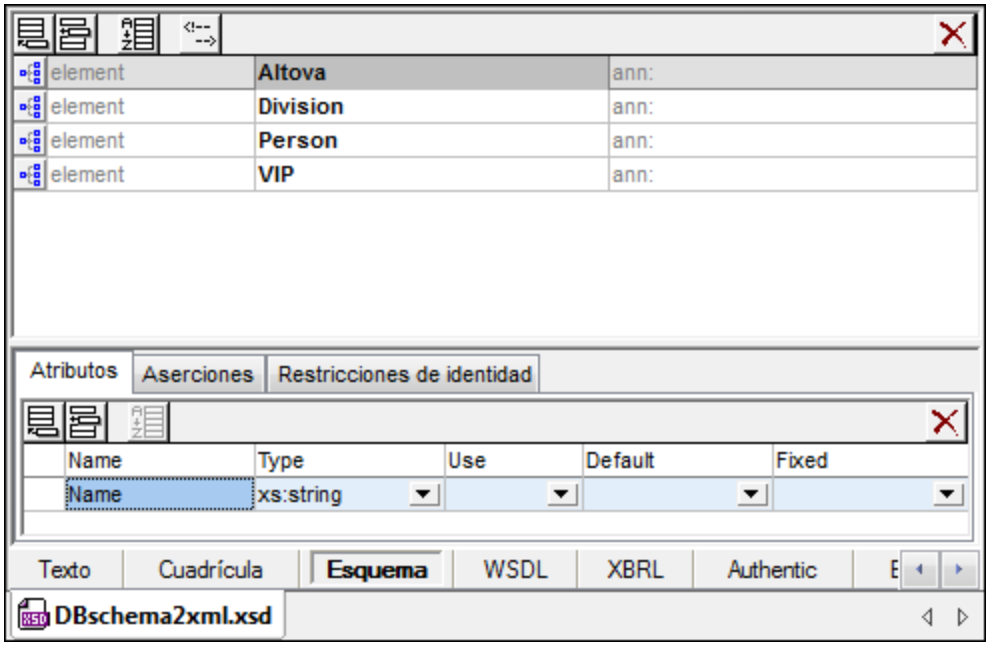
- Microsoft Access 2000 y 2003
- Microsoft SQL Server
- Oracle
- MySQL
- Sybase
- IBM DB2

Los datos que se importan al documento XML son los de la tabla que seleccione en la BD. Para empezar debe abrir en la vista Esquema el esquema XML en el que desea basar la importación. Después se establece la conexión a la BD, se seleccionan las tablas de importación y para terminar se hace clic en **Importar**. Los datos se importan a un documento XML y el documento XML tiene la estructura del esquema XML que estaba activo cuando se importaron los datos.

En el ejemplo siguiente, se importaron los datos de una base de datos MS Access. Este sería el proceso paso a paso:

 © 2016-2022 Altova GmbH

1. Abra el archivo de esquema en la vista Esquema (*imagen siguiente*).



2. Seleccione el comando de menú **Importación de BD basada en el esquema XML**. Aparece el cuadro de diálogo "Conectarse a un origen de datos".
3. Seleccione la opción *Microsoft Access (ADO)* y haga clic en **Siguiente**.
4. Haga clic en **Examinar** y seleccione el archivo de BD. Después haga clic en **Siguiente**.

5. Aparece el cuadro de diálogo "Importación de BD basada en el esquema XML". En la pestaña *Tablas*, seleccione las tablas que desea importar y haga clic en **Importar**. Las tablas seleccionadas se importan a un documento XML que se abre en la vista Cuadrícula.

Conversión de tipo de datos

En los Anexos 2233 de este manual puede encontrar información sobre la conversión de tipos de datos de BD en tipos de datos XML Schema.

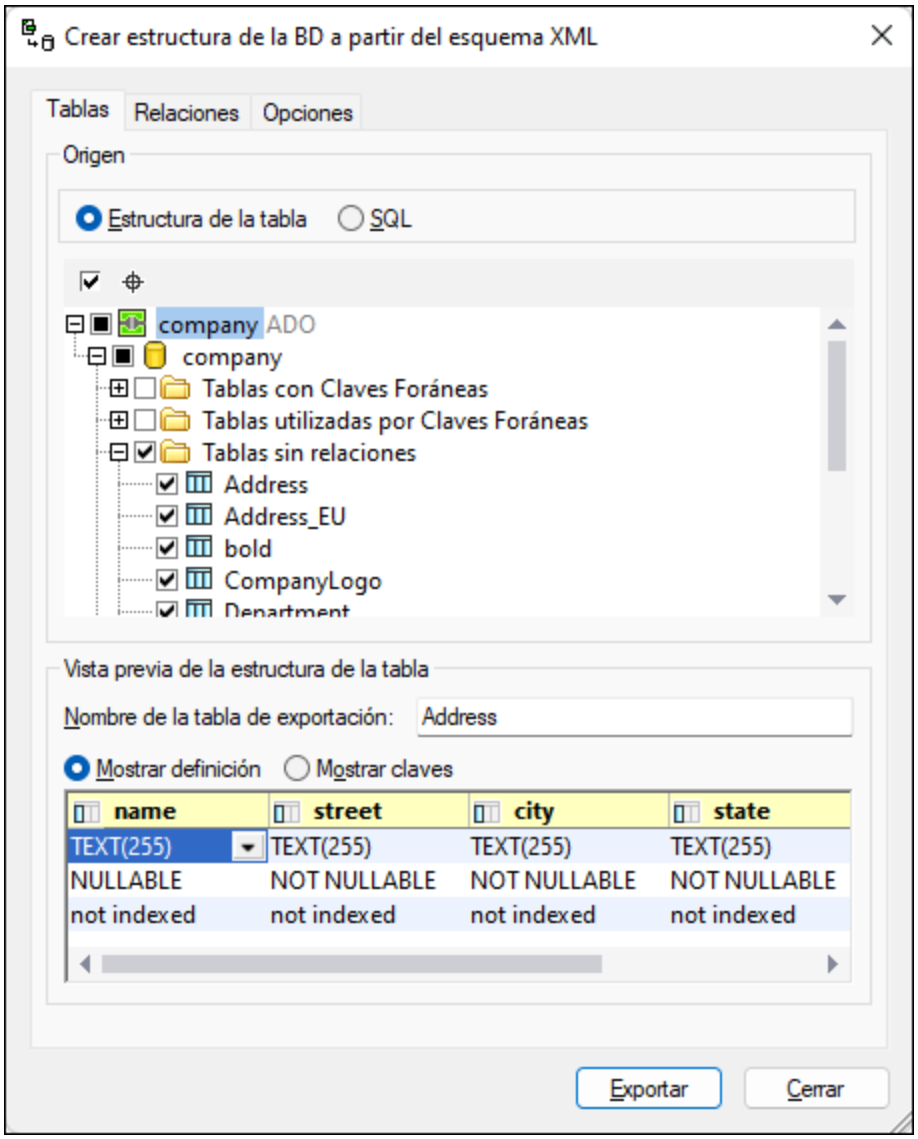
28.11.6 Crear estructura de la BD a partir del esquema XML



En XMLSpy puede crear una BD vacía (o un esqueleto de BD) basándose en un archivo de esquema. A continuación describimos el proceso, que suele ser igual para todos los tipos de BD.

1. Abra el archivo de esquema en la vista Esquema/WSDL
2. Seleccione el comando de menú **Conversión | Crear estructura de la BD a partir del esquema XML**. Aparece el cuadro de diálogo Conectarse a un origen de datos 953.

3. Siga las instrucciones del apartado Conectarse a un origen de datos 953 para conectarse a la BD que desea utilizar. Por ejemplo, para conectarse a una base de datos Microsoft Access, elija el botón de opción *Microsoft Access*. Puede usar una BD ya existente o crear una BD nueva.
4. Las tablas que se crean a partir del esquema se representan en forma de árbol en el diálogo "Crear estructura de la BD a partir del esquema XML", en la posición que tendrán en la BD. Por ejemplo, en la imagen siguiente, se creó y se seleccionó la tabla `Address`. Desactive la casilla de las tablas que no desee exportar.

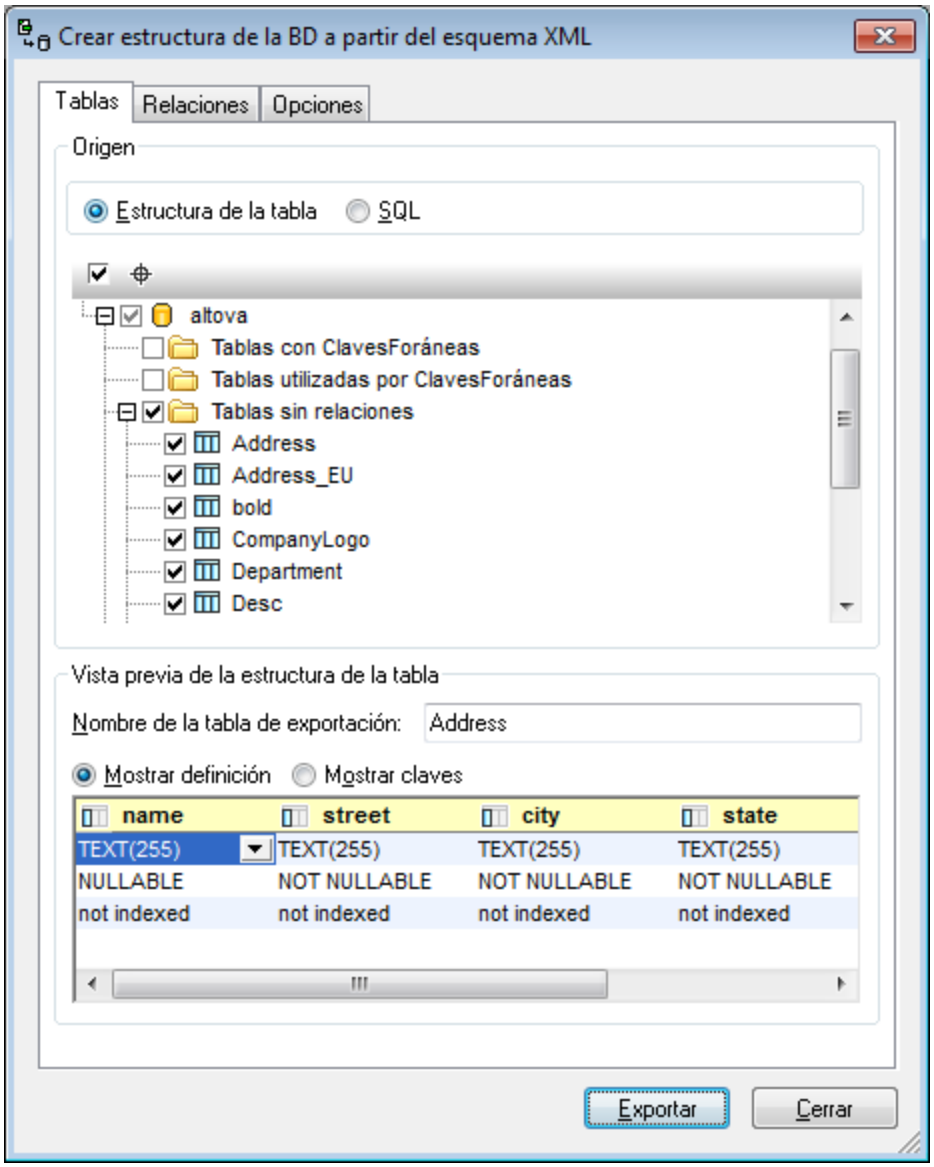


Crear tablas de BD con relaciones

Si el esquema XML en el que se basará la estructura de la BD tiene relaciones definidas en forma de restricciones de identidad, estas relaciones se crean automáticamente en la estructura de BD generada y aparecen en la estructura de la tabla del cuadro de diálogo. Las tablas que tienen relaciones se dividen en dos categorías: *Tablas con ClavesForáneas* y *Tablas utilizadas por ClavesForáneas*. Las tablas que no tienen relaciones aparecen en el nodo *Tablas sin relaciones*.

© 2016-2022 Altova GmbH

En la pestaña *Relaciones* puede crear y modificar las relaciones de las tablas. En esta pestaña se enumeran todas las relaciones de clave principal/clave foránea (*imagen siguiente*).



Para crear una relación:

1. Seleccione una de las relaciones de clave principal/clave foránea.
2. En el panel inferior del cuadro de diálogo, haga clic en el botón + para crear una relación.
3. Seleccione las columnas de cada tabla en las respectivas listas desplegables.

También puede eliminar relaciones: seleccione la relación y haga clic en **Eliminar**.

Notas sobre la estructura de bases de datos y la conexión con bases de datos

La estructura del esquema viene definida por las restricciones de identidad y se refleja en la BD resultante. La tabla que aparece a continuación muestra el tipo de BD que se crea con el comando **Crear estructura de la BD a partir del esquema XML**, sus restricciones y los métodos de conexión.

	Conexión directa	Con ODBC	Con ADO
MS Access (2000 y 2003)	Sí *	Sí	Sí
MS SQL Server	Sí *	Sí	Sí
Oracle	Sí *	Sí	Sí
MySQL	-	Sí *	Sí +
Sybase	-	Sí *	Sí
IBM DB2	-	Sí *	Sí

* *Método de conexión recomendado*
\+ *MySQL: cuando se cree una conexión ADO basada en ODBC, se recomienda usar el DSN de usuario o de sistema.*
\- *Incompatible*

XMLSpy realiza asignaciones de datos tanto con esquemas de formato jerárquico como con esquemas nivelados[1461]. XMLSpy reconoce ambos formatos automáticamente. El formato nivelado se asigna a SQL de dos formas diferentes.

- Bases de datos SQL Server, Oracle o Sybase:
  En un esquema que se generó en formato nivelado para una de estas BD se extrae el nombre del catálogo del esquema y se usa en el script SQL generado como nombre de la BD. Esto significa que el script SQL resultante se ejecuta en una BD de destino cuyo nombre debe ser idéntico al nombre del catálogo del esquema.
- Bases de datos Access (2000 o 2003), MySQL o DB2:
  Un esquema que se generó en formato nivelado para una de estas BD **omitirá** el nombre del catálogo del esquema cuando se genere el script SQL. Esto significa que el script SQL resultante se ejecuta en una BD de destino en la que el usuario inició sesión.

Conversión de tipo de datos

En los Anexos[2233] de este manual puede encontrar información sobre la conversión de tipos de datos de BD en tipos de datos XML Schema.

© 2016-2022 Altova GmbH

28.11.7 Exportar a archivos de texto



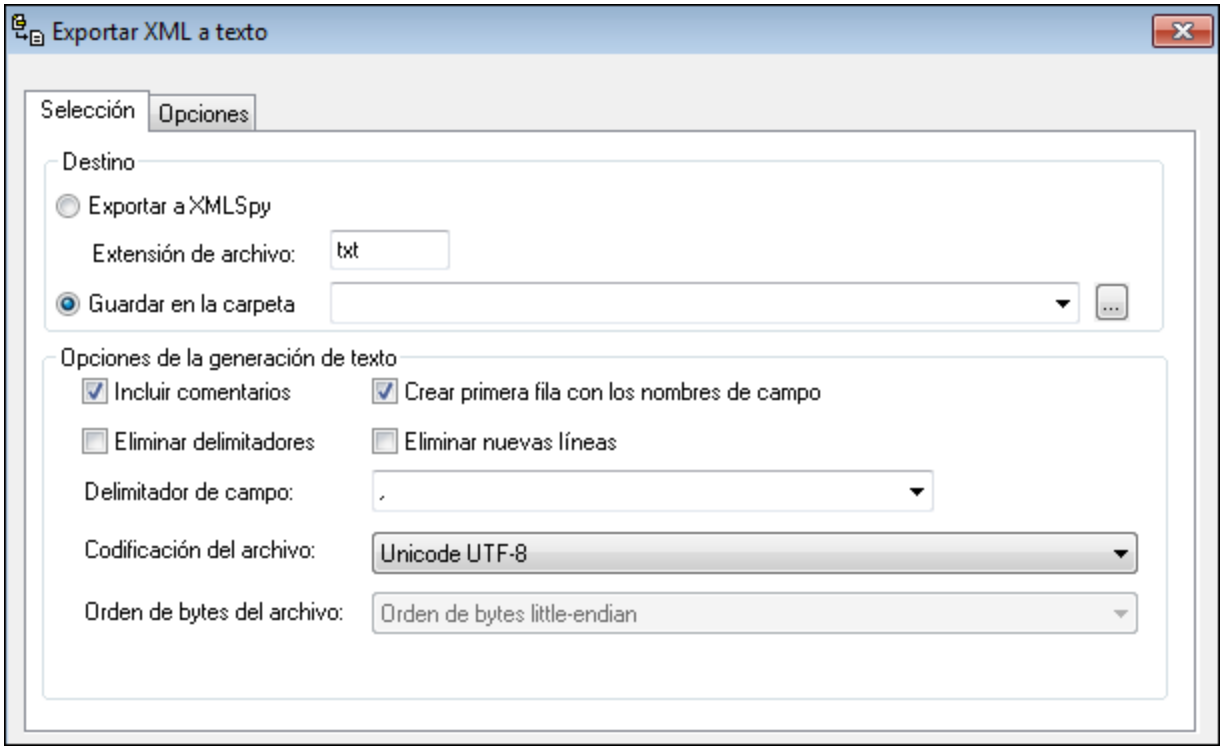
El comando **Conversión | Exportar a archivos de texto** sirve para pasar un archivo XML a formato texto y poder intercambiar los datos XML con bases de datos y otros sistemas heredados. Al hacer clic en este comando se abre el cuadro de diálogo "Exportar XML a texto". El diálogo tiene dos partes:

- la parte superior, que a su vez se divide en dos pestañas: (i) *Selección* y (ii) *Opciones*.
- la parte inferior, una ventana de vista previa.

Cuando termine de seleccionar datos y de configurar las opciones de exportación en este cuadro de diálogo (tal y como se describe más abajo) haga clic en el botón **Exportar** para exportar los datos a archivos de texto.

Selección

En la pestaña *Selección* (*imagen siguiente*) puede seleccionar el destino de los archivos de texto de salida y las opciones de generación de texto.



Destino: puede seleccionar en qué carpeta se guardan los archivos de texto y con qué extensión de archivo. El nombre de los archivos equivale al nombre del elemento (del archivo XML) que se exportó al archivo de texto. También puede generar archivos de texto sin nombre y exportarlos a XMLSpy, desde donde puede guardarlos con el nombre y en la ubicación que quiera.

Incluir comentarios: active esta opción para incluir un comentario generado por XMLSpy en el archivo exportado. El comentario contendrá la consulta SQL utilizada para seleccionar los datos, así como una lista con un elemento de lista por cada encabezado de columnas de la tabla de la BD.

Crear primera fila con los nombres de campo: si marca esta casilla, las tablas exportadas incluyen los nombres de las columnas de la BD. Si no marca esta casilla, los nombres de las columnas no se incluyen en el archivo de texto exportado.

Eliminar delimitadores: elimina de los datos exportados los delimitadores de los valores de texto. Seleccione qué delimitador desea eliminar en el cuadro combinado *Delimitador de campo*. Por ejemplo, si activa esta opción y selecciona el delimitador ' (apóstrofo), cuando exporte el valor XML `Ba'ker` en el texto de exportación la cadena de texto será `Baker`.

Eliminar nuevas líneas: elimina las líneas nuevas en los datos exportados.

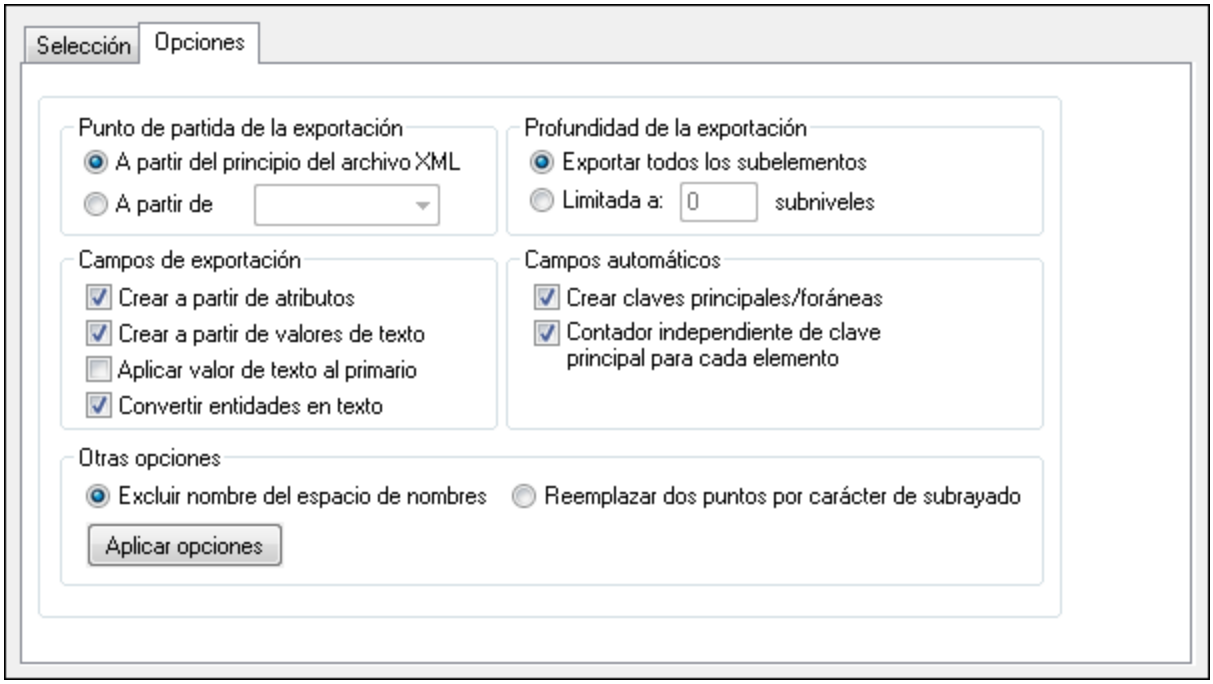
Delimitador de campo: seleccione el carácter de delimitación que desea eliminar durante la exportación. También puede escribir un carácter nuevo en este campo.

Codificación del archivo: seleccione la codificación que desea usar para los archivos de exportación.

Orden de bytes: si exporta archivos Unicode de 16 o 32 bits (UCS-2, UTF-16 o UCS-4), también puede elegir entre un orden de bytes little-endian o big-endian.

Opciones

En la pestaña *Opciones* puede configurar más opciones de exportación (*imagen siguiente*):



Punto de partida de la exportación: puede exportar todo el documento XML o solamente una parte, a partir de un elemento determinado. También puede especificar cuántos subniveles del elemento seleccionado se deben incluir en la exportación (opción Profundidad de la exportación).

© 2016-2022 Altova GmbH

Profundidad de la exportación: aquí puede especificar cuántos subniveles del elemento seleccionado como punto de partida de la exportación se deben exportar.

Campos de exportación: dependiendo de cómo sean sus datos XML, quizás quiera exportar solamente los elementos, los atributos o el contenido de texto de los elementos. En la Vista previa también puede activar/desactivar uno por uno los elementos que se exportan y seleccionar cómo se exportan (como elementos o como atributos).

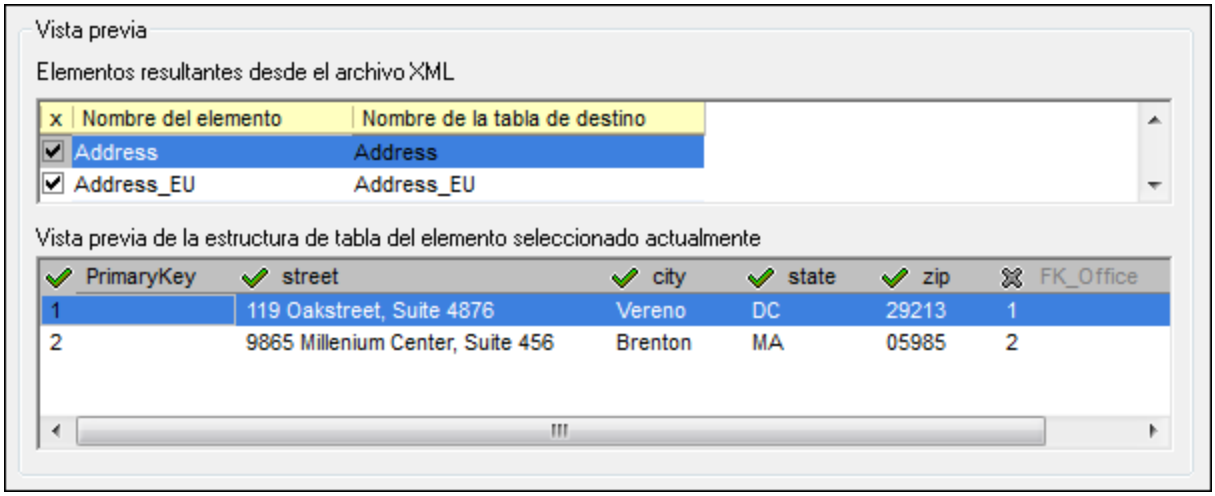
Campos automáticos: XMLSpy genera un archivo o tabla de salida por cada tipo de elemento seleccionado. Puede elegir si se crean pares de claves principales/foráneas para enlazar sus datos en el modelo relacional y definir una clave principal para cada elemento.

Excluir nombre del espacio de nombres: por último, puede elegir entre excluir el prefijo de espacio de nombres de los elementos y atributos (opción Excluir nombre del espacio de nombres) o reemplazar los dos puntos (:) del prefijo de espacio de nombres con un carácter de subrayado (opción Reemplazar dos puntos por carácter de subrayado)

Aplicar opciones: cuando termine de configurar las opciones, haga clic en este botón para aplicarlas. La vista previa se actualiza en base a las nuevas opciones.

Vista previa

El panel *Vista previa* (*imagen siguiente*) está situado en la parte inferior de las pestañas *Selección* y *Opciones*.



El panel *Elementos resultantes desde el archivo XML* muestra el nombre de los nodos que se exportarán y el nombre que recibirán en el archivo generado. Aquí puede seleccionar qué nodos se exportan (marcando su casilla). Cuando seleccione un nodo, en el panel inferior aparece una vista previa de su estructura. En este panel puede seleccionar qué columnas se exportan, haciendo clic en el icono situado a la izquierda del nombre del encabezado de la columna. Por ejemplo, en la imagen anterior, se desactivó la última columna (`FK_Office`).

28.11.8 Exportar a una base de datos



El comando **Conversión | Exportar a una base de datos** sirve para exportar datos XML a una base de datos. Al hacer clic en este comando, se inicia el asistente para la conexión a bases de datos, que le ayudará a conectarse a la BD que desea actualizar. Una vez establecida la conexión con la BD, aparece el cuadro de diálogo "Exportar datos XML a una base de datos". Este cuadro de diálogo tiene dos partes:

- la parte superior, que a su vez se divide en dos pestañas: (i) *Selección* y (ii) *Opciones*.
- la parte inferior, una ventana de vista previa.

Cuando termine de configurar las opciones de este cuadro de diálogo (tal y como se describe más abajo), haga clic en el botón **Exportar** para exportar los datos a la BD.

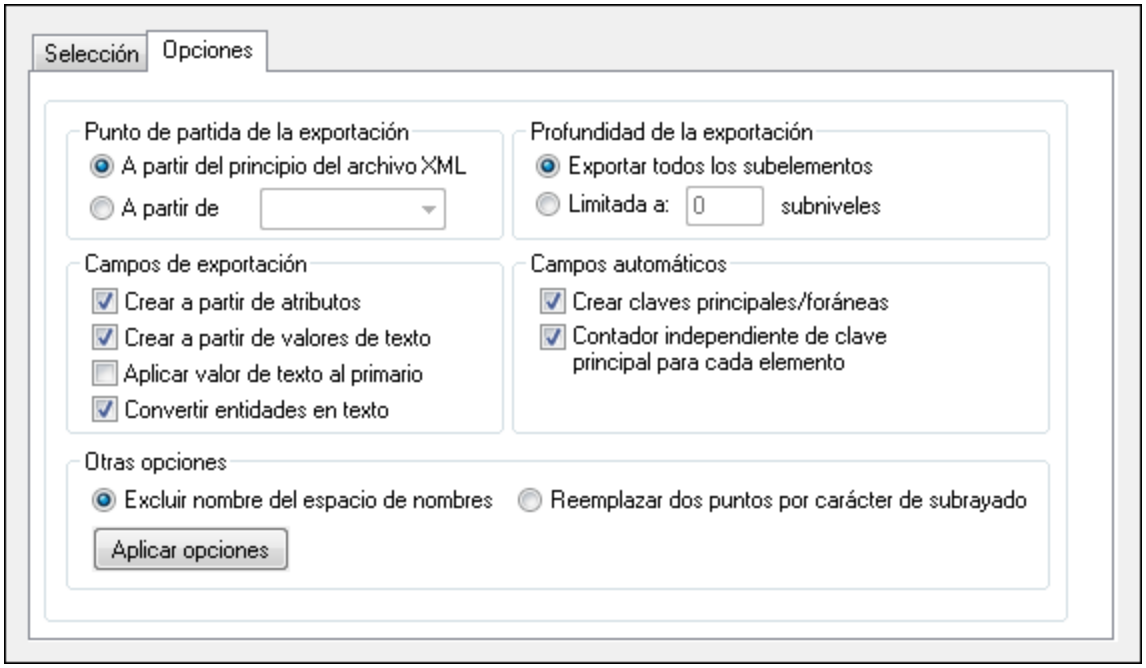
Selección

En la pestaña *Selección* puede seleccionar la BD de destino y las opciones de generación de tablas. En el panel *Destino* puede seleccionar la conexión a la BD. En las *Opciones de generación de base de datos*, debe elegir una de estas tres opciones: (i) si los datos se crean como tablas nuevas, (i) si los datos deben actualizar tablas ya existentes o (iii) si primero se debe intentar actualizar una tabla ya existente y, si no se puede actualizar, entonces se crea una tabla nueva. También puede definir una acción de detención en función de un número determinado de errores. Por último, puede crear un registro de scripts SQL.

Opciones

En la pestaña *Opciones* puede configurar las opciones de exportación (*imagen siguiente*):

© 2016-2022 Altova GmbH



Punto de partida de la exportación: puede exportar todo el documento XML o solamente una parte, a partir de un elemento determinado. También puede especificar cuántos subniveles del elemento seleccionado se deben incluir en la exportación (opción Profundidad de la exportación).

Profundidad de la exportación: aquí puede especificar cuántos subniveles del elemento seleccionado como punto de partida de la exportación se deben exportar.

Campos de exportación: dependiendo de cómo sean sus datos XML, quizás quiera exportar solamente los elementos, los atributos o el contenido de texto de los elementos. En la Vista previa también puede activar/desactivar uno por uno los elementos que se exportan y seleccionar cómo se exportan (como elementos o como atributos).

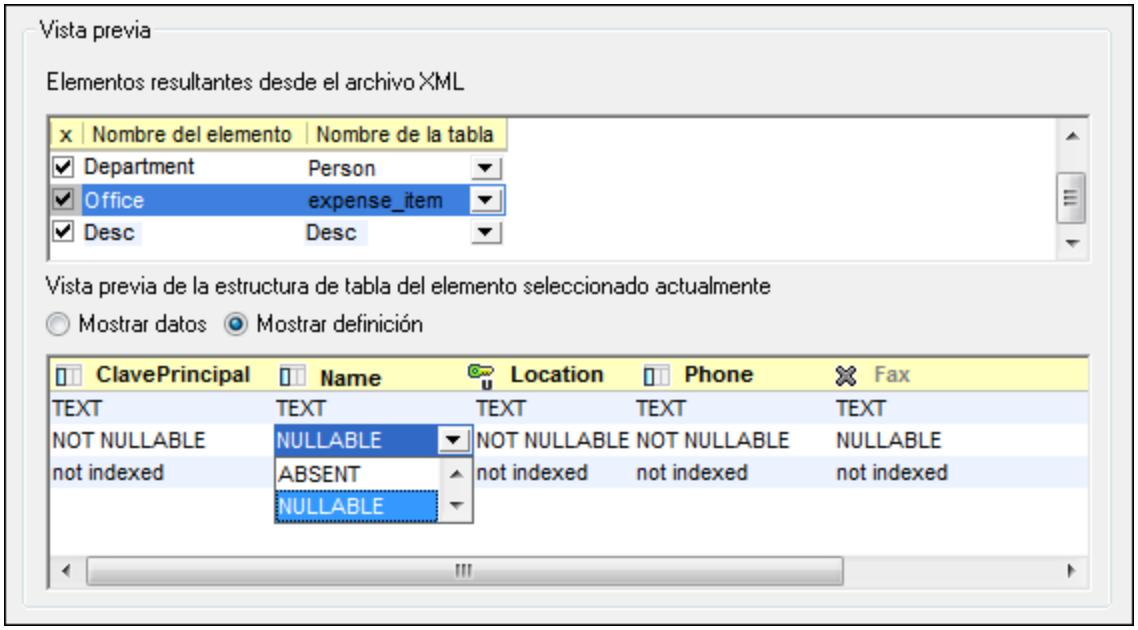
Campos automáticos: XMLSpy genera un archivo o tabla de salida por cada tipo de elemento seleccionado. Puede elegir si se crean pares de claves principales/foráneas para enlazar sus datos en el modelo relacional y definir una clave principal para cada elemento.

Excluir nombre del espacio de nombres: por último, puede elegir entre excluir el prefijo de espacio de nombres de los elementos y atributos (opción Excluir nombre del espacio de nombres) o reemplazar los dos puntos (:) del prefijo de espacio de nombres con un carácter de subrayado (opción Reemplazar dos puntos por carácter de subrayado)

Aplicar opciones: cuando termine de configurar las opciones, haga clic en este botón para aplicarlas. La vista previa se actualiza en base a las nuevas opciones.

Vista previa

El panel de vista previa (*imagen siguiente*) aparece en la parte inferior de las pestañas *Selección* y *Opciones*.



El panel *Elementos resultantes desde el archivo XML* muestra el nombre de los nodos del documento XML que se exportarán y el nombre que recibirán en la tabla generada. Aquí puede seleccionar qué nodos se exportan (marcando su casilla). Cuando seleccione un elemento, en el panel inferior aparece una vista previa de la estructura que tendrá en el archivo generado. En este panel puede elegir entre una vista previa (i) de los datos de la estructura generada (*Mostrar datos*) o (ii) de las definiciones de cada columna en la estructura generada (*Mostrar definición*). La imagen anterior muestra las definiciones de las columnas.

En este segundo panel, puede cambiar la configuración de cada columna, haciendo clic en el icono situado a la izquierda de su nombre. Hay cuatro configuraciones posibles: (i) Incluir en la estructura de la tabla, (ii) Restricción única, (iii) Restricción de clave principal y (iv) Excluir de la estructura de la tabla. En la imagen anterior, por ejemplo, la columna `Location` tiene una restricción única, mientras que la columna `Fax` se excluyó de la estructura de la tabla. Las demás columnas se incluyen en la estructura de la tabla.

Cuando la estructura de la tabla del elemento muestra las definiciones de los campos (*Mostrar definición*), las definiciones se pueden editar seleccionando la definición y seleccionando una opción de la lista desplegable del cuadro combinado (*imagen anterior*).

28.11.9 Convertir instancia XML en instancia JSON y viceversa

Si el documento activo es un documento XML, este comando genera un documento JSON a partir del documento activo. Si, por el contrario, el documento activo es un archivo JSON, este comando genera un documento XML a partir del documento activo. El documento resultante se abre inmediatamente en una ventana nueva y puede guardarse en cualquier ubicación del sistema de archivos.

Más abajo describimos las opciones de conversión. Además, puede encontrar más información sobre las funciones de edición de datos JSON en la sección JSON y esquemas JSON 665.

 © 2016-2022 Altova GmbH

Ejemplos de conversión

A continuación puede ver un documento XML de partida y el documento JSON de salida que se obtiene con el comando **Convertir instancia XML en instancia JSON y viceversa**.

Documento XML (documento de partida)

```
<?xml version="1.0" encoding="UTF-8"?>
<Person first="Jim" last="James">
  <Address>
    <street>4 New Street</street>
    <city>New York</city>
    <state>NY</state>
    <code>10123</code>
  </Address>
  <Tel type="home">
      123 123-1234
  </Tel>
  <Tel type="office">
      123 987-9876
  </Tel>
</Person>
```

Documento JSON (documento de salida)

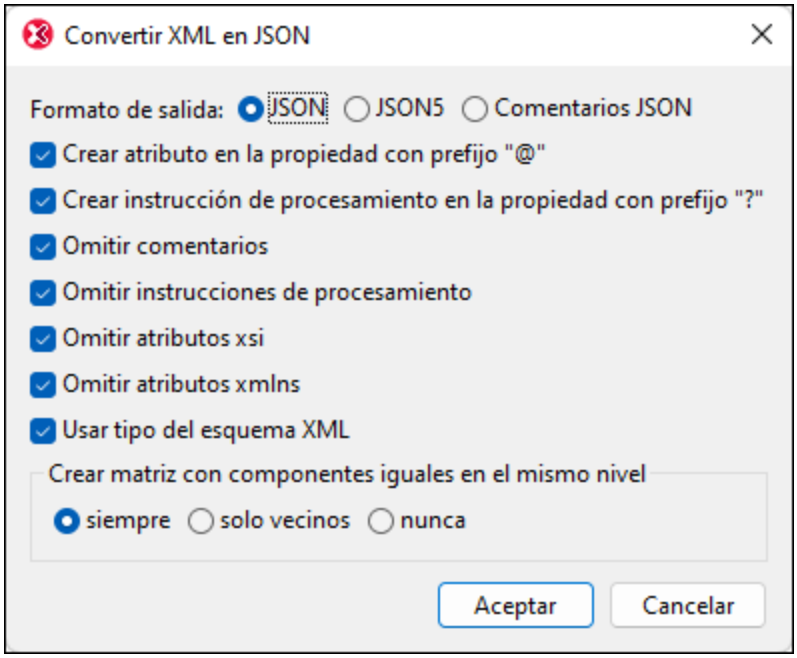
```
{
       "XML": {
              "version": 1.0,
              "encoding": "UTF-8"
       },
       "Person": {
              "first": "Jim",
              "last": "James",
              "Address": {
                     "street": "4 New Street",
                     "city": "New York",
                     "state": "NY",
                     "code": 10123
              },
              "Tel": [ { "type": "home",
                     "Text": "\r      123 123-1234\r  "}, { "type": "office",
                     "Text": "\r      123 987-9876\r  "} ]
       }
}
```

Para convertir un documento JSON en XML, el documento JSON debe estar activo en la vista antes de que se ejecute el comando **Convertir instancia XML en instancia JSON y viceversa**.

Opciones de conversión de datos XML en JSON

Cuando se invoca el comando **Convertir instancia XML en instancia JSON y viceversa** para convertir un documento XML en un documento JSON, aparece el cuadro de diálogo "Convertir XML en JSON" (*imagen*

siguiente). Puede seleccionar si el destino de la conversión es el formato JSON, JSON5 o JSON Comments (*Comentarios JSON*). Después configure las opciones de conversión y haga clic en **Aceptar**. XMLSpy generará un documento de instancia JSON a partir de la instancia XML que se abrirá inmediatamente en una ventana nueva.



Con la dos primeras opciones puede indicar si se añaden prefijos o no a los nombres de propiedad JSON para evitar conflictos con elementos que estén en el mismo nivel. Esto puede verse bien en el ejemplo de código que aparece a continuación. Observe que el *atributo* XML `somenode` se convirtió en la propiedad JSON `@somenode`. Gracias al prefijo `@` se evita un conflicto con la propiedad JSON `somenode` (generada a partir del *elemento* XML `somenode`).

Instancia XML

```
<root somenode="value">
    <somenode>content</somenode>
</root>
```

Instancia JSON

```
{
    "root": {
        "@somenode": "value",
        "somenode": "content"
    }
}
```

Las demás opciones del cuadro de diálogo sirven para especificar si determinados tipos de nodos XML se convierten o no. Si se convierten, los comentarios XML reciben el nombre `"#"`. Los nodos de texto (que suelen aparecer dentro de elementos con contenido mixto) reciben el nombre `"$"`. Si un nodo XML tiene un prefijo de

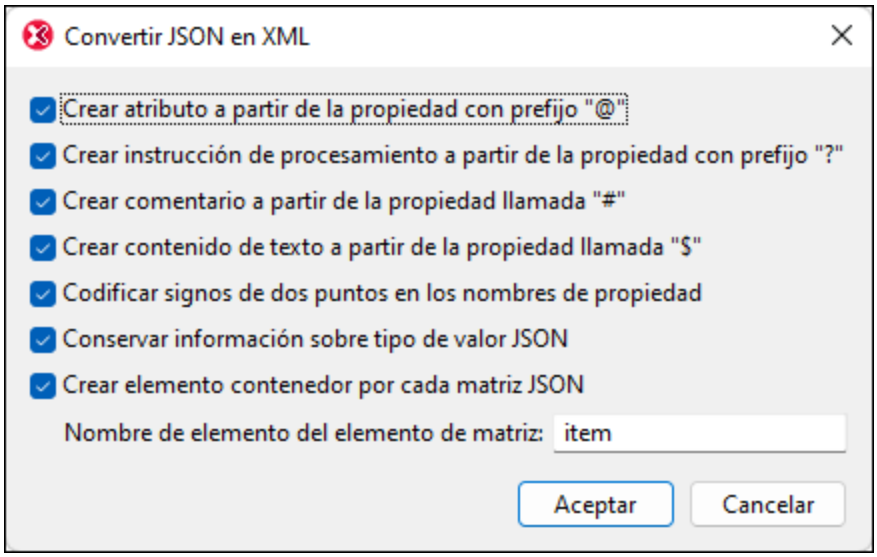
© 2016-2022 Altova GmbH

espacio de nombres, el nombre JSON correspondiente se creará con este prefijo de espacio de nombres. Si en el mismo nivel hay elementos con nombres idénticos, se considerará que son componentes iguales. Asimismo, los nodos como comentarios, instrucciones de procesamiento y text() situados en el mismo nivel se consideran componentes iguales. Si en el mismo nivel hay componentes iguales, podrá elegir si XMLSpy crea una matriz con ellos o no. Es decir, en el cuadro de diálogo puede indicar si se crea una matriz con todos los componentes iguales, si se crea con los componentes iguales que sean vecinos o si no se crea ninguna matriz.

La opción *Usar tipo del esquema XML* permite hacer conversiones basándose en el tipo del esquema XML del nodo de origen. Por ejemplo, si un nodo está definido como de tipo `xs:string`, entonces la propiedad destino del objeto JSON será de tipo JSON `string` y estará entre comillas. Esta opción es útil si, por ejemplo, un número está almacenado como cadena en el nodo XML de origen y la conversión en JSON debe ser fiel en cuanto a los tipos.

Opciones de conversión de datos JSON en XML

Cuando se invoca el comando **Convertir instancia XML en instancia JSON y viceversa** para convertir una instancia JSON en un documento XML, aparece el cuadro de diálogo "Convertir JSON en XML" (*imagen siguiente*). Configure aquí las opciones de conversión y haga clic en **Aceptar**. XMLSpy generará un documento de instancia XML a partir de la instancia JSON que se abrirá inmediatamente en una ventana nueva.



Debe tener en cuenta estos aspectos:

- Las propiedades de objetos JSON se convierten en elementos XML. Las primeras cuatro opciones de diálogo permiten elegir qué tipo de propiedades se convierten y cuáles no.
- *Codificar signos de dos puntos en los nombres de propiedad*: si marca esta casilla, los dos puntos de los nombres JSON se codifican y no se convierten como dos puntos. Si no marca esta casilla, los dos puntos se dejan como están.
- *Conservar información sobre tipo de valor JSON:* si marca esta casilla, la información de tipo JSON de las propiedades se convierte en un par atributo-valor del elemento correspondiente.
- *Crear elemento contenedor para cada matriz JSON:* en el documento XML de salida el elemento contenedor tendrá el nombre de la matriz JSON. Los elementos de la matriz JSON se convierten en elementos XML dentro de este contenedor y cada uno de los elementos recibe el nombre que se especifique en el cuadro de texto.

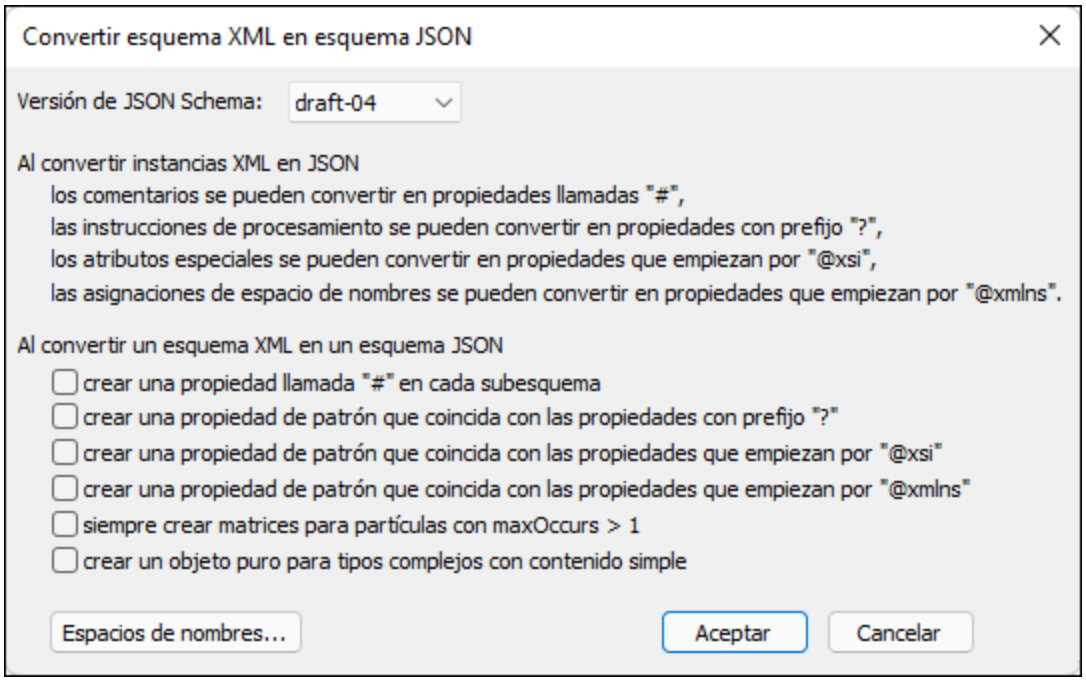
28.11.10 Convertir esquema XML en esquema JSON y viceversa

Si el documento activo es un esquema XML, este comando genera un esquema JSON a partir del documento activo. Si, por el contrario, el documento activo es un esquema JSON, este comando genera un esquema XML a partir del documento activo. El esquema resultante se abre inmediatamente en una ventana nueva y puede guardarse en cualquier ubicación del sistema de archivos. Más abajo describimos las opciones de conversión. Además, puede encontrar más información sobre las funciones de edición de datos JSON en la sección JSON y esquemas JSON[665].

Opciones de conversión de esquemas XML en esquemas JSON

Cuando se invoca el comando **Convertir esquema XML en esquema JSON y viceversa** para convertir un esquema XML en un esquema JSON, aparece el cuadro de diálogo "Convertir esquema XML en esquema JSON" (*imagen siguiente*). Configure aquí las opciones de conversión y haga clic en **Aceptar**. XMLSpy generará un esquema JSON a partir del esquema XML que se abrirá inmediatamente en una ventana nueva.

La estrategia general de conversión es: (i) los tipos simples del esquema XML se asignan a los tipos simples del esquema JSON (por ejemplo, las cadenas de texto y los números); (ii) los tipos complejos del esquema JSON se asignan a objetos JSON.



En la parte superior del cuadro de diálogo se explica cómo se convierten ciertos componentes del esquema XML. En la parte inferior del cuadro de diálogo puede configurar estas opciones:

- *Crear una propiedad llamada "#" en cada subesquema:* marque esta casilla para crear una propiedad llamada así en cada definición del esquema JSON.
- *Crear una propiedad de patrón que coincida con las propiedades con "?", "@xsi", "@xmlns":* con estas casillas puede crear propiedades de patrón por cada uno de estos prefijos y las propiedades de

 © 2016-2022 Altova GmbH

patrón coincidirán con los nombres que tengan estos prefijos. Para más información consulte el apartado Objetos y propiedades JSON[725].

- *Siempre crear matrices para partículas con maxOccurs > 1*: en los esquemas XML las partículas son los elementos de modelos de contenido complejo. Si el número de instancias es superior a 1, entonces las partículas se definen como matriz en el esquema JSON. De lo contrario se definen como propiedades de un objeto JSON.
- *Crear un objeto puro para tipos complejos con contenido simple*: el tipo complejo con contenido simple del esquema XML es un tipo que permite atributos y texto en el contenido, pero no elementos secundarios. Si marca la casilla *Crear un objeto puro*, entonces el tipo complejo se convierte en un objeto JSON. Los atributos del tipo se convierten en propiedades del objeto JSON en las propiedades cuyos nombres empiezan por `@`. Para el contenido de texto del tipo se genera una propiedad llamada `$`. Si no marca la casilla *Crear un objeto puro*, entonces el tipo complejo se convierte en un objeto que puede contener otros objetos, así como tipos JSON simples, como cadenas de texto y números.
- *Crear un esquema local para definiciones de tipos integrada:* si se selecciona, esta opción crea la definición del tipo dentro del objeto mismo. De lo contrario, la definición del tipo será una referencia a un objeto aparte. Puede ver los dos tipos de resultados en los fragmentos de código de esquema JSON que incluimos a continuación.

  Definición del tipo mediante referencia:

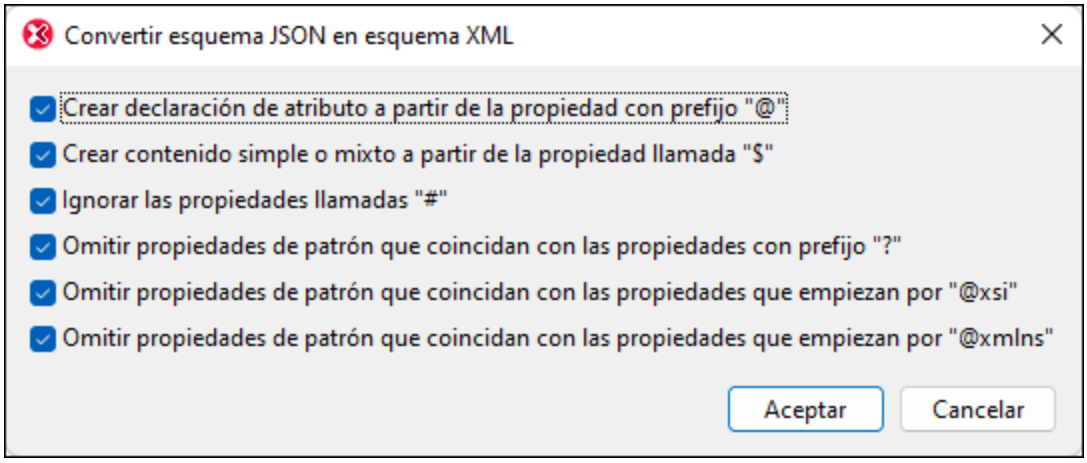
```
"properties": {
    "AccountManager": {
    "$ref": "#/definitions/xs:string"
    }
}
"xs:string": {
    "type": "string"
}
```

  Definición del tipo local:

```
"properties": {
    "AccountManager": {
    "type": "string"
    }
}
```

Opciones de conversión de esquemas JSON en esquemas XML

Cuando se invoca el comando **Convertir esquema XML en esquema JSON y viceversa** para convertir un esquema JSON en un esquema XML, aparece el cuadro de diálogo "Convertir esquema XML en esquema XML" (*imagen siguiente*). Configure aquí las opciones de conversión y haga clic en **Aceptar**. XMLSpy generará un esquema XML a partir del esquema JSON que se abrirá inmediatamente en una ventana nueva.



Aquí puede configurar estas opciones:

- Si se convierten o no las propiedades JSON cuyo nombre empieza por `'@'` y `'$'`. Estas propiedades se convierten en nodos de atributo y nodos de texto respectivamente.
- Si se convierten o no las propiedades que se llaman `'#'`. Estas propiedades se convierten en comentarios XML.
- Si se omiten o no las propiedades de patrón que coinciden con propiedades con el prefijo `'?'`, `"@xsi"` o `"@xmlns"`. Si no se omiten (es decir, si no se marcan estas casillas), las propiedades con prefijo `'?'`, `"@xsi"` y `"@xmlns"` se convierten en instrucciones de procesamiento, atributos con prefijo `xsi:` y atributos con prefijo `xmlns:` respectivamente.

28.11.11 Convertir en OIM xBRL-XML

Convierte los formatos de datos siguientes en OIM xBRL-XML.

- OIM xBRL-JSON
- OIM xBRL-CSV

El archivo que quiera convertir debe ser el archivo activo; después solo tiene que seleccionar el comando.

Se genera el archivo OIM xBRL-XML, que se abre en una ventana distinta y se valida. Después puede guardar el archivo generado en la ubicación que prefiera.

28.11.12 Convertir en OIM xBRL-JSON

Convierte los formatos de datos siguientes en OIM xBRL-JSON.

- Archivo de datos XBRL
- OIM xBRL-XML
- OIM xBRL-CSV

© 2016-2022 Altova GmbH

El archivo que quiera convertir debe ser el archivo activo; después solo tiene que seleccionar el comando.

Se genera el archivo OIM xBRL-JSON, que se abre en una ventana distinta y se valida. Después puede guardar el archivo generado en la ubicación que prefiera.

28.11.13 Convertir en OIM xBRL-CSV

Convierte los formatos de datos siguientes en OIM xBRL-CSV.

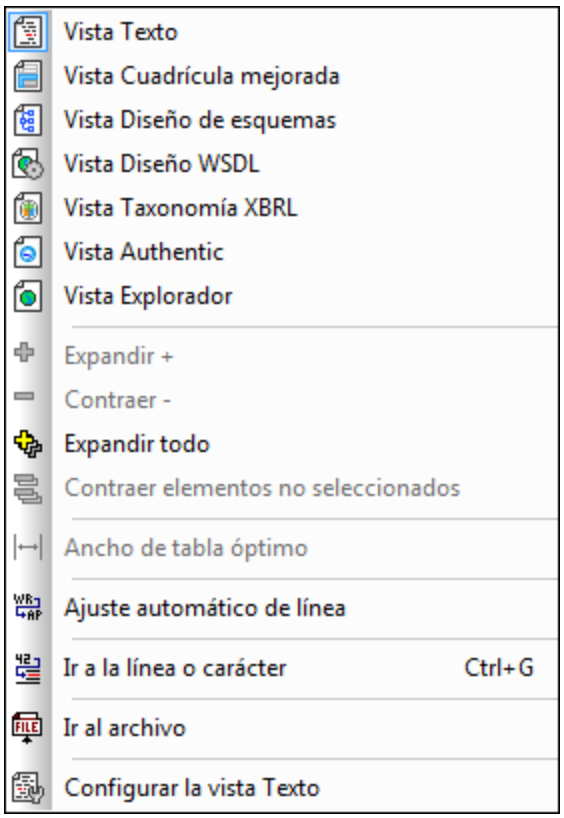
- Archivo de datos XBRL
- OIM xBRL-XML
- OIM xBRL-JSON

El archivo que quiera convertir debe ser el archivo activo; después solo tiene que seleccionar el comando.

La aplicación le pedirá que indique un nombre con el que guardar el archivo JSON. Este archivo JSON contiene las referencias de los archivo de datos CSV que se generan. Después de indicar el archivo JSON y hacer clic en Guardar se generan los archivos JSON y CSV y aparecen los resultados de la validación del archivo generado.

28.12 Menú Vista

El menú **Vista** (*ver imagen*) controla la visualización de la ventana principal[120] activa y permite definir la presentación del documento en pantalla.



28.12.1 Vista Texto



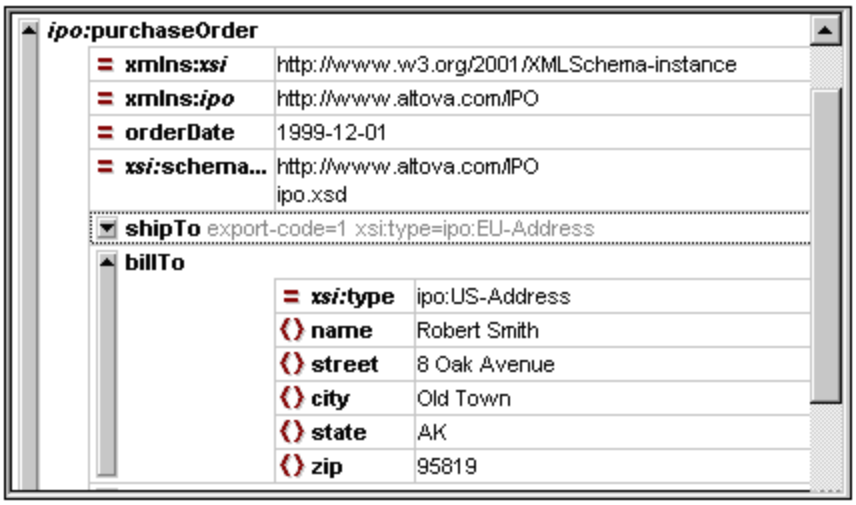
Este comando **cambia** la vista actual del documento a la vista Texto[145], donde puede editar el documento en formato texto. Esta vista ofrece avanzadas funciones de edición de texto, que se describen detalladamente en el apartado Vista Texto[145] del *Manual del usuario*.

Nota: en las secciones del cuadro de diálogo "Opciones" (**Herramientas | Opciones**[1583]) puede configurar varios aspectos de la vista Texto.

© 2016-2022 Altova GmbH

28.12.2 Vista Cuadrícula mejorada

Este comando **cambia** la vista actual del documento a la vista Cuadrícula[169]. Si la vista previa era la vista Texto[145], se comprueba automáticamente el formato XML del documento.



28.12.3 Vista Diseño de esquemas

Este comando **cambia** la vista actual del documento a la vista Esquema si el documento es un esquema XML. Esta vista se describe detalladamente en el apartado Vista Esquema[171] del *Manual del usuario*.

28.12.4 Vista Diseño WSDL

Este comando **cambia** la vista actual del documento a la vista WSDL, si el documento es un archivo WSDL (con la extensión `.wsdl`). Esta vista se describe detalladamente en el apartado Vista WSDL[251] del *Manual del usuario*.

28.12.5 Vista Taxonomía XBRL



Este comando **cambia** la vista actual del documento a la vista Taxonomía XBRL, si el documento es una taxonomía XBRL (con la extensión de archivo `.xsd`). Recuerde que los documentos XBRL de instancia, que son archivos XML y tienen la extensión `.xml`, deben editarse como archivos XML normales en otras vistas de edición y no se pueden editar en la vista Taxonomía XBRL. Esta vista se describe detalladamente en el apartado Vista XBRL[264] del *Manual del usuario*.

28.12.6 Vista Authentic

Este comando **cambia** la vista actual del documento a la vista Authentic[613].

En la vista Authentic puede editar documentos XML **basados en plantillas StyleVision Power Stylesheet creadas con Altova StyleVision.** Estas plantillas muestran los documentos XML en formato gráfico, lo cual facilita la edición de documentos XML.

Si un documento XML tiene asociado un archivo SPS (**Authentic | Asignar una hoja de estilos de StyleVision**[1413]), el documento XML se puede ver en la vista Authentic. También puede abrir un archivo SPS en la vista Authentic. Hay dos maneras de hacerlo:

- Seleccione el comando **Archivo | Nuevo** y haga clic en **Seleccione una hoja de estilos de StyleVision**.
- Seleccione el comando **Authentic | Documento nuevo** y busque el archivo SPS.

Para más información consulte la documentación de la vista Authentic[597] y la documentación de StyleVision.

28.12.7 Vista Explorador



Este comando **cambia** la vista actual del documento a la vista Explorador[279]. El explorador web preparado para XML de esta vista representa los documentos XML sirviéndose de la información de las hojas de estilos CSS/XSL disponibles.

Si selecciona la opción *Validar archivos automáticamente al guardarlos* en la sección *Archivo*[1584] del cuadro de diálogo "Opciones"[1584] (**Herramientas | Opciones**), el documento se valida automáticamente antes de cambiar a la vista Explorador. Para más información, consulte la sección Vista Explorador[279].

© 2016-2022 Altova GmbH

28.12.8 Expandir



Este comando (tecla de acceso rápido: **+** del teclado numérico) está disponible en la vista Cuadrícula y expande el elemento seleccionado un nivel. Después de la expansión el elemento permanece seleccionado. Así puede seguir pulsando la tecla **+** y seguir expandiendo los demás niveles del elemento.

28.12.9 Contraer



Este comando (tecla de acceso rápido: **-** del teclado numérico) está disponible en la vista Cuadrícula y **contrae** el elemento seleccionad un nivel. Los elementos también se pueden expandir y contraer haciendo clic en la barra gris situada a la izquierda de cada elemento.

28.12.10 Expandir todo



Este comando (tecla de acceso rápido: ***** del teclado numérico) está disponible en la vista Cuadrícula y, si está activo el margen de plegamiento, en la vista Texto. Este comando expande todos los nodos secundarios del elemento seleccionado.

28.12.11 Contraer elementos no seleccionados



Este comando (tecla de acceso rápido: **Ctrl** + **-** del teclado numérico) está disponible en la vista Cuadrícula y contrae los elementos que no están seleccionados para que pueda centrarse en el elemento seleccionado y en sus secundarios.

28.12.12 Ancho de tabla óptimo



Este comando está disponible en la vista Cuadrícula y ajusta el ancho de todas las columnas, para que la cadena de texto de mayor longitud quepa en una línea exactamente. En la sección *Vista* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**) puede limitar el ancho de tabla a un número de píxels determinado. Tenga en cuenta que el ancho de tabla óptimo se calcula en base a las celdas visibles de las columnas. Así se puede optimizar la vista cada vez que se expandan o contraigan elementos.

28.12.13 Ajuste automático de línea

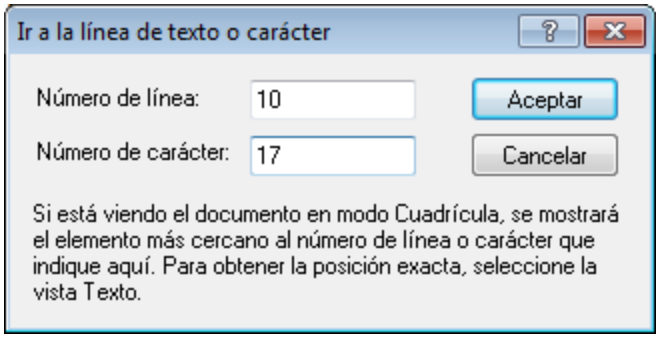


Este comando habilita/deshabilita el ajuste automático de línea en la vista Texto. Si se habilita, el texto se ajusta al borde de la pantalla.

28.12.14 Ir a la línea o carácter



Este comando (**Ctrl+G**) está disponible en las vistas Texto y Cuadrícula. Al ejecutar el comando aparece un cuadro de diálogo (*ver imagen*) donde puede escribir el número de línea/carácter al que quiere ir. En la vista Texto, el cursor saltará hasta la posición especificada en el cuadro de diálogo. En la vista Cuadrícula, se resaltará el nodo más cercano a la posición especificada..



Este comando es muy práctico si necesita navegar rápidamente hasta una posición concreta. Por ejemplo, a la línea donde se encontró un error.

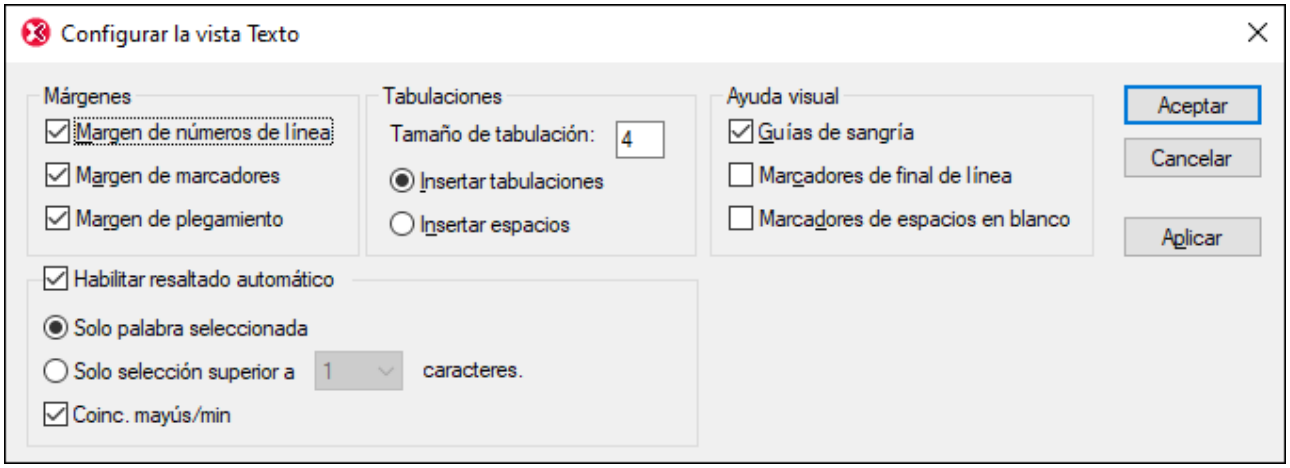
© 2016-2022 Altova GmbH

28.12.15 Ir al archivo

Este comando está disponible en las vistas Texto y Cuadrícula. Cuando se pone el cursor dentro de un texto (vista Texto) que hace referencia a un archivo o dentro de un nodo (vista Cuadrícula) cuyo texto hace referencia a un archivo y se ejecuta este comando, se abre el documento referenciado.

28.12.16 Configurar la vista Texto

El comando **Configurar la vista Texto** está disponible en la vista Texto y abre el cuadro de diálogo del mismo nombre (*ver imagen*). En el cuadro de diálogo puede configurar el aspecto de la vista Texto. Este comando también está disponible en forma de icono ( ) en la barra de herramientas Texto.



Márgenes

En el panel *Márgenes*, puede habilitar/deshabilitar el margen de números de línea, de marcadores y de plegamiento de código. Estos márgenes de la vista Texto muestran respectivamente (i) los números de línea, (ii) los marcadores y (iii) el plegamiento de código (los iconos que permiten expandir y contraer nodos). Con estas casillas puede elegir si los diferentes márgenes aparecen o no en la vista Texto. Los comandos relacionados con los marcadores están en el menú **Edición**. Para poder expandir y plegar nodos en la vista Texto es necesario habilitar el margen de plegamiento.

Tabulaciones

Las opciones del grupo *Tabulaciones* sirven para configurar el tamaño de las tabulaciones (medido en número de espacios). En este grupo de opciones también puede elegir si en la sangría de los documentos se utilizan tabulaciones o espacios cuando está activa la opción *Aplicar sangría* en la sección *Vista* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).

Ayuda visual

En el panel *Ayuda visual* puede habilitar/deshabilitar las guías de sangría (líneas verticales de puntos que señalan la sangría del texto), los marcadores de final de línea y los marcadores de espacios en blanco.

Habilitar resaltado automático

Si habilita esta casilla, en la vista Texto se resaltarán todas las coincidencias de una selección de texto. En este panel puede definir qué se considera una selección: puede ser una palabra entera o un número fijo de caracteres y se pueden tener en cuenta las mayúsculas y minúsculas a la hora de buscar coincidencias. Para la selección de caracteres puede especificar el mínimo de caracteres necesarios para encontrar una coincidencia (p. ej. dos caracteres o más). En la vista Texto se resaltarán todas las coincidencias de secuencias de caracteres que coincidan con la selección. Para la búsqueda de palabras, la aplicación considera como palabras los nombres de elemento, los nombres de atributo, los valores de atributo sin comillas y los paréntesis angulares de etiquetas de elemento.

Mapa de teclado

El mapa de teclado enumera las teclas de acceso rápido de XMLSpy y los comandos que invocan.

© 2016-2022 Altova GmbH

28.13 Menú Explorador

Los comandos del menú **Explorador** están disponibles solamente en la vista Explorador[279].



Atrás, Adelante

El comando **Atrás** (**Alt + flecha izda**) está disponible en las vistas Explorador y Esquema. También puede usar la tecla **Retroceso**. Este comando es muy práctico si hace clic en un enlace del documento XML y quiere volver al documento.

El comando **Adelante** (**Alt + flecha der**) permite avanzar por (i) páginas vistas previamente en la vista Explorador y (ii) vistas previas de componentes en la vista Esquema.

Actualizar

El comando **Actualizar (F5)** está disponible en la vista Explorador y sirve para actualizar la vista, es decir, se vuelve a cargar el documento y su documentos relacionados (las hojas de estilos CSS/XSL, los documentos DTD, etc.).

Fuentes

Al hacer clic en el comando **Fuentes** aparece un submenú donde puede seleccionar el tamaño de fuente predeterminado en el que se representa el texto del documento XML. Este comando está disponible en la vista Explorador.

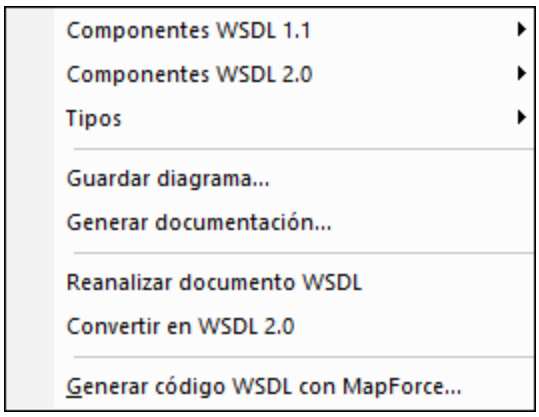
Ventana separada

El comando **Ventana separada** está disponible en la vista Explorador y abre la vista Explorador en otra ventana diferente, separada de las demás vistas. Así puede ver la vista Explorador en paralelo a una vista de edición, por ejemplo.

Si tras realizar cambios en la vista de edición desea actualizar la ventana separada de la vista Explorador, pulse **F5** en la vista de edición. Si desea volver a acoplar la ventana separada en la interfaz, haga clic en la ventana de la vista Explorador y después en el comando **Ventana separada**.

28.14 Menú WSDL

Los comandos del menú WSDL se habilitan cuando se abre un documento WSDL en la vista WSDL 251, un editor gráfico donde puede crear y editar documentos WSDL.

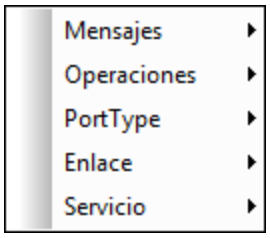


Para más información, consulte la sección Vista WSDL 251. Para empezar a familiarizarse con WSDL, consulte el *Tutorial de WSDL* 776.

Véase también: encontrará más información sobre cómo trabajar con documentos WSDL en las secciones Vista WSDL 251 y Tutorial WSDL 776.

28.14.1 Componentes WSDL 1.1

Al pasar el puntero del ratón sobre la opción de menú **Componentes WSDL 1.1** aparece un submenú (*imagen siguiente*). Las opciones de este submenú sirven para editar componentes WSDL 1.1.



Cada opción (*imagen anterior*) despliega a su vez un submenú con comandos relacionados con el componente seleccionado. Los comandos se describen en detalle en los subapartados siguientes.

28.14.1.1 Mensajes

Insertar mensaje

Añade un mensaje nuevo al documento WSDL. Se crea un componente `message` en el ayudante de entrada Vista general, donde aparece resaltado el mensaje recién creado.

 © 2016-2022 Altova GmbH

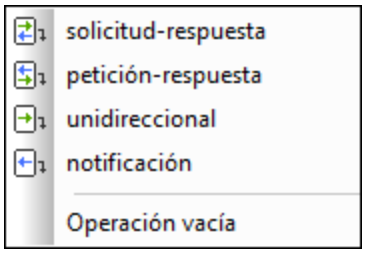
Eliminar mensaje
Elimina el mensaje seleccionado del elemento de entrada o salida.

Agregar parte de mensaje (parameter)
Añade una parte de mensaje (parameter) al mensaje seleccionado.

Eliminar parte de mensaje (parameter)
Elimina una parte de mensaje (parameter) del mensaje seleccionado.

28.14.1.2 Operaciones

Anexar operación
Anexa una operación nueva al portType seleccionado. El tipo de operación se selecciona en el submenú que aparece al pasar el puntero por la opción de menú **Anexar operación** (*imagen siguiente*).



Eliminar operación
Elimina la operación seleccionada.

Agregar elemento de entrada
Añade un elemento de entrada a la operación seleccionada.

Agregar elemento de salida
Añade un elemento de salida a la operación seleccionada.

Agregar elemento de error
Añade un elemento de error a la operación seleccionada.

Eliminar elemento de entrada, salida o error
Elimina el elemento de entrada, salida o error seleccionado

Agregar mensaje nuevo al elemento de entrada, salida o error

Añade un mensaje (predeterminado) nuevo al elemento de entrada/salida/error seleccionado

28.14.1.3 PortType

Insertar portType

Añade un portType nuevo a la columna *PortTypes* de la vista WSDL.

Eliminar portType

Elimina el portType seleccionado de la columna *PortTypes* de la vista WSDL.

28.14.1.4 Enlace

Insertar enlace

Añade un enlace nuevo a la columna *Bindings* de la vista WSDL.

Eliminar enlace

Elimina el enlace seleccionado la columna *Bindings* de la vista WSDL.

Anexar secundario

Permite añadir un elemento de extensibilidad nuevo a un mensaje de entrada o salida. Si esta opción de menú está deshabilitada, significa que la operación no está permitida en la posición actual. Para más información sobre los componentes de extensibilidad, consulte la especificación WSDL del W3C.

Estos son los componentes de extensibilidad disponibles:

- soap:body
- soap:header
- soap:headerfault
- soap:fault
- mime:content
- mime:multipartrelated
- mime:part
- mime:mimeXml
- http:urlencoded
- http:urlreplacement

Eliminar elemento de extensibilidad

Elimina el componente de extensibilidad seleccionado.

© 2016-2022 Altova GmbH

28.14.1.5 Servicio

Insertar servicio

Añade un servicio nuevo en la columna *Services* de la vista WSDL.

Eliminar servicio

Elimina el servicio seleccionado de la columna *Services* de la vista WSDL.

Insertar puerto

Añade un puerto nuevo al servicio seleccionado en la columna *Services* de la vista WSDL.

Eliminar puerto

Elimina el puerto seleccionado del servicio activo.

28.14.2 Componentes WSDL 2.0

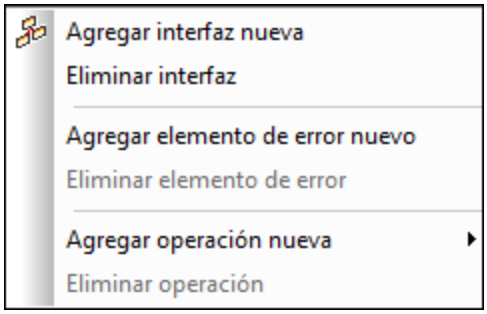
Al pasar el puntero del ratón sobre la opción de menú **Componentes WSDL 2.0** aparece un submenú (*imagen siguiente*). Las opciones de este submenú sirven para editar componentes WSDL 2.0.



Cada opción (*imagen anterior*) despliega a su vez un submenú con comandos relacionados con el componente seleccionado. Los comandos se describen en detalle en los subapartados siguientes.

28.14.2.1 Interfaz

El menú **Interfaz** (*imagen siguiente*) ofrece estos comandos:



Agregar interfaz nueva

Añade un nuevo recuadro de interfaz en la columna *Interfaces* de la ventana principal. El nombre predeterminado de la interfaz aparece resaltado en el recuadro para que pueda introducir un nombre nuevo inmediatamente.

Eliminar interfaz

Elimina la interfaz seleccionada.

Agregar elemento de error nuevo

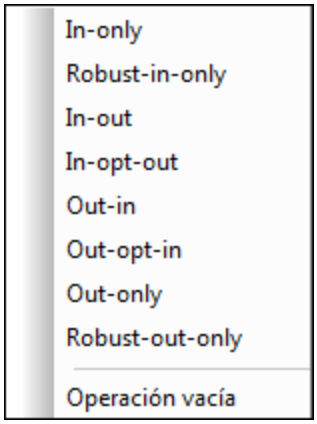
Añade un elemento de error `fault` nuevo a la interfaz seleccionada. El nombre predeterminado del elemento `fault` aparece resaltado en el recuadro para que pueda introducir un nombre nuevo inmediatamente.

Eliminar elemento de error

Elimina el elemento `fault` seleccionado.

Agregar operación nueva

Añade un elemento `operation` nuevo a la interfaz seleccionada. En el menú emergente (*imagen siguiente*) puede seleccionar qué tipo de operación se añade a la interfaz:



El nombre predeterminado del elemento `operation` aparece resaltado en el recuadro para que pueda introducir un nombre nuevo inmediatamente.

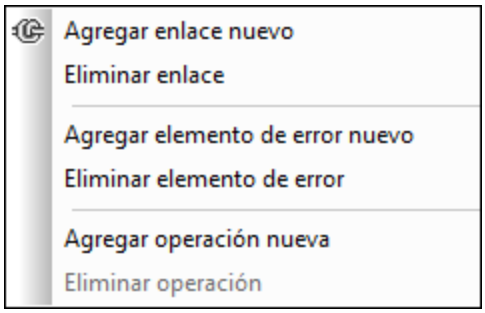
Eliminar operación

Elimina la operación seleccionada.

© 2016-2022 Altova GmbH

28.14.2.2 Enlace

El menú **Enlace** (*imagen siguiente*) ofrece estos comandos:



Agregar enlace nuevo

Añade un nuevo recuadro de enlace en la columna *Bindings* de la ventana principal. El nombre predeterminado del enlace aparece resaltado en el recuadro para que pueda introducir un nombre nuevo inmediatamente.

Eliminar enlace

Elimina el enlace seleccionado.

Agregar elemento de error nuevo

Añade un elemento de error `fault` nuevo al enlace seleccionado. El elemento `fault` de un enlace contiene un atributo `ref`, que hace referencia al elemento `fault` declarado en una interfaz. El elemento `fault` recién creado en el enlace incluye un cuadro combinado donde puede seleccionar el elemento `fault` de interfaz al que se debe hacer referencia.

Eliminar elemento de error

Elimina el elemento `fault` seleccionado.

Agregar operación nueva

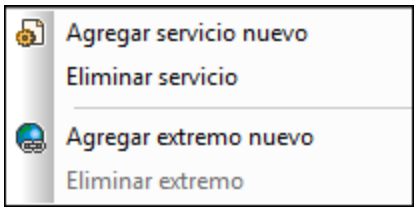
Añade un elemento `operation` nuevo al enlace seleccionado. El elemento `operation` de un enlace contiene un atributo `ref`, que hace referencia al elemento `operation` declarado en una interfaz. El elemento `operation` recién creado en el enlace incluye un cuadro combinado donde puede seleccionar el elemento `operation` de interfaz al que se debe hacer referencia.

Eliminar operación

Elimina la operación seleccionada.

28.14.2.3 Servicio

El menú **Servicio** (*imagen siguiente*) ofrece estos comandos:



Agregar servicio nuevo

Añade un recuadro de servicio nuevo en la columna *Services* de la ventana principal. El nombre predeterminado del servicio aparece resaltado en el recuadro para que pueda introducir un nombre nuevo inmediatamente. La referencia a la interfaz se puede seleccionar en el cuadro combinado de la propiedad `interface`.

Eliminar servicio

Elimina el servicio seleccionado.

Agregar extremo nuevo

Añade un elemento `endpoint` nuevo al servicio seleccionado. El nombre predeterminado del extremo aparece resaltado en el recuadro para que pueda introducir un nombre nuevo inmediatamente. La referencia al enlace se puede seleccionar en el cuadro combinado de la propiedad `binding`. La dirección del extremo debe introducirse en el campo de la propiedad `address`.

Eliminar extremo

Elimina el elemento `endpoint` seleccionado.

28.14.3 Tipos, Guardar diagrama

El menú **Tipos** incluye un submenú con varios comandos que se describen a continuación:

- **Esquema nuevo**
- **Incrustar esquema**
- **Extraer esquemas**
- **Editar esquemas en la vista Esquema**

El comando **Guardar diagrama** guarda el diagrama del diseño en un archivo PNG.

Tipos | Esquema nuevo

Esta opción se habilita solamente si el archivo WSDL no contiene un elemento `schema`.

© 2016-2022 Altova GmbH

Recuerde que al usar la opción de menú **Archivo | Nuevo** se incluye un elemento `schema` en la estructura base del archivo WSDL. En este caso no estaría disponible la opción de menú **Esquema nuevo**.

```
<types>
            <xs:schema/>
</types>
```

Tipos | Incrustar esquema

Este comando abre el cuadro de diálogo "Incrustar esquema", donde puede buscar el archivo de esquema que desea incrustar. El esquema se crea como esquema alineado dentro del elemento `types`. Si el esquema seleccionado ya se importó, aparece un diálogo preguntando si desea incrustar el esquema ya importando. Si elige incrustar el esquema importado, se convertirá en un esquema alineado dentro del elemento `types`.

Tipos | Extraer esquemas

Al seleccionar este comando se abren como archivos temporales en la vista Esquema todos los esquemas incrustados (los esquemas alineados del elemento `types`) y se abre un cuadro de diálogo "Guardar como" para cada archivo. Si decide guardar algún archivo de esquema, este se extrae del archivo WSDL, se guarda en la ubicación indicada y se importa en el archivo WSDL. Es decir, el esquema deja de ser un esquema incrustado y pasa a ser un esquema externo importado.

Tipos | Editar esquemas en la vista Esquema

Abre un archivo de esquema base si el archivo WSDL no contiene una referencia a ningún esquema concreto. Tal es el caso si usó la opción de menú **Archivo | Nuevo** para crear el documento WSDL. Si existe una referencia a un esquema concreto, el esquema se abre en la vista Esquema del editor WSDL gráfico.

28.14.4 Generar documentación

El comando **WSDL | Generar documentación** genera documentación detallada del archivo WSDL actual. Los archivos de documentación de salida pueden ser en formato HTML, MS Word, RTF y PDF y se puede modificar y utilizar sin restricciones. La documentación incluye información sobre los componentes seleccionados en el cuadro de diálogo "Documentación WSDL" (que aparece al seleccionar el comando **Generar documentación**). Los componentes relacionados entre sí se enlazan mediante hipervínculos, para que pueda navegar por la documentación de componente en componente. Recuerde que también puede generar documentación WSDL para los **archivos WSDL y esquemas XML importados**. Las opciones de generación de documentación se describen en el subapartado Opciones de documentación 1500.

Nota: para generar la documentación WSDL en formato MS Word es necesario tener MS Word (versión 2000 o superior) instalado en el equipo.

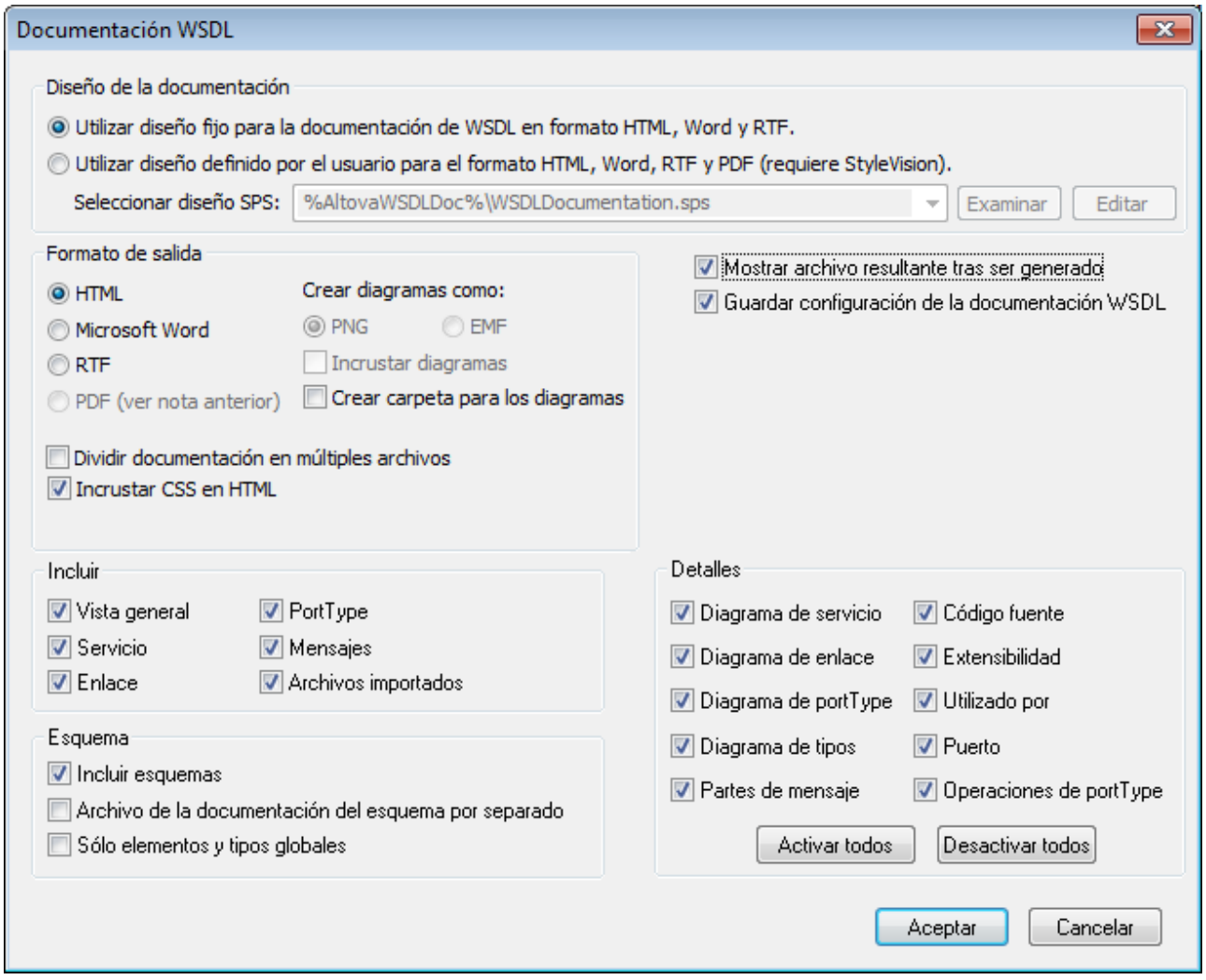
Puede usar el diseño fijo de XMLSpy para generar la documentación o puede usar un archivo SPS de StyleVision para personalizar el diseño de la documentación WSDL. Para más información, consulte el apartado Diseño definido por el usuario 1502.

Nota: es necesario tener StyleVision instalado en el equipo para generar la documentación WSDL usando un archivo SPS.

28.14.4.1 Opciones de documentación

Al hacer clic en el comando **WSDL | Generar documentación** aparece el cuadro de diálogo "Documentación WSDL" (*imagen siguiente*), donde puede seleccionar las opciones de configuración de la documentación.

En el panel *Diseño de la documentación* de este cuadro de diálogo puede elegir si utiliza el diseño fijo de XMLSpy para generar la documentación o si utiliza un diseño personalizado creado en un archivo SPS de StyleVision. Observe que la documentación en formato PDF está disponible solamente si utiliza un archivo SPS de StyleVision, pero no si genera la documentación con el diseño fijo. Para más información, consulte el apartado Diseño definido por el usuario[1502].



A continuación describimos las demás opciones del cuadro de diálogo "Documentación WSDL". Las opciones de los paneles *Incluir* y *Detalles* difieren dependiendo de si el documento activo es un archivo WSDL 1.1 o WSDL 2.0. La imagen anterior muestra el cuadro de diálogo "Documentación WSDL" de un documento WSDL 1.1.

- Puede elegir el formato de la documentación en el panel *Formato de salida:* las opciones disponibles son HTML, Microsoft Word, RTF y PDF. (El formato PDF solamente está disponible si usa un archivo

© 2016-2022 Altova GmbH

SPS de StyleVision para generar la documentación.) Al hacer clic en **Aceptar** aparece un diálogo donde debe introducir un nombre para el archivo de salida y seleccionar la ubicación donde desea guardarlo.

- Si usa el diseño fijo para generar la documentación en formato Word, los archivos se guardan con la extensión de archivo `.doc`. Si usa un archivo SPS de StyleVision, entonces se guardan con la extensión de archivo `.docx`.
- La documentación se puede generar en un solo archivo o dividirse en varios archivos más pequeños. Si prefiere la segunda opción, se crea un archivo por cada componente. Para elegir los componentes que se incluyen en la documentación, use las casillas del panel *Incluir*. Además, si usa el diseño fijo para generar la documentación, se crean enlaces entre los diferentes documentos.
- Si elige el formato de salida HTML, las definiciones de estilos CSS se pueden guardar en un archivo CSS o se pueden incrustar en el archivo HTML (en el elemento `<head>`). Si prefiere crear un archivo CSS, este lleva el mismo nombre que el archivo HTML, pero con la extensión de archivo `.css`. Elija la opción deseada activando o desactivando la casilla *Incrustar CSS en HTML*.
- La casilla *Incrustar diagramas* se habilita cuando se selecciona el formato de salida MS Word, RTF o PDF. Si marca esta casilla, los diagramas se incrustan en el archivo de la documentación, en formato PNG. Si no marca esta casilla, los diagramas se guardan en archivos PNG y el archivo de la documentación incluye enlaces a ellos por medio de vínculos de objeto.
- Si elige el formato de salida HTML, los diagramas se guardan en archivos PNG. Si marca la casilla *Crear carpeta para los diagramas*, se crea una carpeta con el mismo nombre que el archivo HTML y en ella se guardan todos los archivos PNG. El nombre de la carpeta sigue este modelo: `NombreArchivoHTML_diagrams`. Si no marca la casilla *Crear carpeta para los diagramas*, los archivos PNG se guardan en la misma carpeta que el archivo HTML.
- En el panel *Incluir* puede seleccionar qué componentes comprende la documentación. Si marca la casilla *Vista general*, al principio de la documentación aparece una lista con todos los componentes, ordenados por tipo. Si marca la casilla *Archivos importados* (WSDL 1.1) o *Archivos importados o incluidos* (WSDL 2.0), los componentes de los archivos importados (y de los archivos incluidos si trabaja con WSDL 2.0) se incluyen en la documentación del esquema.
- En el panel *Esquema* puede elegir si se documentan o no los esquemas del archivo. Si marca la casilla *Incluir esquemas* también puede elegir (i) si la documentación del esquema se guarda por separado en otro archivo o si se incluye en el archivo de documentación principal y (ii) si se documenta todo el esquema o solamente los elementos, los tipos simples y los tipos complejos globales.
- El panel *Detalles* enumera los detalles de los componentes que puede incluir la documentación. Puede usar los botones **Activar todos** y **Desactivar todos** para seleccionar las opciones más rápido.
- La casilla *Mostrar archivo resultante tras ser generado* está disponible para todos los formatos de salida. Si marca esta casilla, el archivo resultante se presenta en la vista Explorador (formato HTML), en MS Word (formato MS Word) y en las aplicaciones predeterminadas para archivos `.rtf` (formato RTF) y archivos `.pdf` (formato PDF).

Valores de parámetro

Si el archivo SPS de StyleVision contiene definiciones de parámetros, al hacer clic en **Aceptar** aparece un cuadro de diálogo que enumera todos los parámetros definidos en el archivo SPS. En este cuadro de diálogo puede introducir valores nuevos si desea reemplazar los valores predeterminados que se asignaron en el archivo SPS.

28.14.4.2 Diseño definido por el usuario

Si lo prefiere, puede crear un diseño personalizado para la documentación WSDL en lugar de usar el diseño fijo estándar de XMLSpy. Este diseño personalizado se crea en un archivo SPS de StyleVision, que no es más que una plantilla de diseño para el documento de salida.

Cómo crear un archivo SPS

El archivo SPS (o StyleVision Power Stylesheet) se crea con Altova StyleVision. El archivo SPS necesario para generar documentación WSDL debe estar basado en un esquema XML que defina la estructura de la documentación WSDL. XMLSpy viene con dos esquemas de muestra, uno para WSDL 1.1 y otro para WSDL 2.0. Se trata de los esquemas `WSDLDocumentation.xsd` y `WSDL20Documentation.xsd`, que están en estos directorios de la carpeta (Mis) Documentos[36]:

- `C:\Documents and Settings\<nombreUsuario>\Mis Documentos\Altova\XMLSpy2023\Documentation\WSDL`.
- `C:\Documents and Settings\<nombreUsuario>\Mis Documentos\Altova\XMLSpy2023\Documentation\WSDL20`.

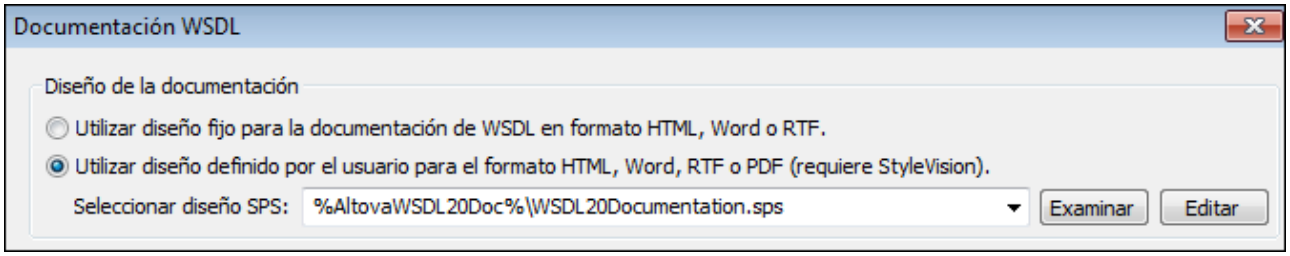
El proceso de creación de un diseño SPS en StyleVision consiste en colocar nodos del esquema en la plantilla del diseño y asignarles estilos y propiedades. Después se añaden al diseño componentes adicionales como vínculos, tablas, imágenes, etc. Para más información sobre cómo crear un diseño SPS consulte el *Manual del usuario de StyleVision*.

La ventaja de usar un SPS a la hora de generar la documentación WSDL es que puede personalizar totalmente el aspecto de la documentación. Además, recuerde que la documentación WSDL se puede generar en formato PDF solamente si usa un archivo SPS. La documentación WSDL no se puede generar en formato PDF si usa el diseño fijo de XMLSpy.

Cómo elegir el SPS para generar la documentación WSDL

Una vez creado, el archivo SPS se puede usar para generar la documentación WSDL. Abra el cuadro de diálogo "Documentación WSDL" (con el comando **WSDL | Generar documentación**) y seleccione el botón de opción *Utilizar diseño definido por el usuario.* Después haga clic en el botón **Examinar** y busque el SPS que desea usar. Para terminar haga clic en el botón **Aceptar** del diálogo y en el cuadro de diálogo que aparece seleccione una carpeta y un nombre para el archivo de salida.

Nota: el archivo SPS debe localizar correctamente el esquema en el que se basa: `WSDLDocumentation.xsd` o `WSDL20Documentation.xsd` (*véase más arriba*).



XMLSpy viene con dos diseños SPS editables, uno para WSDL 1.1 y otro para WSDL 2.0. Estos archivos están en los directorios `WSDL` y `WSDL20` de la carpeta (Mis) Documentos[36]: `C:\Documents and`

 © 2016-2022 Altova GmbH

`Settings\<nombreUsuario>\Mis Documentos\Altova\XMLSpy2011\Documentation\`. Los archivos se llaman:

- `WSDL\WSDLDocumentation.sps`
- `WSDL20\WSDL20Documentation.sps`

Estos archivos, junto con los archivos SPS examinados recientemente, están disponibles en el cuadro combinado de la opción *Utilizar diseño definido por el usuario* (*imagen anterior*).

Si hace clic en el botón **Editar** del panel *Diseño de la documentación*, el archivo SPS seleccionado se abre en StyleVision. Si desea obtener una vista previa del archivo resultante en StyleVision, necesitará un archivo XML de trabajo. Puede usar los archivos XML de muestra `TimeService.xml` y `TimeService20.xml` que vienen con XMLSpy. Estos archivos están en la carpeta (Mis) Documentos[36]:

```
C:\Documents and Settings\<nombreUsuario>\Mis
Documentos\Altova\XMLSpy2023\Documentation\WSDL(20)\SampleData
```

Nota: es necesario tener StyleVision instalado en el equipo para generar la documentación WSDL usando un archivo SPS.

28.14.5 Reanalizar el documento WSDL

Este comando vuelve a analizar todo el documento WSDL, lo cual puede ser necesario en algunos casos. Por ejemplo, cuando se modifica un esquema asociado al documento WSDL. Al hacer clic en este comando el documento WSDL se vuelve a analizar y se actualiza con los datos actualizados del documento de esquema.

28.14.6 Convertir en WSDL 2.0

El comando **Convertir en WSDL 2.0** está disponible solamente si el documento activo en la vista WSDL es un archivo WSDL 1.1. Este comando genera un documento WSDL 2.0 a partir del documento WSDL 1.1 activo. Al hacer clic en este comando aparece el cuadro de diálogo "Guardar archivo", donde puede seleccionar la ubicación y el nombre del archivo WSDL 2.0 que será generado por XMLSpy.

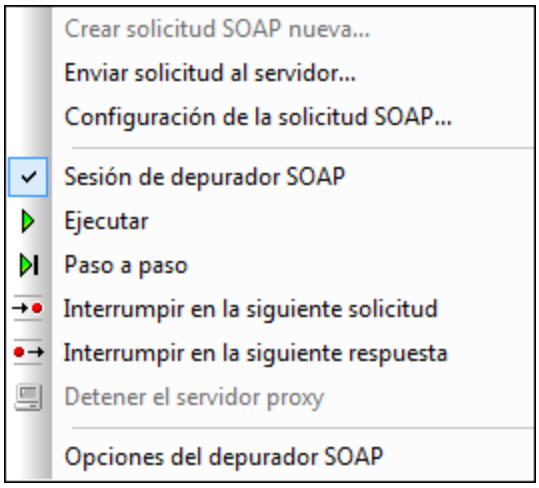
Al hacer clic en el botón **Aceptar** del diálogo "Guardar archivo" se genera un documento WSDL en la ubicación elegida. El documento WSDL generado se abre automáticamente en una pestaña nueva de la vista WSDL, donde puede seguir editando el archivo.

28.14.7 Generar código WSDL con MapForce

El comando **Generar código WSDL con MapForce** inicia la aplicación Altova MapForce (si está instalada en el equipo). En MapForce puede realizar asignaciones de datos entre un esquema y otros archivos DTD, XSD o BD y generar un archivo XML o código de programa.

28.15 Menú SOAP

XMLSpy es compatible con la versión 1.1 y 1.2 de SOAP y con la versión 1.1 y 2.0 de WSDL.



La sección *SOAP* [788] *del Manual del usuario* explica cómo utilizar el depurador SOAP usando el servidor **nanonull.com timeservice** de Altova. Para probar el depurador SOAP use este servicio. En los apartados siguientes se menciona el servicio web AirportWeather, pero en algunos casos este servicio no está disponible.

Puede usar las funciones SOAP de XMLSpy para:

- probar sus servicios web sin necesidad de implementar aplicaciones cliente
- probar rápidamente los servicios web de terceros

Información adicional

Para más información sobre estas especificaciones consulte estas páginas:

SOAP http://www.w3.org/TR/SOAP/
WSDL http://www.w3.org/TR/wsdl

28.15.1 Crear solicitud SOAP nueva

Este comando crea un documento de solicitud SOAP nuevo. El proceso incluye estos pasos:

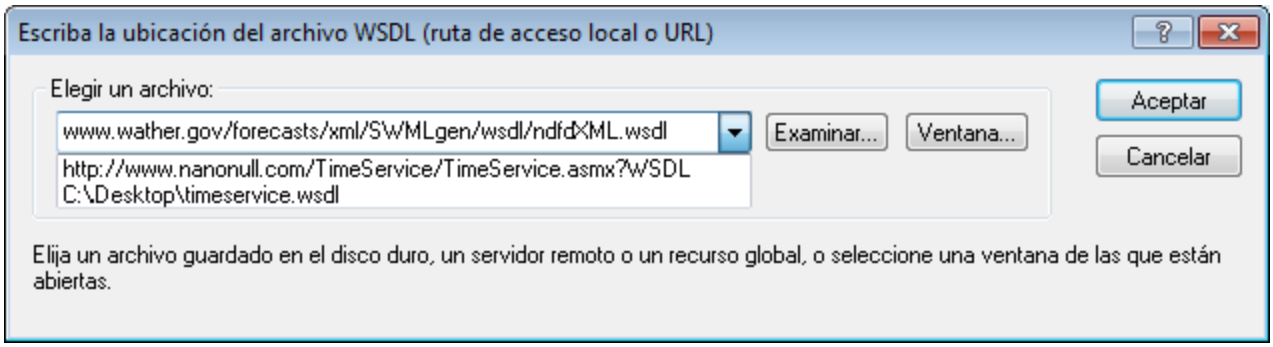
1. Se introduce la ubicación del archivo WSDL y se establece la conexión con el servidor [1505].
2. El servidor responde con una lista de operaciones. Se selecciona la operación SOAP [1505].
3. El servidor responde con un formulario de solicitud SOAP en formato XML. Se define el formulario de solicitud SOAP [1505].

Para ilustrar el proceso vamos a crear una solicitud SOAP para el servicio SOAP "US National Digital Forecast Database (NDFD)" (http://www.nws.noaa.gov/xml/).

© 2016-2022 Altova GmbH

Conectarse al servidor SOAP

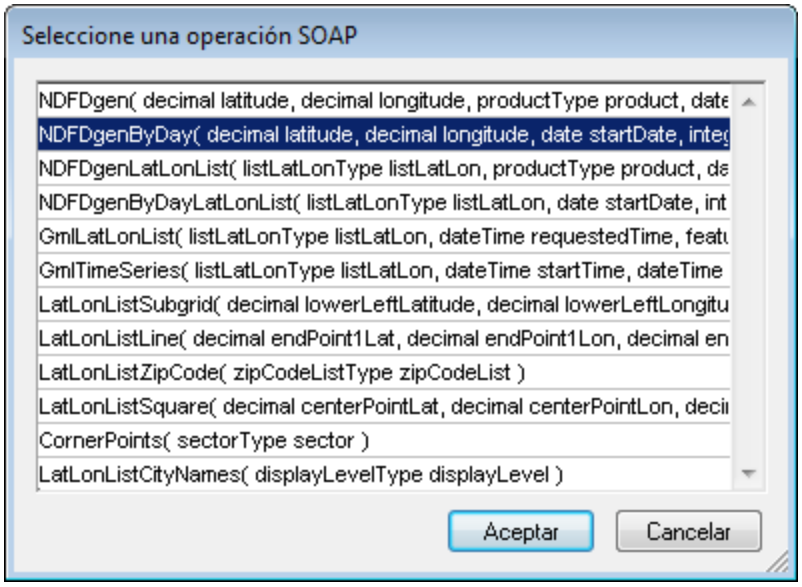
La conexión al servidor SOAP se establece por medio de un archivo WSDL. En nuestro ejemplo, el URI del archivo WSDL es: http://www.weather.gov/forecasts/xml/DWMLgen/wsdl/ndfdXML.wsdl. Para establecer la conexión haga clic en el comando **Crear solicitud SOAP nueva** y escriba el URI del archivo en el cuadro de diálogo que aparece (*imagen siguiente*).



Haga clic en **Aceptar** para confirmar la selección.

Seleccionar la operación SOAP

El servidor responde con una lista de operaciones (*imagen siguiente*).



Seleccione una operación y haga clic en **Aceptar**. En nuestro ejemplo seleccionamos la operación `NDFDgenByDay`.

Definir la solicitud SOAP

El servidor responde enviando un archivo XML, que se muestra en la vista Texto de XMLSpy. En nuestro ejemplo recibimos este archivo XML:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:m0="http://www.weather.gov/forecasts/xml/DWMLgen/schema/DWML.xsd">
   <SOAP-ENV:Body>
      <m:NDFDgenByDay
xmlns:m="http://www.weather.gov/forecasts/xml/DWMLgen/wsdl/ndfdXML.wsdl"
       SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <latitude xsi:type="xsd:decimal">0.0</latitude>
         <longitude xsi:type="xsd:decimal">0.0</longitude>
         <startDate xsi:type="xsd:date">1967-08-13</startDate>
         <numDays xsi:type="xsd:integer">0</numDays>
         <format xsi:type="m0:formatType">String</format>
      </m:NDFDgenByDay>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Y después rellenamos los parámetros necesarios (marcados en rojo en el ejemplo siguiente; rellene la fecha de inicio con la fecha actual o una de la semana siguiente):

```
<SOAP-ENV:Body>
   <m:NDFDgenByDay
xmlns:m="http://www.weather.gov/forecasts/xml/DWMLgen/wsdl/ndfdXML.wsdl"
    SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <latitude xsi:type="xsd:decimal">45</latitude>
         <longitude xsi:type="xsd:decimal">-90</longitude>
         <startDate xsi:type="xsd:date">2019-12-10</startDate>
         <numDays xsi:type="xsd:integer">1</numDays>
         <format xsi:type="m0:formatType">24 hourly</format>
   </m:NDFDgenByDay>
</SOAP-ENV:Body>
```

Ya hemos terminado de **definir** la solicitud SOAP. El paso siguiente es enviar la solicitud[1506].

28.15.2 Enviar solicitud al servidor

En la sección Crear solicitud SOAP nueva[1504] se describe cómo definir una solicitud SOAP. Una vez se ha creado la solicitud SOAP, que es un documento XML, conviértala en el documento activo. Después, seleccione este comando para enviar la solicitud SOAP al servidor SOAP.

Después de enviar la solicitud SOAP, recibirá una respuesta del servidor SOAP. Esta respuesta es un documento XML que se presenta en la vista Texto de XMLSpy. Por ejemplo, en la imagen siguiente aparece el documento XML que se recibió en respuesta a la solicitud SOAP que definimos en el apartado Crear solicitud SOAP nueva[1504].

© 2016-2022 Altova GmbH



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SO
http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:
http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/enc
   <SOAP-ENV:Body>
      <ns1:NDFDgenByDayResponse xmlns:ns1="http://www.weather.gov/forecasts/xml/DWMLgen/wsdl/n
         <dwmlByDayOut xsi:type="xsd:string"><?xml version="1.0"?>
<dwml version="1.0" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/
xsi:noNamespaceSchemaLocation="http://www.nws.noaa.gov/forecasts/xml/DWMLgen/schema/DWML.xsd">
  <head>
  <data>
    <location>
      <location-key>point1</location-key>
      <point latitude="45.00" longitude="-90.00"/>
    </location>
    <moreWeatherInformation applicable-location="point1">
http://forecast.weather.gov/MapClick.php?textField1=45.00&amp;amp;textField2=-90.00
    </moreWeatherInformation>
    <time-layout time-coordinate="local" summarization="24hourly">
    <time-layout time-coordinate="local" summarization="12hourly">
    <parameters applicable-location="point1">
      <temperature type="maximum" units="Fahrenheit" time-layout="k-p24h-n1-1">
        <name>Daily Maximum Temperature</name>
        <value>45</value>
      </temperature>
      <probability-of-precipitation type="12 hour" units="percent" time-layout="k-p12h-n2-2">
        <name>12 Hourly Probability of Precipitation</name>
        <value>14</value>
        <value xsi:nil="true"/>
      </probability-of-precipitation>
    </parameters>
  </data>
</dwml>
</dwmlByDayOut>
      </ns1:NDFDgenByDayResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Guardar y volver a utilizar una solicitud SOAP

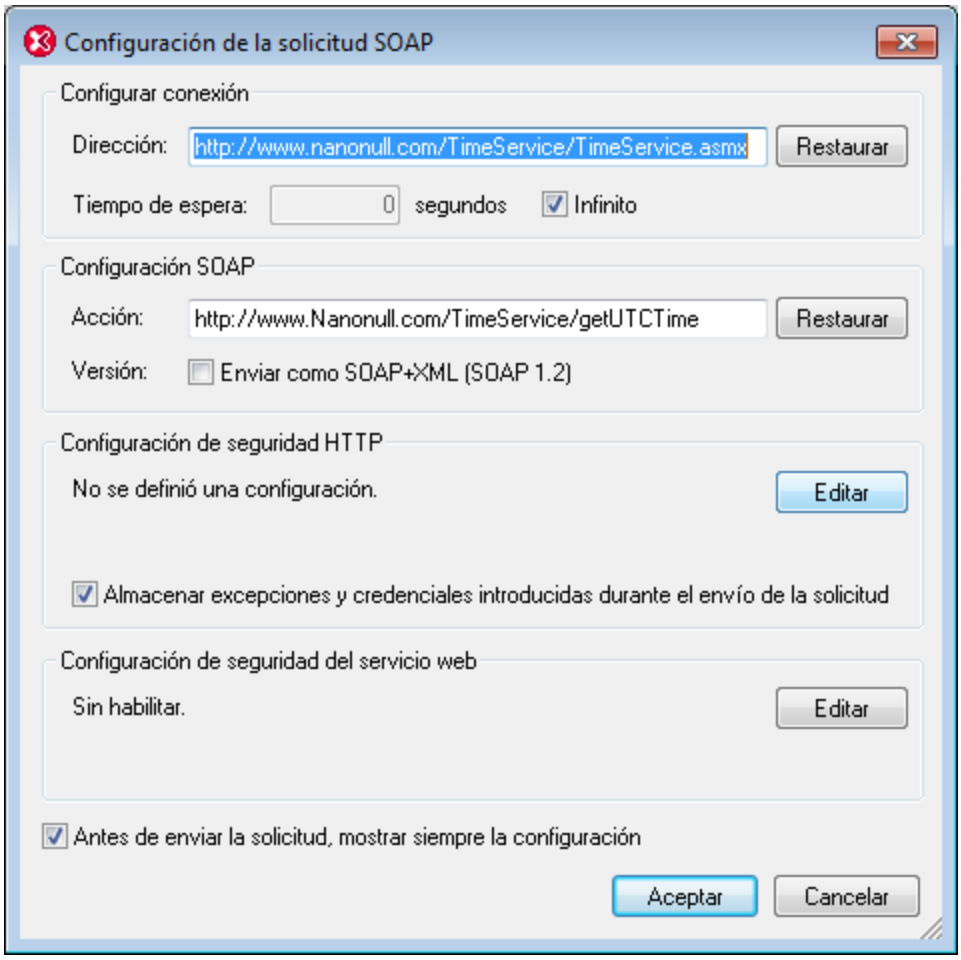
En XMLSpy puede guardar la solicitud SOAP para volver a enviarla más tarde. Siga estos pasos:

1. Guarde el documento XML de solicitud SOAP (**Archivo | Guardar como**).
2. Cierre el archivo de solicitud SOAP.
3. Vuelva a abrir el documento XML de solicitud SOAP y seleccione la opción de menú **SOAP | Enviar solicitud al servidor**. (Recuerde que puede usar cualquier archivo XML como solicitud SOAP.)

28.15.3 Configuración de la solicitud SOAP

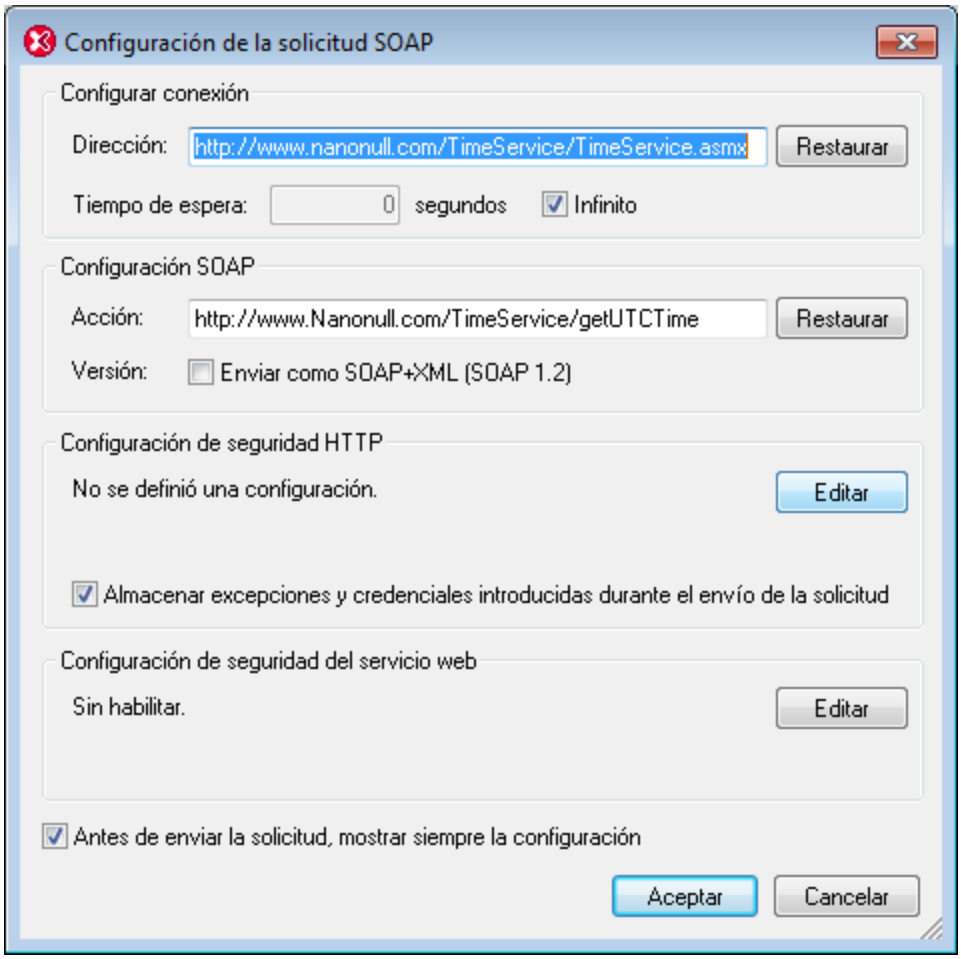
Este comando abre el cuadro de diálogo "Configuración de la solicitud SOAP" (imagen siguiente) donde puede especificar cómo se configura la solicitud SOAP[1504].

1. Asegúrese de que el documento activo es la solicitud SOAP.
2. Seleccione el comando de menú **SOAP | Configuración de la solicitud SOAP**. Aparece este cuadro de diálogo.



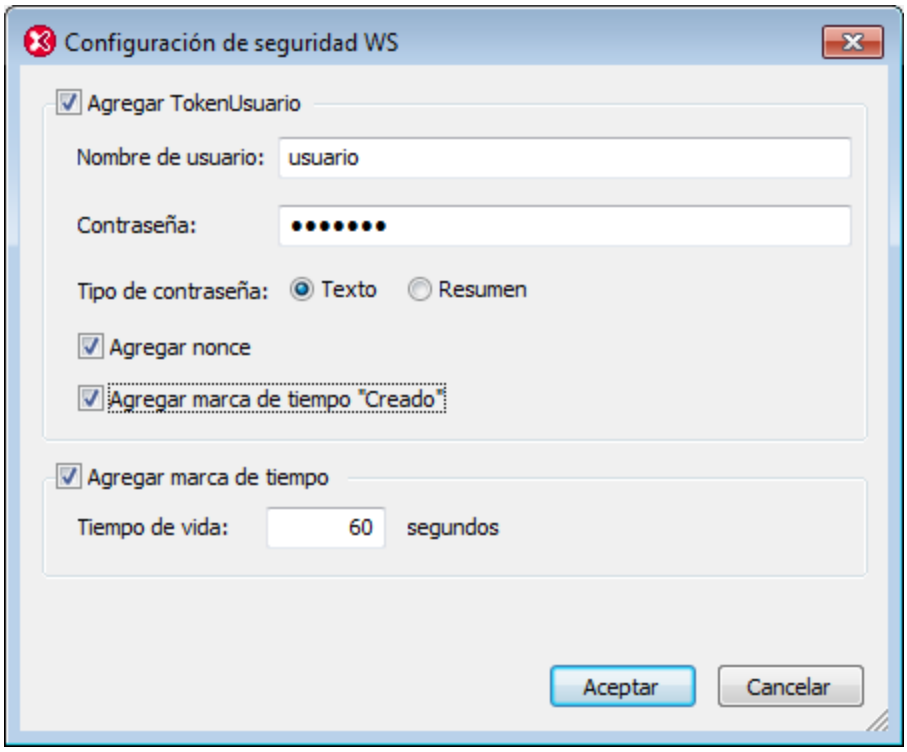
3. En el campo *Dirección* introduzca el destino de la conexión. Si la solicitud SOAP se creó a partir de un archivo WSDL en XMLSpy, entonces el valor de este campo será la ubicación del extremo seleccionado en el archivo WSDL. Haga clic en **Restaurar** para obtener este extremo. También puede especificar el valor del tiempo de espera de la conexión en segundos o indicar un tiempo de espera infinito.
4. En el campo *Acción* introduzca la acción SOAP que debe llevarse a cabo. Para enviar la solicitud como SOAP 1.2 marque la casilla *Enviar como SOAP+XML (SOAP 1.2)*. Si la solicitud SOAP se creó a partir de un archivo WSDL en XMLSpy, entonces la acción SOAP se recibe del elemento de extensión situado bajo la operación de enlace SOAP correspondiente del archivo WSDL. En este caso la versión SOAP ya estará seleccionada previamente. La versión SOAP afecta al valor del encabezado HTTP Content-Type: `text/xml` or `application/soap+xml`. Haga clic en **Restaurar** para obtener la acción SOAP del archivo WSDL.
5. El panel *Configuración de seguridad HTTP* ofrece un resumen de las opciones de seguridad. Si marca la casilla *Almacenar excepciones y credenciales introducidas durante el envío de la solicitud,* todas las opciones se guardarán cuando se envíe la solicitud y podrán volver a utilizarse en la próxima solicitud. Haga clic en **Editar** para abrir el cuadro de diálogo "Configuración de seguridad HTTP". Más abajo encontrará instrucciones para instalar certificados de servidor.

© 2016-2022 Altova GmbH



Si desea **permitir diferencias entre el nombre de host** del certificado de servidor y la dirección que utiliza o **permitir certificados expirados**, marque las casillas correspondientes en el cuadro de diálogo. Si el servidor exige un certificado de cliente, aquí puede especificar su ubicación. Si el servidor exige una autenticación, indique aquí el usuario y la contraseña para la autenticación estándar. Además, se conoce como **autenticación preventiva** el proceso en el que la solicitud cliente inicial contiene la información de autenticación necesaria. Si el servidor lo exige, marque la casilla *Autenticación preventiva*. De lo contrario, deje esta casilla sin marcar.

6. Ademas de estas opciones de seguridad, puede configurar la seguridad del servicio web. Haga clic en el botón **Editar** del panel *Configuración de seguridad del servicio web*. En el cuadro de diálogo que aparece puede configurar el nombre de usuario, la contraseña, el tipo de contraseña, una cadena de código nonce generada automáticamente y una marca de tiempo. También puede indicar el período de validez de la información de seguridad con la casilla *Agregar marca de tiempo*. Con las opciones elegidas en este cuadro de diálogo se crea un fragmento XML que contiene la información de seguridad y que se incrusta en la solicitud SOAP (*ver ejemplo de código que aparece al final de este apartado*).



7. Cuando termine de elegir las opciones haga clic en **Aceptar**.

Qué son los certificados de confianza

Los productos de Altova utilizan Internet Explorer (IE) para acceder y gestionar certificados de confianza de servidores web seguros. Instalar el certificado de un servidor web en IE permite a IE acceder al servidor web sin emitir una advertencia ni cancelar el proceso. Para instalar el
certificado de un servidor web seguro siga estos pasos:

- Abra el sitio web seguro en Internet Explorer 8 (o superior).
- Seleccione **Archivo | Propiedades** y haga clic en el botón **Certificados**.
- Haga clic en **Instalar certificado** e inicie el asistente de importación de certificados. (Este asistente también se puede abrir desde **Herramientas | Opciones de Internet | Contenido | Certificados | Importar**.)
- El certificado debería estar en el almacén "Entidades de certificación raíz de confianza", que se puede buscar manualmente.
- Siga los pasos restantes del asistente y cierre los cuadros de diálogo Certificados y Propiedades haciendo clic en **Aceptar** en cada uno de ellos. Quizás sea necesario reiniciar Internet Explorer.

Nota: no recomendamos cambiar la configuración de la acción SOAP si no tiene acceso a la lista completa de métodos SOAP y de sus acciones SOAP.

Información de seguridad de servicio web

Algunos servicios web exigen autenticar a los usuarios. La capa de seguridad del servicio web complementa a la capa de seguridad HTTP que implementa el servidor. La información de autenticación del servicio web se almacena en la solicitud SOAP en forma de fragmento XML con una estructura como la que aparece más

© 2016-2022 Altova GmbH

abajo. Este fragmento XML se genera automáticamente en la solicitud SOAP a partir de la información de autenticación de servicio web que se introdujo en el cuadro de diálogo "Configuración de seguridad del servicio web".

```
<wsse:Security xmlns:wsse="…" xmlns:wsu="…" SOAP-ENV:mustUnderstand="true">
   <wsse:UsernameToken>
      <wsse:Username>usr</wsse:Username>
      <wsse:Password Type="...#PasswordText">pwd</wsse:Password>
      <wsse:Nonce EncodingType="…#Base64Binary">UqrtD963797WBRgWiJPu2w==</wsse:Nonce>
      <wsu:Created>2014-11-17T16:08:07.016Z</wsu:Created>
   </wsse:UsernameToken>
   <wsu:Timestamp>
      <wsu:Created>2014-11-17T16:08:07.016Z</wsu:Created>
      <wsu:Expires>2014-11-17T16:09:07.016Z</wsu:Expires>
   </wsu:Timestamp>
</wsse:Security>
```

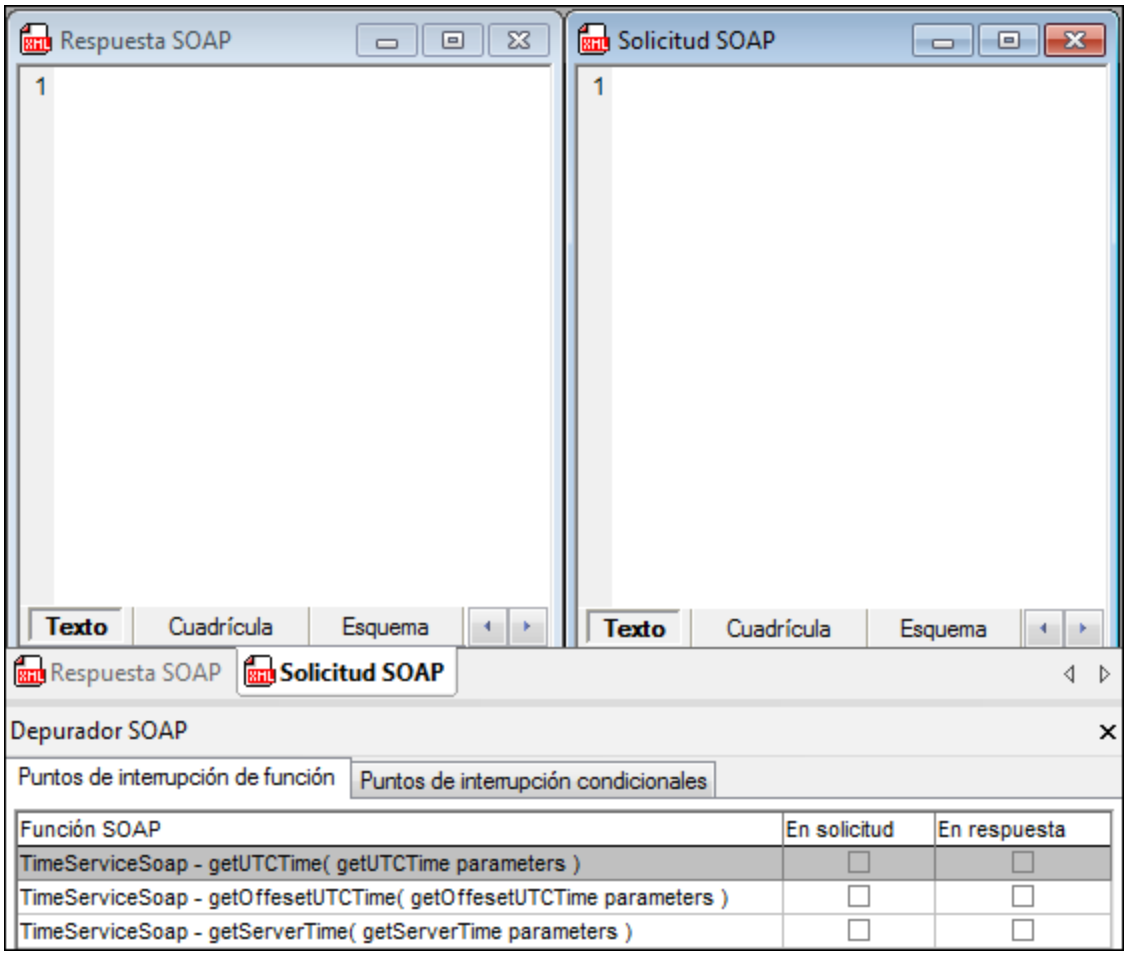
28.15.4 Sesión de depurador SOAP

Este comando inicia una sesión del depurador SOAP.

- Al hacer clic en este comando aparece un cuadro de diálogo donde debe seleccionar la ubicación del archivo WSDL (que suele ser una URL). También puede seleccionar un archivo a través de un recurso global (haga clic en el botón Recursos globales y seleccione un recurso global en el cuadro de diálogo que aparece) o un archivo de las ventanas que están abiertas en XMLSpy.
- Aparece otro cuadro de diálogo. Seleccione los puertos de origen y destino necesarios para el servidor proxy del depurador y para el servicio web. Haga clic en **Aceptar**.

  Se abre el servidor proxy del depurador SOAP en su estado inactivo. Al hacer clic en uno de los iconos de la barra de herramientas SOAP se inicia el depurador y este espera las respuestas del cliente.

  Consulte la sección SOAP 788 del *Manual del usuario* para más información sobre cómo usar el depurador.



28.15.5 Ejecutar



Este comando activa el servidor proxy SOAP y procesa el archivo WSDL hasta que encuentra un punto de interrupción. El documento SOAP correspondiente aparece en una de las ventanas de documentos SOAP.

28.15.6 Paso a paso



© 2016-2022 Altova GmbH

Este comando sirve para recorrer paso a paso las solicitudes y respuestas SOAP entrantes y salientes. El depurador SOAP se detiene en cada solicitud y respuesta. Además, se inicia el servidor proxy si estaba inactivo.

28.15.7 Interrumpir en la siguiente solicitud



Este comando hace que el depurador se detenga en la siguiente solicitud SOAP y muestre los datos en la ventana de documento *Solicitud SOAP*. Los datos se pueden editar en esta ventana directamente antes de enviarse al servicio web.

28.15.8 Interrumpir en la siguiente respuesta



Este comando hace que el depurador se detenga en la siguiente respuesta SOAP y muestre los datos en la ventana de documento *Respuesta SOAP*. Los datos se pueden editar en esta ventana directamente antes de enviarse al cliente.

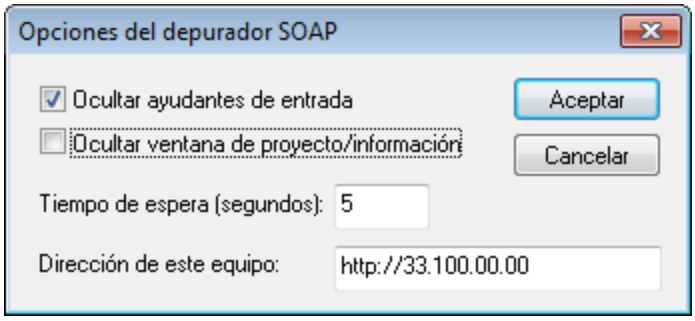
28.15.9 Detener el servidor proxy



Este comando detiene el servidor proxy del depurador.

28.15.10 Opciones del depurador SOAP

Este comando abre el cuadro de diálogo "Opciones del depurador SOAP" donde puede indicar la dirección IP del equipo y varias opciones de depuración más.



- *Dirección de este equipo:* se trata de la dirección del servidor proxy desde el que se ejecuta el depurador. El depurador toma solicitudes de equipos de la red y las envía al servicio web. Como el depurador se ejecuta dentro de XMLSpy, el equipo donde está instalado XMLSpy también hace de servidor proxy. La dirección IP del equipo se detecta e inserta automáticamente en este campo. Es decir, solamente deberá introducir la dirección IP (como dirección HTTP) si XMLSpy no es capaz de detectarla automáticamente. Para averiguar la dirección IP del equipo abra una ventana del símbolo del sistema, introduzca el comando `ipconfig /all` y pulse **Entrar**.
- *Tiempo de espera (segundos):* tiempo que permanece el depurador en un punto de interrupción. El valor predeterminado es 5 segundos.
- *Ocultar ayudantes de entrada y Ocultar ventana de proyecto/información:* estas opciones sirven para obtener más espacio en pantalla para la ventana del depurador SOAP.

© 2016-2022 Altova GmbH

28.16 Menú XBRL

Los comandos del menú **XBRL** (*ver imagen*) se habilitan cuando se abre una taxonomía en la vista XBRL. Los comandos del menú XBRL se describen en los siguientes apartados.

- **Funciones de arco**[1515]**:** para definir funciones de arco
- **Funciones de vínculo**[1517]**:** para definir funciones de vínculo
- **Prefijos de espacios de nombres**[1519]**:** para gestionar los espacios de nombres de la taxonomía
- **Establecer espacio de nombres de destino**[1519]**:** para definir y declarar el espacio de nombres de destino de la taxonomía
- **Valores de parámetro**[1520]**:** muestra los parámetros de fórmula y los parámetros de tabla en un cuadro de diálogo, donde se pueden editar.
- **Importar o hacer referencia**[1521]**:** para importar una taxonomía XBRL o hacer referencia a una base de enlaces
- **Buscar componente de fórmula por su Id**[1523]: busca componentes de fórmula en función del Id. dado por el usuario
- **Generar documentación**[1523]**:** para generar documentación de la taxonomía XBRL activa
- **Configurar la vista**[1527]**:** para establecer la configuración predeterminada de la vista XBRL
- **Generar XBRL con MapForce a partir de BD, Excel, CSV**[1528]**:** inicia Altova MapForce para generar un archivo de instancia XBRL
- **Presentar XBRL como HTML, PDF o Word con StyleVision**[1528]**:** inicia Altova StyleVision para diseñar un informe XBRL
- **Ejecutar fórmula:**[1528] para ejecutar fórmulas y aserciones desde la DTS asociada con el documento de instancia XBRL activo.
- **Generar tabla**[1531]**:** para generar tablas XBRL a partir de un documento de instancia XBRL.
- **Detectar duplicados**[1533]**:** detecta hechos duplicados en instancias XBRL.
- **Ejecutar XULE**[1534]**:** ejecuta un documento XULE en un documento de instancia XBRL.
- **Validar EDGAR en el servidor**[1536]**:** Valida instancias XBRL con EDGAR en RaptorXML+XBRL Server
- **Transformar Inline XBRL**[1535]**:** para transformar la parte Inline XBRL de un documento XHTML en datos XBRL.
- **Opciones de procesamiento**[1536]**:** permite configurar las opciones del procesamiento de instancias XBRL.

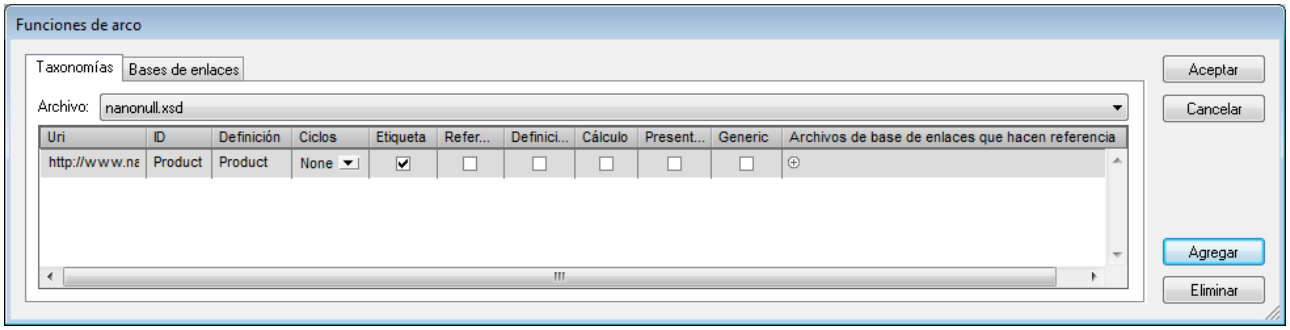
Para más información consulte las secciones XBRL[817] y Vistas de edición | Vista XBRL[264].

Para información sobre la conversión en formatos OIM xBRL, consulte el Menú Conversión[1452].

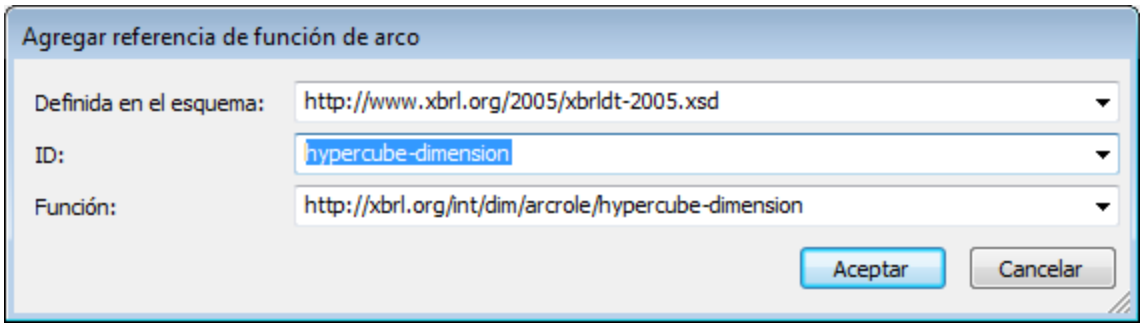
28.16.1 Funciones de arco

El comando **Funciones de arco** abre el cuadro de diálogo del mismo nombre (*imagen siguiente*), donde puede crear funciones de arco para una taxonomía. Las funciones de arco se almacenan en el archivo de definiciones de conceptos, dentro del elemento `appinfo`. Especifican la función de un arco.

En la pestaña ***Taxonomías*** del cuadro de diálogo "Funciones de arco" (*imagen siguiente*) hay un cuadro combinado que enumera las taxonomías que se pueden editar o que contienen una función de arco o función de vínculo. Para añadir una función de arco a la taxonomía, haga clic en el botón **Agregar**. Después defina el *URI*, el *ID*, la *definición* y los *ciclos* de la función de arco. Para definir en qué clase de relaciones debe estar disponible el arco, marque la casilla de la relación pertinente. En la columna *Archivos de base de enlaces que hacen referencia* puede añadir las bases de enlace que hacen referencia a una función de arco.



La pestaña ***Bases de enlaces*** ofrece otra vista de las funciones de arco de la taxonomía. En esta vista puede añadir y ver funciones de arco según la base de enlaces. Seleccione una base de enlaces en el cuadro combinado y añada o elimine funciones de arco a la base de enlaces. Al hacer clic en el botón **Agregar** aparece el cuadro de diálogo "Agregar referencia de función de arco" (*imagen siguiente*).

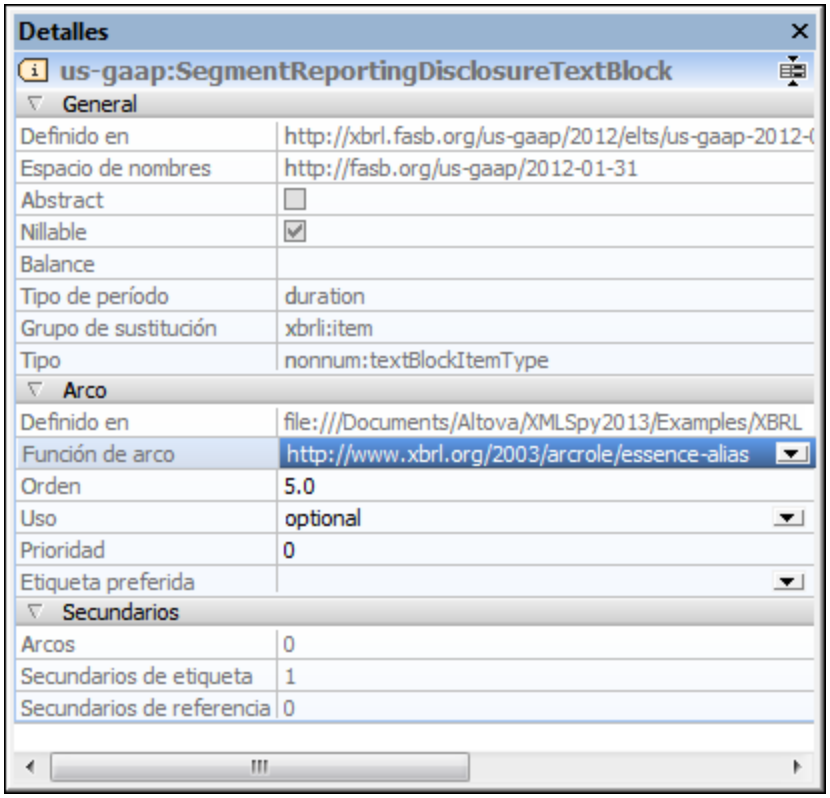


Este cuadro de diálogo contiene tres cuadros combinados. En el campo *Definida en el esquema* puede seleccionar la taxonomía en la que está definida la función de arco. Los campos *ID* y *Función* incluyen las funciones de arco disponibles. Seleccione la función de arco y haga clic en **Aceptar**. La referencia se añade a la base de enlaces. En la pestaña ***Taxonomías***, la función de arco a la que acaba de hacer referencia muestra la base de enlaces de referencia en la columna *Archivos de base de enlaces que hacen referencia*. Si desea que esta función de arco esté disponible para determinado tipo de relación, marque la casilla correspondiente.

Una vez creada, puede usar la función de arco para especificar la función que tiene un arco en el tipo de relación para el que está disponible según su definición. En la imagen siguiente, por ejemplo, la función de arco está disponible para los arcos de las relaciones de etiqueta.

La función de arco de un arco se selecciona en el ayudante de entrada Detalles (*imagen siguiente*).

© 2016-2022 Altova GmbH



Seleccione el elemento del extremo `to` del arco en el ayudante de entrada Detalles y seleccione una opción de la lista desplegable de la entrada *Función de arco*.

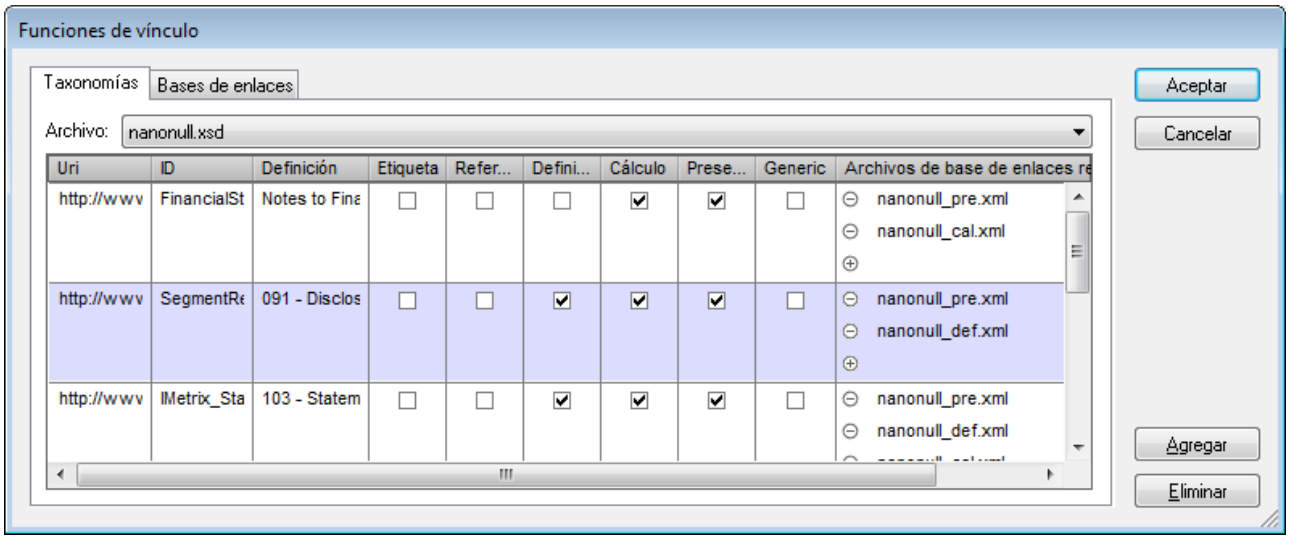
28.16.2 Funciones de vínculo

El comando **Funciones de vínculo** abre el cuadro de diálogo del mismo nombre (*imagen siguiente*), donde puede crear funciones de vínculo para una taxonomía. Las funciones de vínculo se almacenan en el archivo de definiciones de conceptos, dentro del elemento `appinfo` (*véase el ejemplo de código siguiente*). Las funciones de vínculo no solamente se usan en los elementos `definitionLink`, también se utilizan en los elementos contenedores de otros tipos de relación (como en los elementos `calculationLink` y `presentationLink`, por ejemplo).

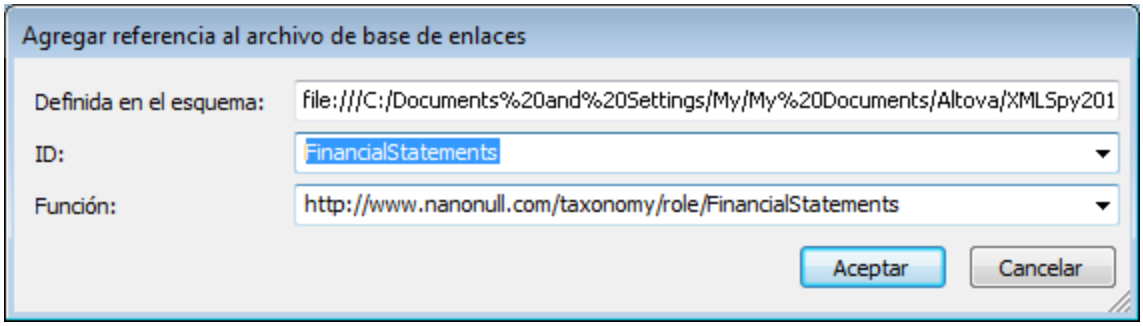
```
<xs:appinfo>
   <link:roleType id="SegmentRevenueAndOperatingIncome"
   roleURI="http://www.nanonull.com/taxonomy/role/SegmentRevenueAndOperatingIncome">
      <link:definition>006091 - Disclosure - Segment Revenue and Operating
Income</link:definition>
      <link:usedOn>link:calculationLink</link:usedOn>
      <link:usedOn>link:definitionLink</link:usedOn>
      <link:usedOn>link:presentationLink</link:usedOn>
   </link:roleType>
</xs:appinfo>
```

Observe cómo en el ejemplo de código anterior los elementos `usedOn` especifican en qué tipo de relación se puede utilizar esta función de vínculo.

En la pestaña ***Taxonomías*** del cuadro de diálogo "Funciones de vínculo" (*imagen siguiente*) puede añadir una función de vínculo a un archivo de taxonomía haciendo clic en el botón **Agregar**. Después puede definir el *URI* y el *ID* de la función de vínculo (*véase el ejemplo de código anterior*). Para definir en qué tipo de relación puede estar disponible la función de vínculo, marque las casillas correspondientes. En la columna *Archivos de base de enlaces que hacen referencia* puede añadir/eliminar las bases de enlace que hacen referencia a cada función de vínculo.



La pestaña ***Bases de enlaces*** ofrece otra vista de las funciones de vínculo de la taxonomía. En esta vista puede añadir y ver funciones de vínculo según la base de enlaces. Seleccione una base de enlaces en el cuadro combinado y añada o elimine funciones de vínculo a la base de enlaces. Al hacer clic en el botón **Agregar** aparece el cuadro de diálogo "Agregar referencia al archivo de base de enlaces" (*imagen siguiente*).



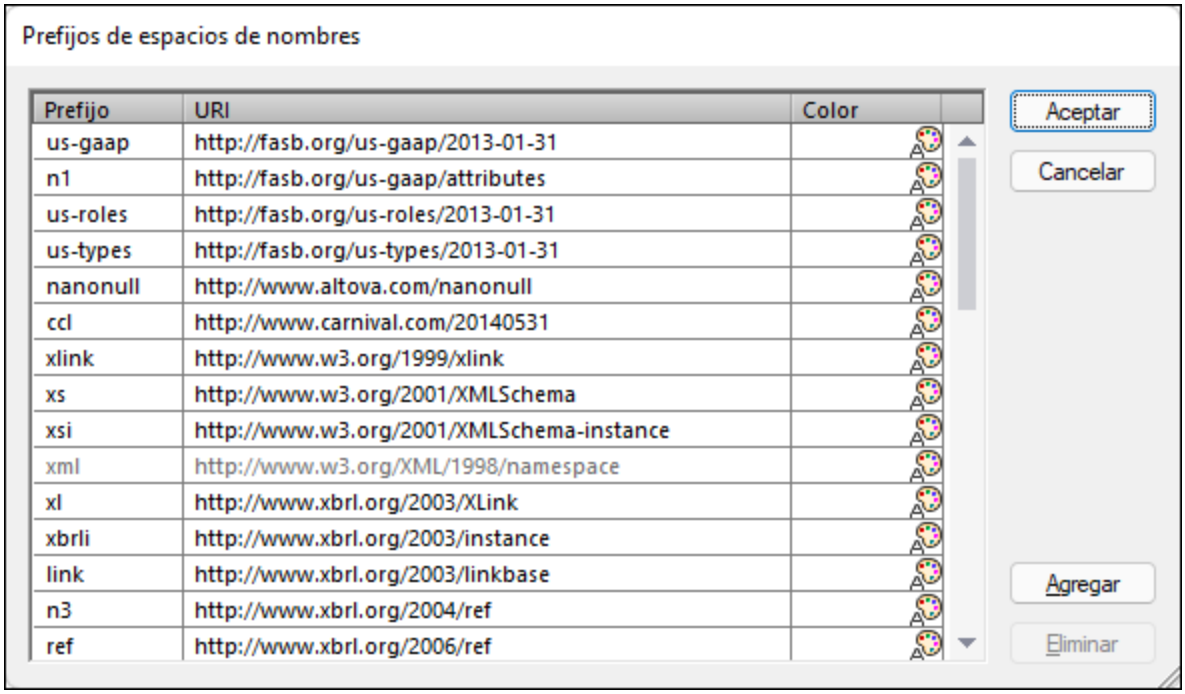
Este diálogo incluye tres cuadros combinados. En el campo *Definida en el esquema* puede seleccionar la taxonomía en la que está definida la función de vínculo. Los cuadros combinados *ID* y *Función* ofrecen las funciones de vínculo disponibles. Después de seleccionar la función de vínculo, haga clic en **Aceptar** y la referencia se añade a la base de enlaces. En la pestaña ***Taxonomías***, la función de vínculo a la que hizo referencia muestra ahora la base de enlaces de referencia en la columna *Archivos de base de enlaces que hacen referencia.* Si desea que esta función de vínculo esté disponible para determinado tipo de relación, marque la casilla correspondiente.

© 2016-2022 Altova GmbH

Una vez creada, la función de vínculos se puede utilizar para crear relaciones[855].

28.16.3 Prefijos de espacios de nombres

El comando **Prefijos de espacio de nombres** abre el cuadro de diálogo del mismo nombre (*imagen siguiente*), que enumera todos los espacios de nombre de la taxonomía, incluidos los de las taxonomías importadas. En el cuadro de diálogo "Prefijos de espacios de nombres" puede editar los espacios de nombres y los prefijos y definir su color de fondo. Si define un color de fondo para un espacio de nombres, los elementos que estén en ese espacio de nombres aparecen con dicho color de fondo tanto en la ventana principal como en los ayudantes de entrada. Recuerde que el color de fondo elegido se utiliza en todas las taxonomías que estén abiertas en la vista XBRL.



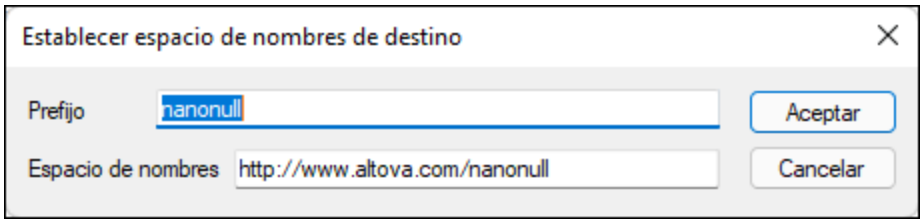
Haga clic en los botones **Agregar** y **Eliminar** para añadir y eliminar espacios de nombres respectivamente. El color de fondo se asigna haciendo clic en el icono en forma de paleta de colores. Cuando termine de configurar los espacios de nombres, haga clic en **Aceptar**.

El espacio de nombres de destino de la taxonomía también aparece en este cuadro de diálogo. No obstante, para modificar el espacio de nombres de destino debe utilizar el comando **Establecer espacio de nombres de destino**[1519]. Para más información consulte la sección XBRL[843] del *Manual del usuario*.

28.16.4 Establecer espacio de nombres de destino

El comando **Establecer espacio de nombres de destino** sirve para definir el espacio de nombres de destino del documento de taxonomía. Al hacer clic en este comando se abre el cuadro de diálogo del mismo nombre

(*imagen siguiente*), donde puede introducir el espacio de nombres de destino elegido y su prefijo. Para terminar haga clic en **Aceptar**.



Como resultado se define y declara el espacio de nombres de destino:

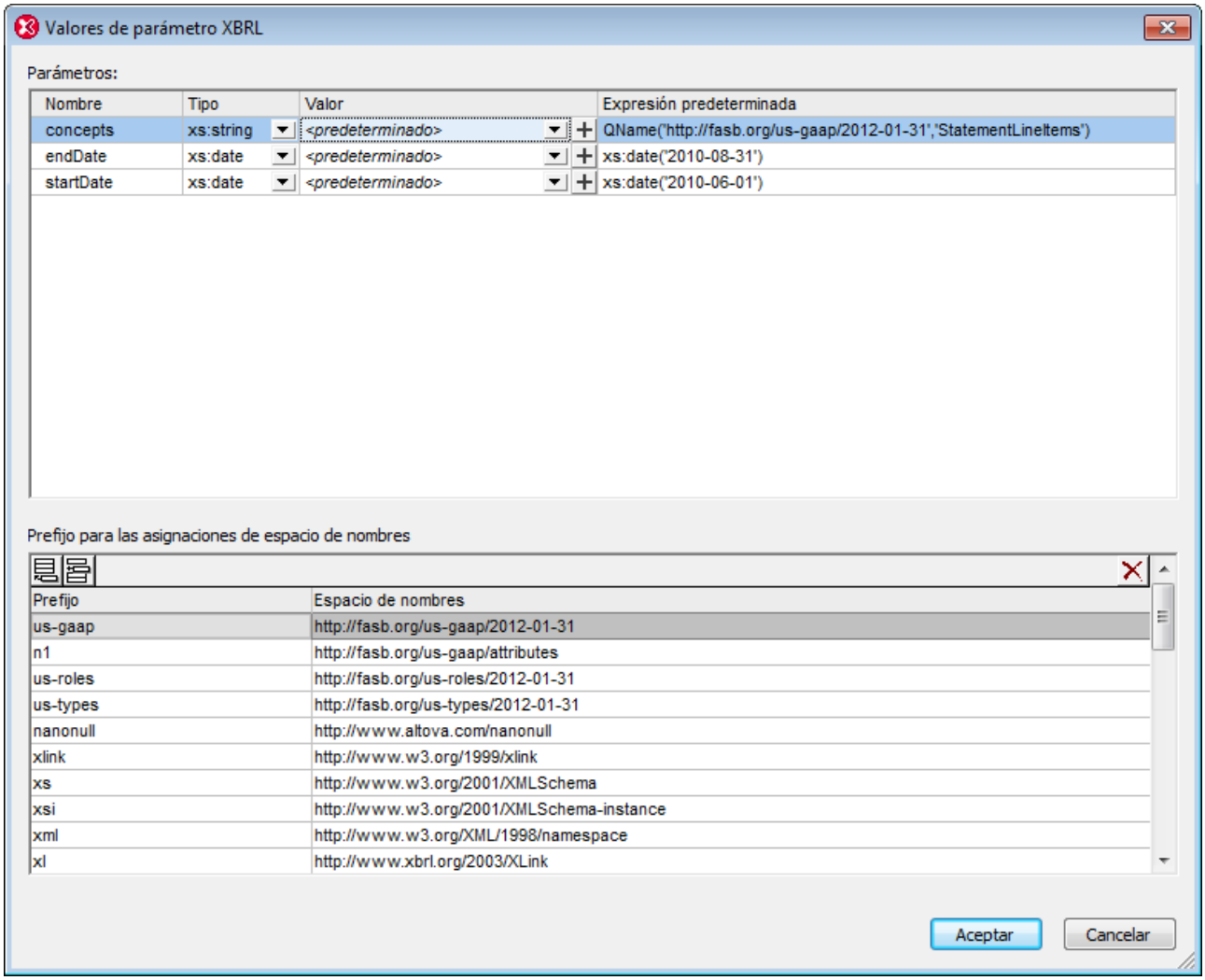
```
<xs:schema targetNamespace="http://www.altova.com/XBRL/Taxonomies"
           xmlns:ns1="http://www.altova.com/XBRL/Taxonomies" >
      ...
</xs:schema>
```

Tal y como muestra el ejemplo de código anterior, el espacio de nombres de destino se define con el atributo `targetNamespace` y se declara con el prefijo `ns1`.

28.16.5 Valores de parámetro

El comando **Valores de parámetro** abre el cuadro de diálogo "Valores de parámetro XBRL" (*imagen siguiente*), que muestra los parámetros definidos en la pestaña **Fórmula** (parámetros de fórmulas) y en la pestaña **Tabla** (parámetros de tablas) del editor de taxonomías XBRL. En este cuadro de diálogo puede editar el tipo de datos de los parámetros y ofrecer un valor que reemplace el valor predeterminado de los parámetros. Como los parámetros de tablas pueden tomar varios parámetros, con el icono **+** de la columna *Valor* puede añadir más valores para el mismo parámetro de tabla.

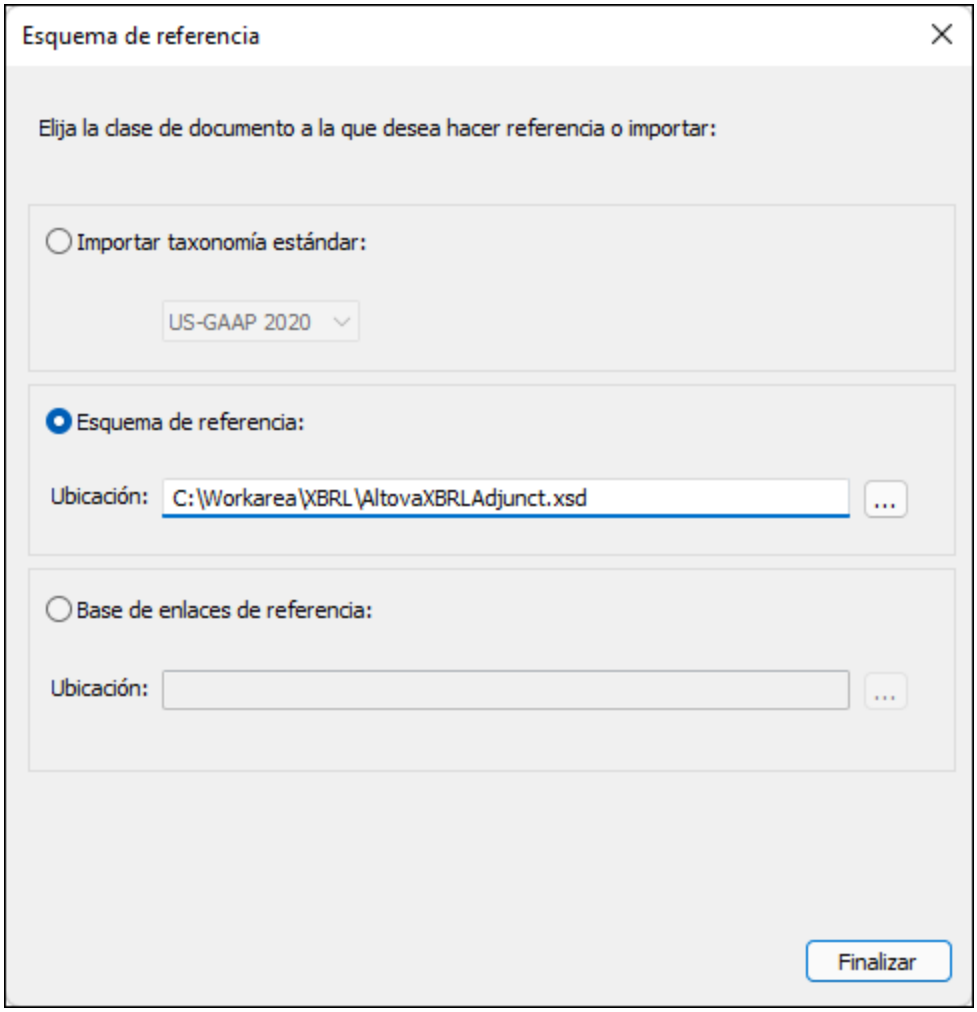
© 2016-2022 Altova GmbH



Los valores de parámetros globales asignados en este cuadro de diálogo solamente se evalúan para los parámetros de tablas. Los parámetros de fórmulas aparecen en el cuadro de diálogo pero no se pueden editar.

28.16.6 Importar o hacer referencia

El comando **Importar o hacer referencia** abre un cuadro de diálogo (*imagen siguiente*) donde puede seleccionar el esquema que desea importar o la base de enlaces a la que desea hacer referencia.



El cuadro de diálogo ofrece tres botones de opción:

- *Importar taxonomía estándar:* para importar una taxonomía US-GAAP o IFRS con facilidad. Seleccione la taxonomía estándar en el menú desplegable del cuadro combinado y haga clic en **Siguiente**. En la siguiente pantalla del diálogo puede seleccionar más opciones de importación. El proceso se describe en el apartado Crear una taxonomía nueva[838].
- *Importar cualquier taxonomía (Esquema de referencia):* para importar cualquier taxonomía con solo especificar la ubicación del archivo de taxonomía (el archivo `.xsd`)

- *Hacer referencia a una base de enlaces:* puede especificar una base de enlaces para incluirla en la taxonomía. Para ello, indique la ubicación del archivo de base de enlaces y haga clic en **Finalizar**. En la taxonomía se crea una referencia al archivo de base de enlaces. El tipo de relación de la base de enlaces a la que acaba de hacer referencia se puede especificar haciendo clic con el botón derecho en el nombre del archivo y seleccionando el comando **Establecer clase de base de enlaces**[845].

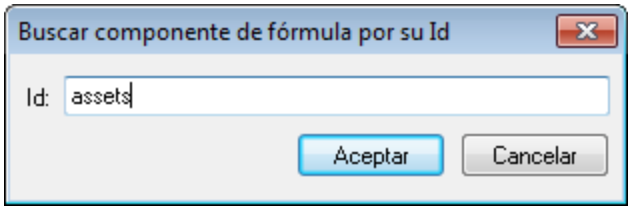
```
<xsl:template match="*">
   <xsl:copy>
      <xsl:copy-of select="@*[not(.='')]"/>
      <xsl:apply-templates/>
```

 © 2016-2022 Altova GmbH

```
    </xsl:copy>
</xsl:template>
```

28.16.7 Buscar componente por su Id

En las taxonomías con bases de enlaces de fórmula de gran tamaño que contienen varios componentes del mismo tipo, lo más conveniente puede ser buscar componentes por su Id. Para ello utilice el comando de menú **XBRL | Buscar componente por su Id**. Al hacer clic en este comando se abre un cuadro de diálogo (*imagen siguiente*) donde puede escribir el Id del componente que desea buscar.



Haga clic en **Aceptar** para iniciar la búsqueda.

28.16.8 Generar documentación

El comando **XBRL | Generar documentación** genera documentación detallada de la taxonomía XBRL activa. La documentación puede generarse en formato HTML, MS Word, RTF y PDF y se puede modificar y utilizar sin restricciones. La documentación incluye información sobre los componentes seleccionados en el cuadro de diálogo "Documentación de la taxonomía XBRL" (que aparece al seleccionar el comando **Generar documentación**). Los componentes relacionados entre sí se enlazan mediante hipervínculos, para que pueda navegar por la documentación de componente en componente. Las opciones de generación de documentación se describen en el subapartado Opciones de documentación[1523].

Nota: para generar la documentación en formato MS Word es necesario tener MS Word (versión 2000 o superior) instalado en el equipo.

Puede usar el diseño fijo de XMLSpy para generar la documentación o puede usar un archivo SPS de StyleVision para personalizar el diseño de la documentación. Para más información, consulte el apartado Diseño definido por el usuario[1502].

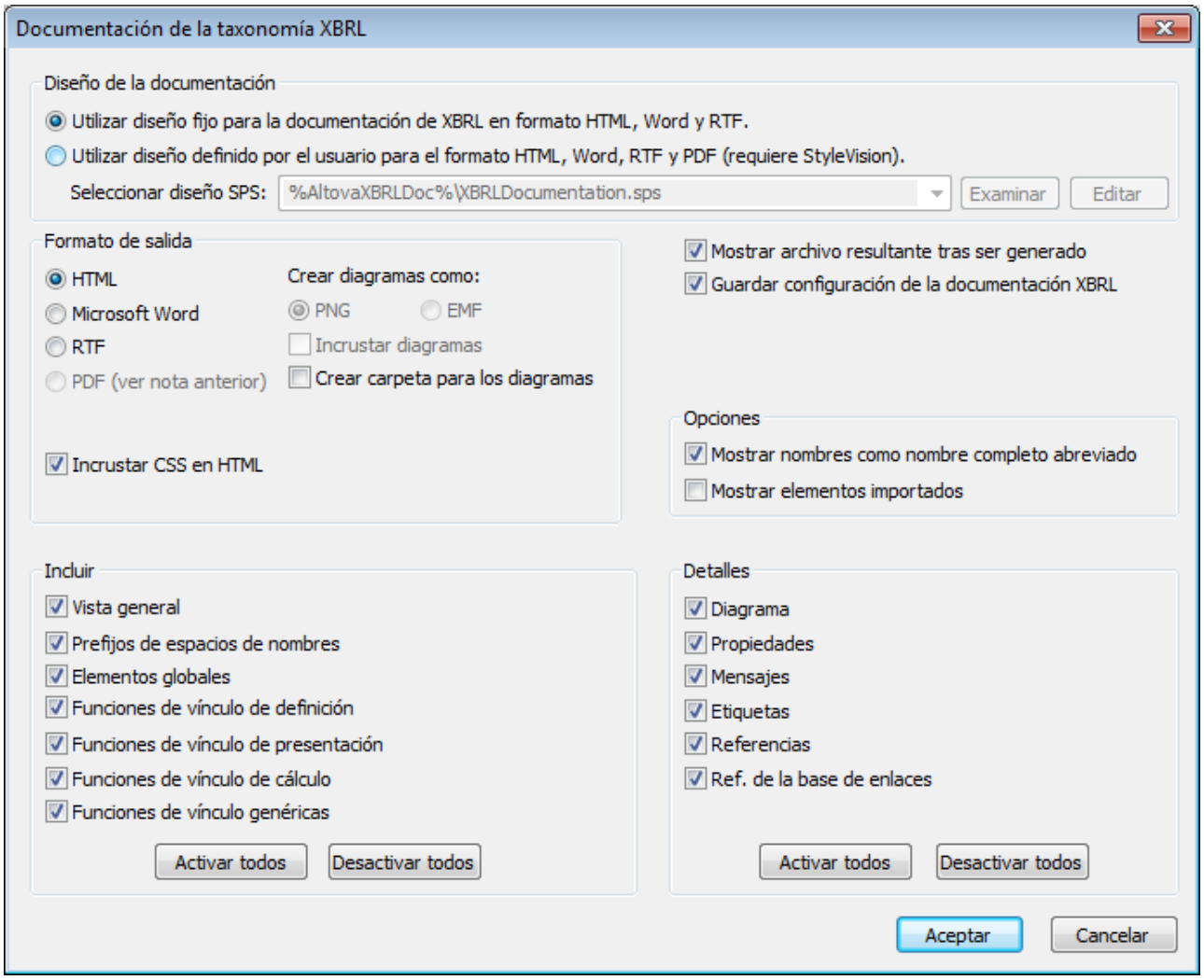
Nota: es necesario tener StyleVision instalado en el equipo para generar la documentación XBRL usando un archivo SPS.

28.16.8.1 Opciones de documentación

Al hacer clic en el comando **XBRL | Generar documentación** aparece el cuadro de diálogo "Documentación de la taxonomía XBRL" (*imagen siguiente*), donde puede seleccionar las opciones de configuración de la documentación.

En el panel *Diseño de la documentación* de este cuadro de diálogo puede elegir si utiliza el diseño fijo de XMLSpy para generar la documentación o si utiliza un diseño personalizado creado en un archivo SPS de StyleVision. Recuerde que la documentación en formato PDF está disponible solamente si utiliza un archivo SPS de StyleVision, pero no si genera la documentación con el diseño fijo. Para más información, consulte el apartado Diseño definido por el usuario[1525].

Nota: es necesario tener StyleVision instalado en el equipo para generar la documentación XBRL usando un archivo SPS.



Al hacer clic en el comando **Generar documentación** aparece el cuadro de diálogo "Documentación de la taxonomía XBRL":

- Puede elegir el formato de la documentación en el panel *Formato de salida:* las opciones disponibles son HTML, Microsoft Word, RTF y PDF. (El formato PDF solamente está disponible si usa un archivo SPS de StyleVision para generar la documentación.) Al hacer clic en **Aceptar** aparece un diálogo donde debe introducir un nombre para el archivo de salida y seleccionar la ubicación donde desea guardarlo.

© 2016-2022 Altova GmbH

- Si usa el diseño fijo para generar la documentación en formato Word, los archivos se guardan con la extensión de archivo `.doc`. Si usa un archivo SPS de StyleVision, entonces se guardan con la extensión de archivo `.docx`.
- Si elige el formato de salida HTML, las definiciones de estilos CSS se pueden guardar en un archivo CSS o se pueden incrustar en el archivo HTML (en el elemento `<head>`). Si prefiere crear un archivo CSS, este lleva el mismo nombre que el archivo HTML, pero con la extensión de archivo `.css`. Elija la opción deseada activando o desactivando la casilla *Incrustar CSS en HTML*.
- La casilla *Incrustar diagramas* se habilita cuando se selecciona el formato de salida MS Word, RTF o PDF. Si marca esta casilla, los diagramas se incrustan en el archivo de la documentación, en formato PNG. Si no marca esta casilla, los diagramas se guardan en archivos PNG y el archivo de la documentación incluye enlaces a ellos por medio de vínculos de objeto.
- Si elige el formato de salida HTML, los diagramas se guardan en archivos PNG. Si marca la casilla *Crear carpeta para los diagramas*, se crea una carpeta con el mismo nombre que el archivo HTML y en ella se guardan todos los archivos PNG. El nombre de la carpeta sigue este modelo: `NombreArchivoHTML_diagrams`. Si no marca la casilla *Crear carpeta para los diagramas*, los archivos PNG se guardan en la misma carpeta que el archivo HTML.
- En el panel *Incluir* puede seleccionar qué componentes comprende la documentación. Si marca la casilla *Vista general*, al principio de la documentación aparece una lista con todos los componentes, ordenados por tipo. Puede usar los botones **Activar todos** y **Desactivar todos** para seleccionar las opciones más rápido.
- El panel *Detalles* enumera los detalles de los componentes que puede incluir la documentación. Puede usar los botones **Activar todos** y **Desactivar todos** para seleccionar las opciones más rápido.
- La casilla *Mostrar archivo resultante tras ser generado* está disponible para todos los formatos de salida. Si marca esta casilla, el archivo resultante se presenta en la vista Explorador (formato HTML), en MS Word (formato MS Word) y en las aplicaciones predeterminadas para archivos `.rtf` (formato RTF) y archivos `.pdf` (formato PDF).
- En el panel *Opciones* puede elegir (i) si el nombre de los elementos aparecen solamente con un prefijo (nombre completo abreviado) o en su formato expandido (con el espacio de nombres completo) y (ii) si los elementos importados también aparecen en la documentación. La casilla *Mensajes* solamente se habilita si está marcada la casilla *Funciones de vínculo genéricas* en el panel *Incluir*. Las demás casillas se habilitan solamente si en el panel *Incluir* están marcadas las casillas *Elementos globales* y *Funciones de vínculo genéricas*.

Valores de parámetro

Si el archivo SPS de StyleVision contiene definiciones de parámetros, al hacer clic en **Aceptar** aparece un cuadro de diálogo que enumera todos los parámetros definidos en el archivo SPS. En este cuadro de diálogo puede introducir valores nuevos si desea reemplazar los valores predeterminados que se asignaron en el archivo SPS.

28.16.8.2 Diseño definido por el usuario

Si lo prefiere, puede crear un diseño personalizado para la documentación de la taxonomía XBRL en lugar de usar el diseño fijo estándar de XMLSpy. Este diseño personalizado se crea en un archivo SPS de StyleVision, que no es más que una plantilla de diseño para el documento de salida.

Cómo crear un archivo SPS

El archivo SPS (o StyleVision Power Stylesheet) se crea con Altova StyleVision. El archivo SPS necesario para generar documentación XBRL debe estar basado en un esquema XML que defina la estructura de la

documentación XBRL. Se trata del esquema `XBRLDocumentation.xsd` que viene con XMLSpy. Está guardado en la carpeta `C:\Documents and Settings\<nombreUsuario>\Mis Documentos\Altova\XMLSpy2023\Documentation\XBRL`.

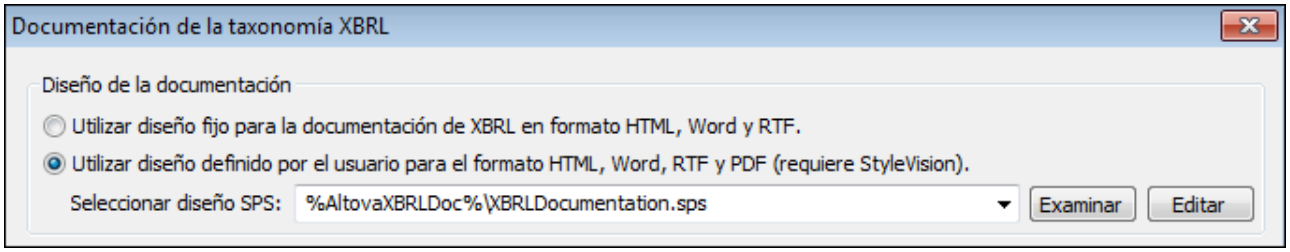
El proceso de creación de un diseño SPS en StyleVision consiste en colocar nodos del esquema `XBRLDocumentation.xsd` en la plantilla del diseño y asignarles estilos y propiedades. Después se añaden al diseño componentes adicionales como vínculos, tablas, imágenes, etc. Para más información sobre cómo crear un diseño SPS consulte el *Manual del usuario de StyleVision*.

La ventaja de usar un SPS a la hora de generar la documentación de la taxonomía XBRL es que puede personalizar totalmente el aspecto de la documentación. Además, recuerde que la documentación se puede generar en formato PDF solamente si usa un archivo SPS. La documentación no se puede generar en formato PDF si usa el diseño fijo de XMLSpy.

Cómo elegir el SPS para generar la documentación de la taxonomía XBRL

Una vez creado, el archivo SPS se puede usar para generar la documentación de la taxonomía. Abra el cuadro de diálogo "Documentación de la taxonomía XBRL" (con el comando **XBRL | Generar documentación**) y seleccione el botón de opción *Utilizar diseño definido por el usuario.* Después haga clic en el botón **Examinar** y busque el SPS que desea usar. Para terminar haga clic en el botón **Aceptar** del diálogo y en el cuadro de diálogo que aparece seleccione una carpeta y un nombre para el archivo de salida.

Nota: el archivo SPS debe localizar correctamente el esquema en el que se basa: `XBRLDocumentation.xsd` (*véase más arriba*).



XMLSpy viene con un diseño SPS editable para generar documentación XBRL. Este archivo se llama `XBRLDocumentation.sps` y está en la carpeta `C:\Documents and Settings\<nombreUsuario>\Mis Documentos\Altova\XMLSpy2011\Documentation\XBRL\`. Este archivo, junto con los archivos SPS examinados recientemente, están disponibles en el cuadro combinado de la opción *Utilizar diseño definido por el usuario* (*imagen anterior*).

Si hace clic en el botón **Editar** del panel *Diseño de la documentación*, el archivo SPS seleccionado se abre en StyleVision. Si desea obtener una vista previa del archivo resultante en StyleVision, necesitará un archivo XML de trabajo. Puede usar el archivo XML de muestra `nanonull.xml` que viene con XMLSpy. Este archivo está en esta carpeta:

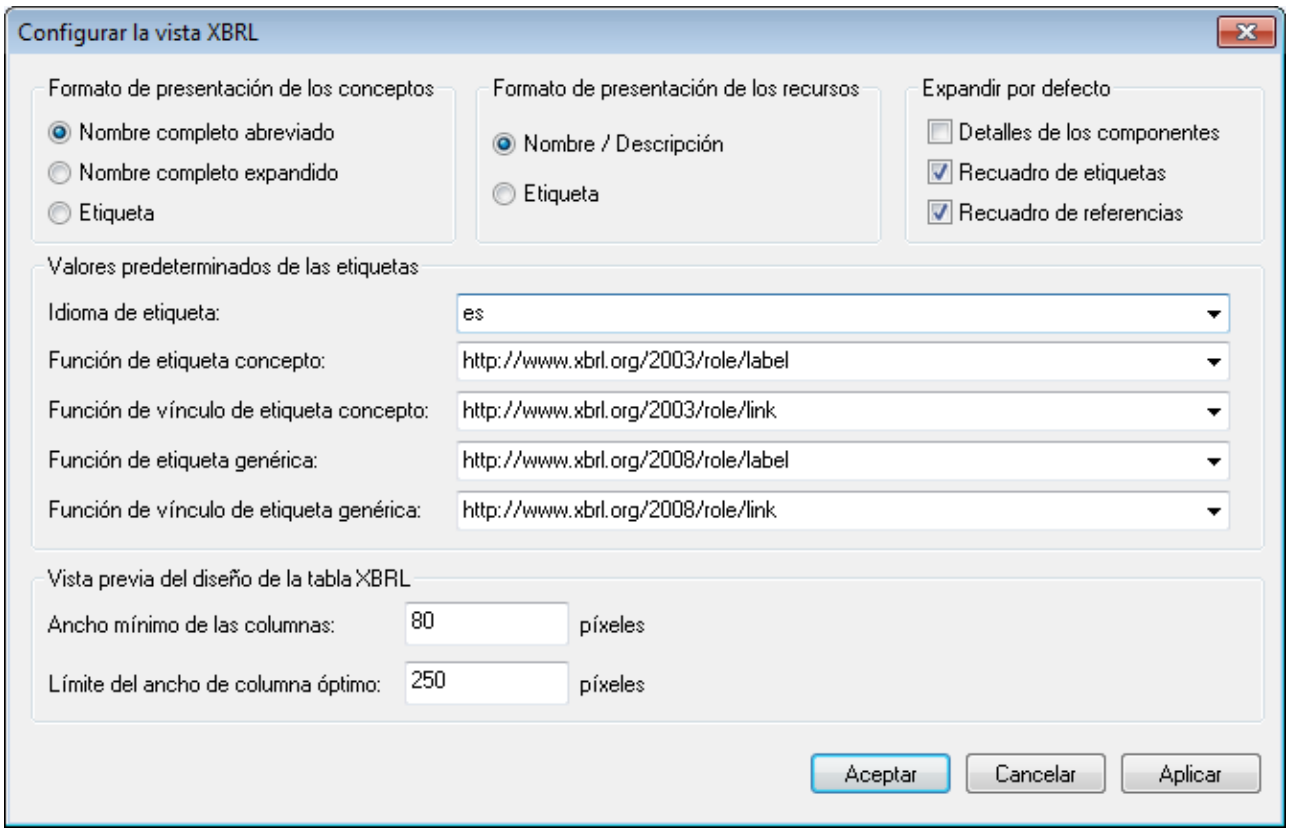
`C:\Documents and Settings\<nombreUsuario>\Mis Documentos\Altova\XMLSpy2023\Documentation\XBRL\SampleData`

Nota: es necesario tener StyleVision instalado en el equipo para generar la documentación de la taxonomía XBRL usando un archivo SPS.

© 2016-2022 Altova GmbH

28.16.9 Configurar la vista

El comando **Configurar la vista** abre el cuadro de diálogo "Configurar la vista XBRL" (*imagen siguiente*), donde puede definir las opciones de configuración predeterminadas para la vista XBRL.



Estas son las opciones de configuración:

- ***Formato de presentación de los conceptos:*** los elementos se pueden presentar con su nombre completo abreviado, su nombre completo expandido o su etiqueta. Este formato de presentación afecta a la ventana principal y al ayudante de entrada Detalles, pero no al ayudante de entrada Elementos globales. . El formato de presentación de los nombres de elemento en el ayudante de entrada Elementos globales se configura en la barra de menú del ayudante de entrada[274].

- ***Formato de presentación de los recursos:*** en las pestañas Fórmula y Tabla los recursos se pueden presentar por su nombre o con una etiqueta (o con una descripción del recurso si no se asignó ningún nombre).

- ***Expandir por defecto:*** en la ventana principal puede expandir por defecto los detalles de los elementos, los recuadros de las etiquetas y los recuadros de las referencias. No obstante, si los recuadros de las etiquetas o referencias se expanden por defecto, los recuadros expandidos solo pueden ser visibles si están expandidos los detalles del elemento (automática o manualmente). Cada vez que se actualiza la vista (p. ej. al cambiar a la vista **Texto** y volver a la vista **XBRL**), la vista XBRL vuelve a la configuración predeterminada.

- ***Valores predeterminados de las etiquetas:*** indica el idioma predeterminado y las funciones de etiqueta predeterminadas que se deben usar si no se definieron etiquetas. El cuadro combinado de cada propiedad incluye una lista de posibles valores.

- ***Vista previa del diseño de la tabla XBRL:*** aquí puede configurar el ancho mínimo y máximo de las columnas de la tabla (en píxeles).

28.16.10 Generar XBRL con MapForce a partir de BD, Excel, CSV

Este comando inicia la generación de un archivo de instancia XBRL basado en la taxonomía activa. Los datos que se utilizan para el archivo de instancia se obtienen de una hoja de datos MS Excel, de una BD o de un archivo CSV. El archivo de instancia XBRL se genera con la herramienta Altova MapForce, que debe estar instalada en el equipo. El comando **Generar XBRL con MapForce a partir de BD, Excel, CSV** inicia Altova MapForce y carga la taxonomía en esta aplicación. Después puede especificar el archivo de datos de origen y diseñar de forma gráfica el archivo de instancia de salida.

28.16.11 Presentar XBRL como HTML, PDF o Word con StyleVision

Este comando carga la taxonomía activa en Altova StyleVision, donde puede generar un diseño para crear informes basados en la taxonomía. Para poder usar este comando es necesario tener la herramienta Altova StyleVision instalada en el equipo.

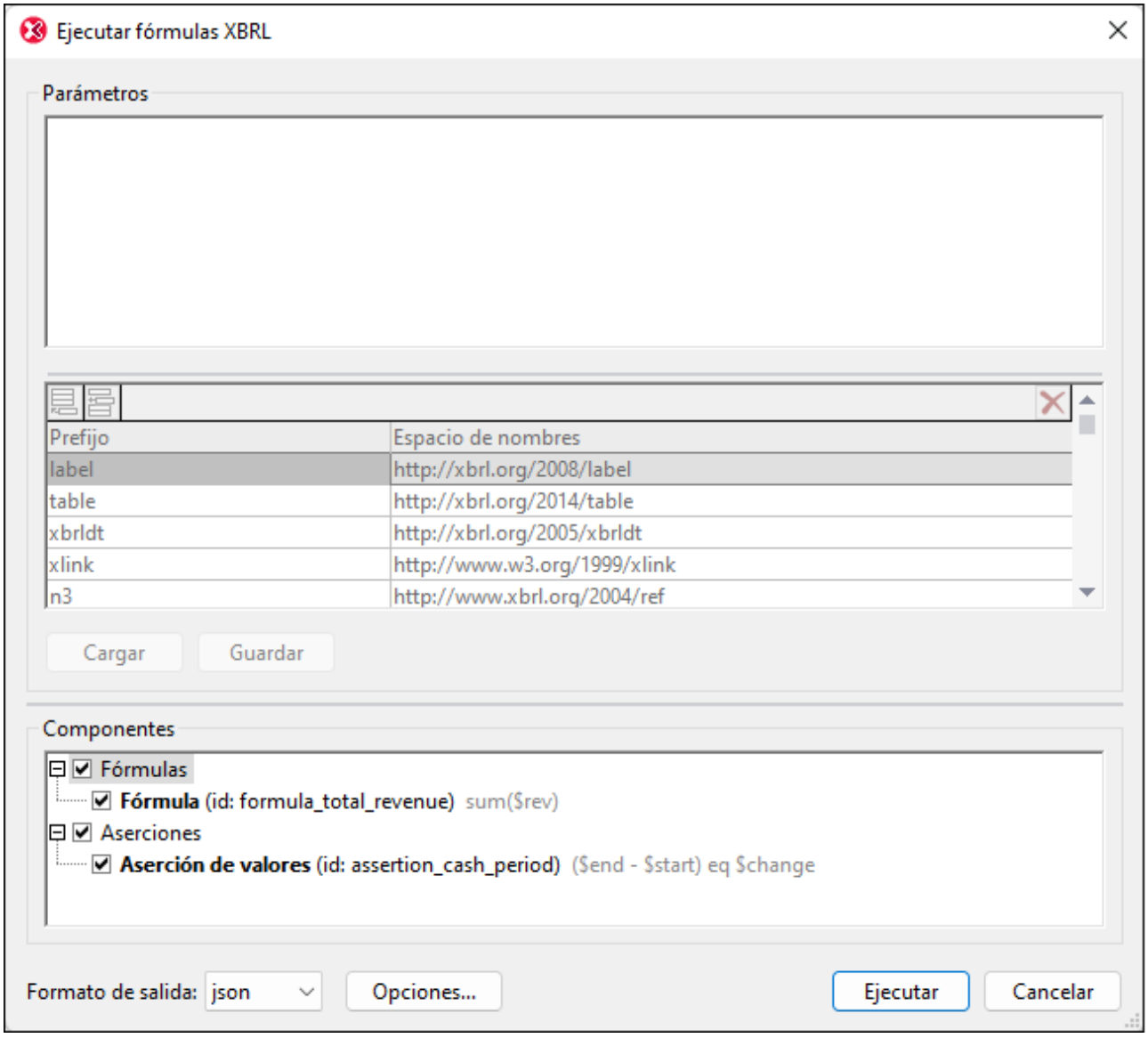
28.16.12 Ejecutar fórmula (en el servidor)

Los comandos **Ejecutar fórmula** y **Ejecutar fórmula en el servidor (alto rendimiento)** se habilitan cuando el documento activo de la vista **Texto** o **Cuadrícula** es un documento XBRL de instancia (`.xbrl`). Estos comandos ejecutan las fórmulas y aserciones definidas en la DTS asociada al archivo de instancia XBRL (una DTS, *Discoverable Taxonomy Set*, es una colección de taxonomías). Las fórmulas se evalúan con los datos del archivo XBRL de instancia y los resultados se exportan a un archivo de instancia XBRL. Las aserciones se evalúan por separado y los resultados se exportan a un archivo JSON o XML.

El comando **Ejecutar fórmula en el servidor (alto rendimiento)** utiliza un servidor RaptorXML+XBRL Server asociado[1068] para ejecutar las fórmulas. Utilice el comando **Herramientas | Gestionar servidores Raptor**[1559] para configurar un servidor RaptorXML+XBRL Server.

Si no hay fórmulas ni aserciones en la DTS, aparece un mensaje a tal efecto. Si en la DTS hay una fórmula o aserción válida, entonces aparece el cuadro de diálogo "Ejecución XBRL" (*imagen siguiente*).

© 2016-2022 Altova GmbH



Parámetros
Si hay parámetros definidos en la DTS, aparecerán en el panel *Parámetros,* donde puede introducir un valor para cada uno de ellos. Los nombres de parámetro son de solo lectura. Los parámetros obligatorios están marcados con un signo de exclamación rojo y el botón **Aceptar** no se habilita hasta que se asigne un valor al parámetro. Los parámetros opcionales tienen un valor predeterminado. Si se especifica un tipo necesario, aparece. Los parámetros que necesitan varios valores se marcan con el icono **+** (haga clic en este icono para añadir un valor nuevo). Recuerde que los parámetros opcionales que no tengan un valor no se pasarán al motor. Los valores predeterminados son de solo lectura y se ejecutarán si el usuario no introduce un valor distinto.

Asignaciones de espacios de nombres
En el panel central del cuadro de diálogo aparece una tabla donde puede ver qué prefijos se usan en los QName de los parámetros y tipos. Aquí puede definir los espacios de nombres que se deben usar en la evaluación de parámetros.

Guardar y cargar los parámetros
La configuración de los parámetros, incluidas las asignaciones de espacios de nombres, se puede guardar en formato JSON o XML haciendo clic en el botón **Guardar**. El formato de archivo depende de la extensión dada al archivo. No olvide que los parámetros opcionales que no tengan un valor no se guardan. Una vez guardado, el archivo de parámetros puede cargarse al cuadro de diálogo con el botón **Cargar**.

Opciones de procesamiento XBRL
El botón **Opciones** abre el cuadro de diálogo "Opciones de procesamiento XBRL" 1536, en el que puede activar la deduplicación (ignorar automáticamente los hechos duplicados).

Componentes

Este panel ofrece una vista en forma de árbol donde puede seleccionar los componentes de la fórmula/aserción que se deben ejecutar. Cada componente va acompañado de un icono y una descripción y, si están disponibles, el ID y la expresión. Marque la casilla del componente que desea ejecutar. El resultado de la ejecución de aserciones puede generarse en formato JSON o XML (seleccione el formato de salida en el cuadro combinado *Formato de salida:*). El formato de las ejecuciones de fórmulas XBRL siempre es XML.

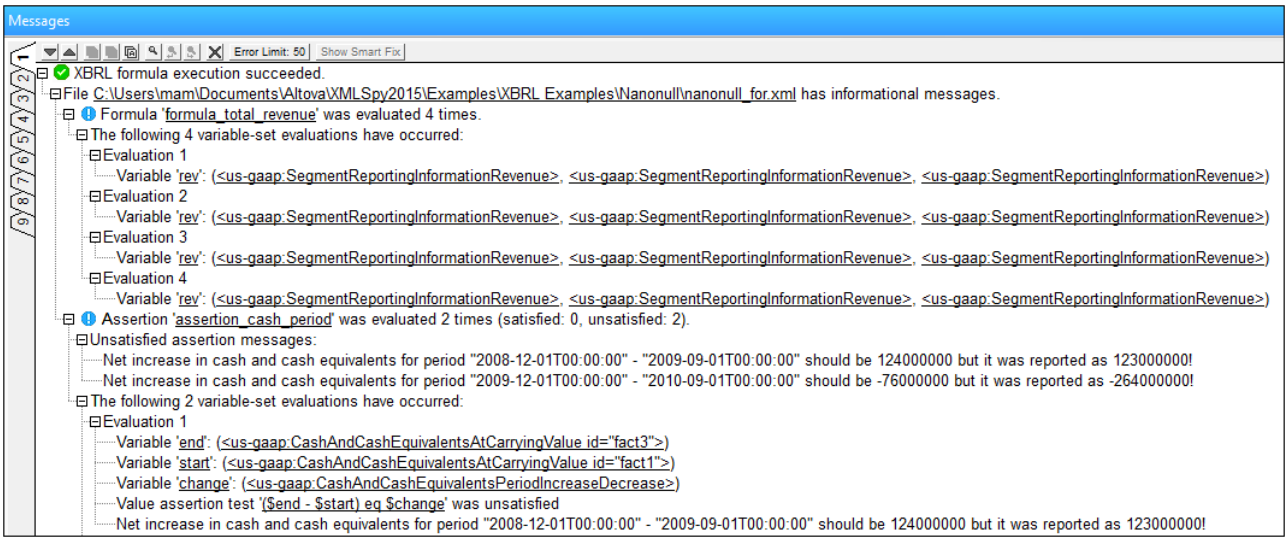
Ejecución

Si se produce un error de ejecución, en la ventana Mensajes aparece un mensaje de error a tal efecto. De lo contrario aparece un mensaje confirmando que la ejecución finalizó correctamente. Los archivos de salida `assertions-ouput-file.xml/json` y `formula-output-file.xml` se abren en ventanas de documento nuevas, pero no se guardan en disco.

Trace

If you select **<trace>** in the *Output Format* combo box (at the bottom left of the dialog), extra debug information for all “variable set evaluations” will be collected during the formula execution and then be displayed in the Messages window (*see screenshot below*). The trace lists the individual variable set evaluations for each formula/assertion at the points where the actual assignment of the variables in that evaluation are displayed. If the variables reference instance facts, clicking on the values takes you to the corresponding fact element in the instance. Clicking on the formula/assertion or variable name will take you to the corresponding definition in the formula linkbase files. In the case of validation assertions, the assertion messages that have been generated in that evaluation step are also displayed.

© 2016-2022 Altova GmbH



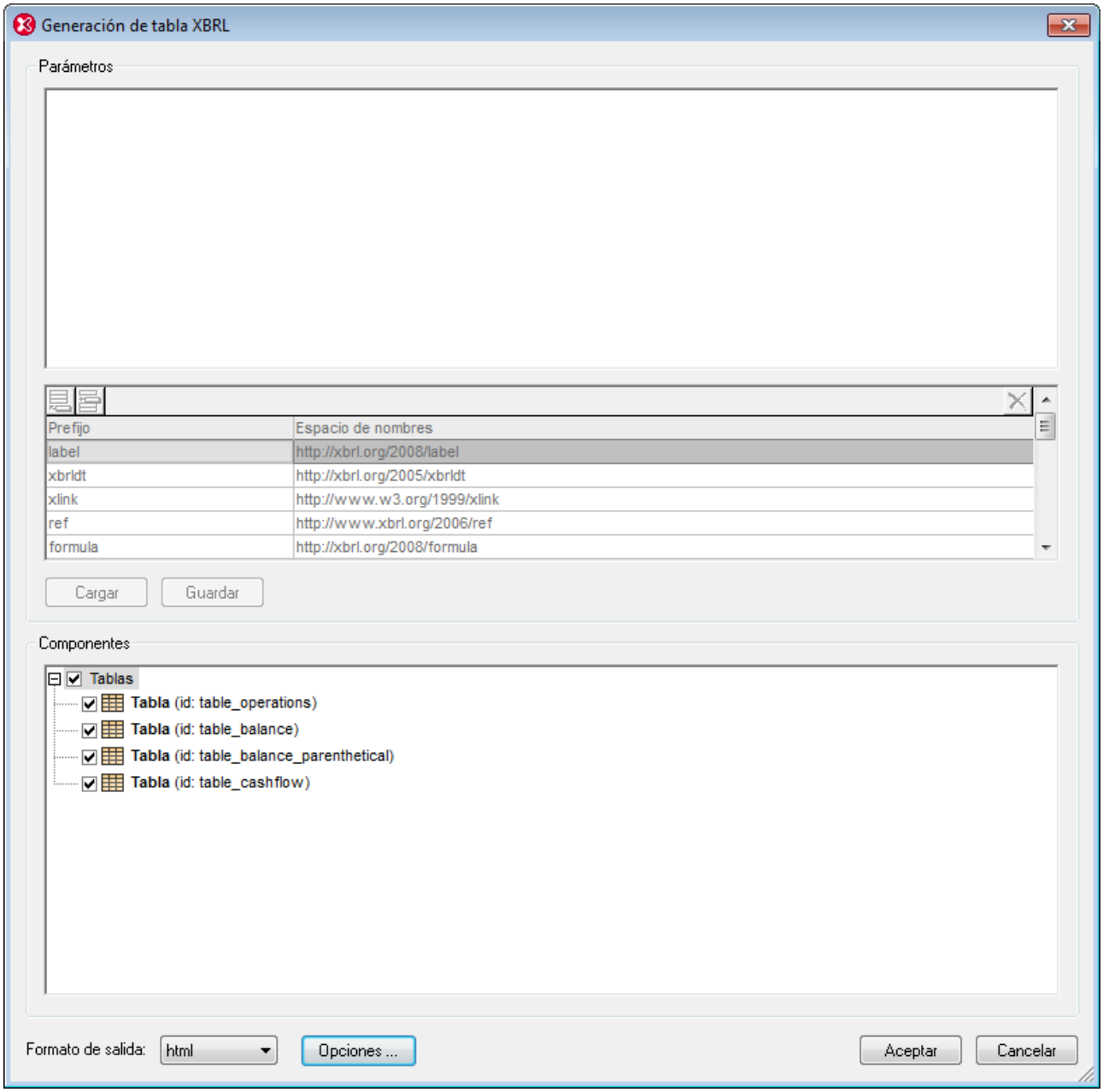
Note: Running a trace can require significant overheads in terms of memory as well as computation speed. When using large XBRL instances, assertions can be evaluated millions of times, and each evaluation might need to store the values of up to 40 variables. So, this feature should only be used for debugging with small/reduced samples, otherwise execution will be slow and XMLSpy might even run out of memory. For this reason, each trace is has a hard-coded limit of 1000 evaluations.

28.16.13 Generar tabla (en el servidor)

Los comandos **Generar tabla** y **Generar tabla en el servidor (alto rendimiento)** se habilitan cuando (i) el documento activo de la vista **Texto** o **Cuadrícula** es un documento de instancia XBRL (`.xbrl`) o cuando (ii) el documento activo de la vista **XBRL**, **Texto**, **Cuadrícula** o **Esquema** es una taxonomía XBRL. Estos comandos generan un documento XML o HTML con las tablas XBRL definidas en la DTS asociada al documento activo (una DTS, *Discoverable Taxonomy Set*, es una colección de taxonomías). En el caso de archivos de instancia XBRL, las tablas se generan a partir de los datos del archivo de instancia XBRL.

El comando **Generar tabla en el servidor (alto rendimiento)** utiliza un servidor RaptorXML+XBRL Server asociado[1068] para generar las tablas. Utilice el comando **Herramientas | Gestionar servidores Raptor**[1559] para configurar un servidor RaptorXML+XBRL Server.

Si no hay tablas en la DTS, aparece un mensaje a tal efecto. Si en la DTS hay una definición de tabla válida, entonces aparece el cuadro de diálogo "Generación de tabla XBRL" (*imagen siguiente*).



Parámetros

Si hay parámetros definidos en la DTS, aparecerán en el panel *Parámetros,* donde puede introducir un valor para cada uno de ellos. Los nombres de parámetro son de solo lectura. Los parámetros obligatorios están marcados con un signo de exclamación rojo y el botón **Aceptar** no se habilita hasta que se asigne un valor al parámetro. Los parámetros opcionales tienen un valor predeterminado. Si se especifica un tipo necesario, aparece. Los parámetros que necesitan varios valores se marcan con el icono **+** (haga clic en este icono para añadir un valor nuevo). Recuerde que los parámetros opcionales que no tengan un valor no se pasarán al motor. Los valores predeterminados son de solo lectura y se ejecutarán si el usuario no introduce un valor distinto.

© 2016-2022 Altova GmbH

Asignaciones de espacios de nombres
En el panel central del cuadro de diálogo aparece una tabla donde puede ver qué prefijos se usan en los QName de los parámetros y tipos. Aquí puede definir los espacios de nombres que se deben usar en la evaluación de parámetros.

Guardar y cargar los parámetros
La configuración de los parámetros, incluidas las asignaciones de espacios de nombres, se puede guardar en formato JSON o XML haciendo clic en el botón **Guardar**. El formato de archivo depende de la extensión dada al archivo. No olvide que los parámetros opcionales que no tengan un valor no se guardan. Una vez guardado, el archivo de parámetros puede cargarse al cuadro de diálogo con el botón **Cargar**.

Opciones de procesamiento XBRL
El botón **Opciones** abre el cuadro de diálogo "Opciones de procesamiento XBRL" 1536, en el que puede activar la deduplicación (ignorar automáticamente los hechos duplicados).

Componentes

Este panel ofrece una vista en forma de árbol donde puede seleccionar los componentes de la tabla que se deben ejecutar. Cada componente va acompañado de un icono, de una descripción y del ID (si está disponible). Marque la casilla del componente que desea ejecutar. El resultado de la ejecución de aserciones puede generarse en formato XML o HTML (seleccione el formato de salida en el cuadro combinado *Formato de salida:*). Haga clic en el botón Opciones para abrir el cuadro de diálogo "Opciones de procesamiento XBRL" 1536, en el que puede configurar la generación de tablas XBRL y determinar si los duplicados deben ignorarse o no. Si hay etiquetas preferidas disponibles, se usan estas; en caso contrario se usan las opciones predeterminadas del cuadro de diálogo "Opciones de procesamiento XBRL" 1536.

Ejecución

Si se produce un error de ejecución, en la ventana Mensajes aparece un mensaje de error a tal efecto. De lo contrario aparece un mensaje confirmando que la tabla se generó correctamente. Los archivos de salida `table-ouput-file.xml/html` se abren en ventanas de documento nuevas, pero no se guardan en disco.

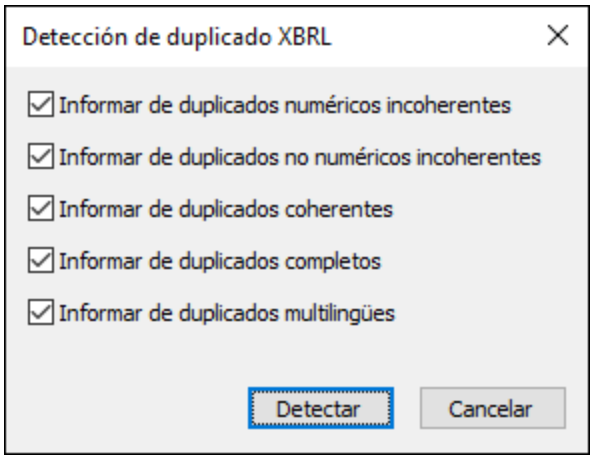
28.16.14 Detectar duplicados (en el servidor)

Los comandos **Detectar duplicados** y **Detectar duplicados en el servidor (alto rendimiento)** se habilitan cuando un documento de instancia XBRL es el documento activo en la vista Texto o en la vista Cuadrícula. Estos comandos comprueban el documento de instancia en busca de hechos duplicados y, de encontrarlos, informan sobre ellos en la ventana Mensajes. El comando **Detectar duplicados en el servidor (alto rendimiento)** usa un servidor RaptorXML+XBRL Server asociado 1068 para buscar duplicados. Use el comando Herramientas **Herramientas | Gestionar servidores Raptor** 1559 para configurar un servidor RaptorXML+XBRL. Al hacer clic en cualquiera de esos comandos se abre el cuadro de diálogo Detectar duplicados XBRL (*imagen siguiente*). En este cuadro de diálogo puede seleccionar qué tipos de duplicados busca.

Los distintos tipos de duplicados se explican en detalle en la especificación Handling Duplicate Facts in XBRL and Inline XBRL 1.0, pero aquí tiene un resumen:

- *Duplicados completos:* son duplicados iguales en nombre, contexto, valor;
- *Duplicados coherentes:* son hechos numéricos con el mismo valor hasta el lugar decimal indicado para el redondeo;
- *Duplicados multilingües:* repiten el mismo hecho en distintos idiomas;

- *Duplicados incoherentes:* son duplicados que cumplen con las condiciones para duplicar hechos descritas en la especificación XBRL 2.1 pero que no son duplicados completos, coherentes o multilingües (por ejemplo, un duplicado numérico con un valor numérico distinto).



28.16.15 Ejecutar XULE

El comando **Ejecutar XULE** ejecuta reglas XULE en un documento de instancia XBRL. Las reglas XULE pueden estar en un único archivo `.xule` o en un archivo zip (`.zip`) que contiene varios documentos XULE. Véase la *sección XBRL | XULE* 921 para más información.

Este comando se habilita en varios casos:

- si el documento activo es un documento XULE. Un documento XULE suele tener la extensión `.xule`. En este caso la aplicación le pedirá que elija la instancia XBRL con respecto a la cual se debe procesar el documento XULE.
- si el documento activo es un documento de instancia XBRL (que suele tener las extensiones `.xbrl` o `.xml`). En este caso la aplicación le pedirá que seleccione el documento XULE o el archivo zip con archivos XULE (conjunto de reglas XULE) que quiere usar.

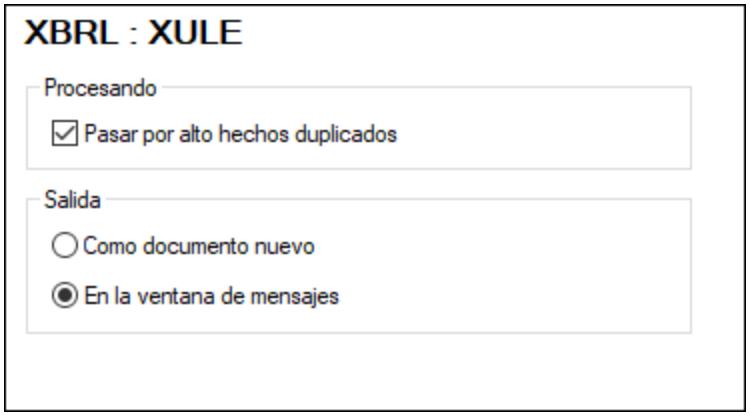
Nota: si tanto el documento XULE como el documento XBRL forman parte de un proyecto de XMLSpy 1060, entonces puede indicar el archivo de instancia XBRL en las propiedades del proyecto de XMLSpy 1328. Si después hace clic con el botón derecho en el archivo XULE y selecciona el comando **Ejecutar XULE**, entonces la ejecución se lleva a cabo con el documento XBRL que se haya definido como destino de la ejecución XULE.

Opciones de ejecución

Resultados XULE

El resultado de la ejecución XULE se envía: (i) a la ventana *Mensajes* o (ii) a un documento nuevo que se muestra en una ventana nueva de XMLSpy y que se almacena temporalmente en la memoria; este documento se puede guardar en un archivo con el comando **Archivo | Guardar como** 1269. Para indicar si el resultado debe ir a una ventana o a un documento nuevo seleccione la opción que prefiera (*imagen siguiente*) en la pestaña de opciones XBRL 1624 *XULE* 1624 (**Herramientas | Opciones | XBRL | XULE**).

© 2016-2022 Altova GmbH



Hechos duplicados
Los hechos duplicados son varias referencias al mismo hecho. Para evitar que los hechos duplicados aparezcan en los resultados marque la casilla *Ignorar hechos duplicados* en la pestaña de opciones XBRL 1624 *XULE* 1624 (**Herramientas | Opciones | XBRL | XULE**). En este caso (*imagen siguiente*) solo se informa una vez de los hechos duplicados.

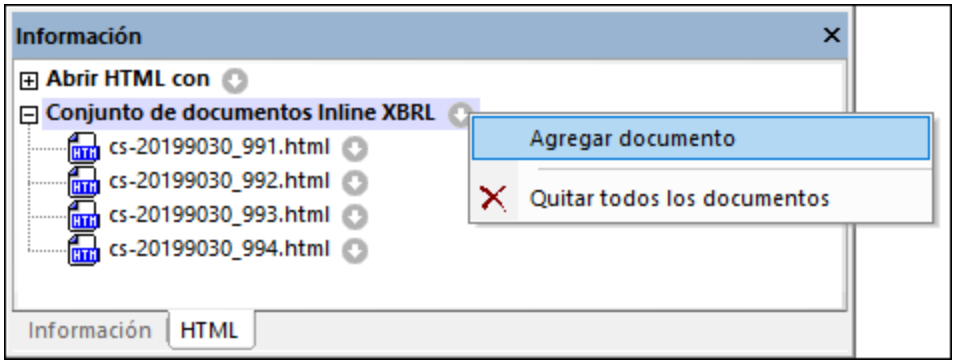
28.16.16 Transformar Inline XBRL

El comando **Transformar Inline XBRL** se habilita cuando el documento activo de la vista Texto, Cuadrícula o Explorador es un documento XHTML que contiene datos Inline XBRL. Este comando extrae los datos Inline XBRL del documento XHTML activo y genera un documento XBRL que contiene los datos extraídos. El documento XBRL que se genera se abre en ventanas nuevas y se puede guardar en disco. Para que este comando funcione correctamente es imprescindible que todos los recursos a los que hace referencia el documento Inline XBRL estén disponibles para poder procesarlos.

Nota: en el cuadro de diálogo "Opciones de procesamiento XBRL" 1536, al que se accede con el comando de menú **XBRL | Opciones de procesamiento** 1536.

Procesar varios documentos Inline XBRL

Para procesar varios documentos Inline XBRL añada los documentos XBRL a *Conjunto de documentos Inline XBRL*, en la pestaña HTML de la ventana Información 124 (*imagen siguiente*). Esta pestaña solo aparece en la ventana Información si hay un documento HTML activo en la ventana principal.



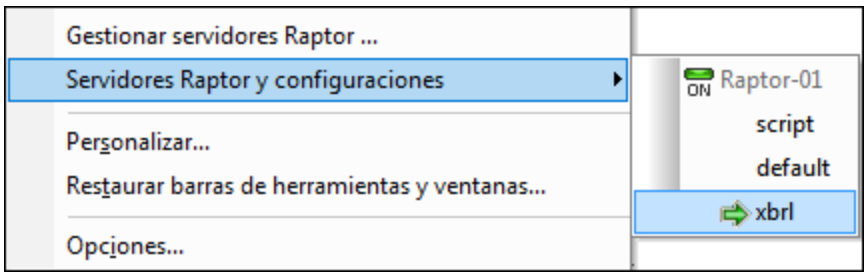
Haga clic en el botón de menú de *Conjunto de documentos Inline XBRL* y después en **Agregar documento** (*imagen anterior*), luego navegue hasta los archivos Inline XBRL que quiera añadir. Cuando ejecute el comando **Transformar Inline XBRL** se procesarán el archivo activo y los archivos del conjunto de documentos. Los datos Inline XBRL extraídos de todos estos archivos se combinan en un único documento XBRL que se abre en una ventana nueva.

28.16.17 Validate EDGAR on Server

El comando **Validar EDGAR en el servidor (alto rendimiento)** valida el documento de instancia XBRL activo usando el RaptorXML+XBRL Server activo en ese momento[1563] y su configuración activa[1563]. Al validar con EDGAR, Raptor valida el documento de instancia XBRL con un script EDGAR interno[1088]. Este comando valida el documento de inmediato y muestra los resultados en la ventana Mensajes.

Nota: el rendimiento real depende del número de núcleos del procesador del PC que use RaptorXML+XBRL Server para la validación: cuantos más núcleos se usen, más rápido será el proceso.

Si definió varias configuraciones en varios servidores, podrá seleccionar un servidor y una de sus configuraciones como configuración activa. La configuración activa se usará para todas las validaciones siguientes. Pase el puntero por encima del comando **Herramientas | Servidores Raptor y configuraciones** (*imagen siguiente*) para abrir el submenú que contiene todos los servidores que se añadieron, así como sus configuraciones. Seleccione la configuración de servidor que desea usar como configuración activa. Por ejemplo, en la imagen siguiente, se seleccionó la configuración `xbrl` del servidor `Raptor-01`. La fecha verde indica que se trata de la configuración activa.



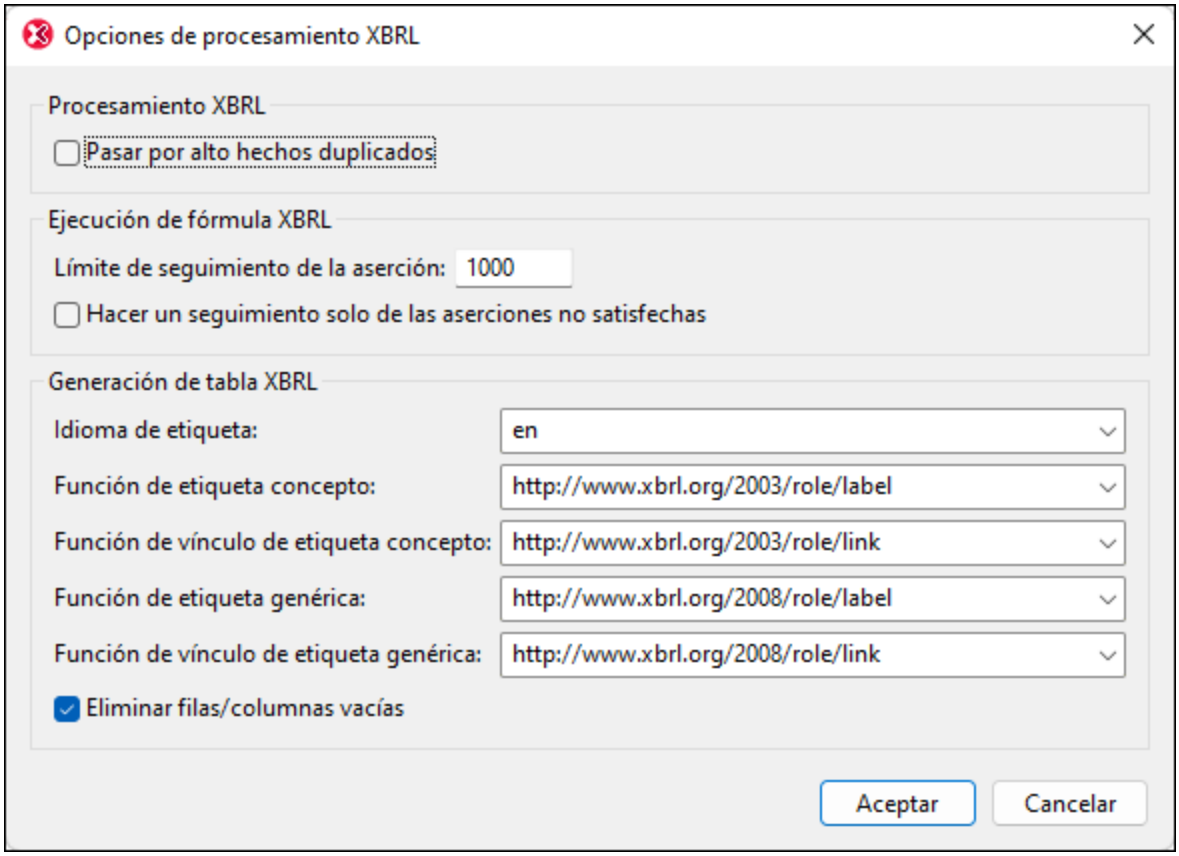
Para más información consulte la sección que describe cómo usar RaptorXML+XBRL Server[1068].

28.16.18 Opciones de procesamiento

El comando **Opciones de procesamiento** XBRL abre el cuadro de diálogo "Opciones de procesamiento XBRL", en el que puede especificar:

- que se ignoren los hechos duplicados para: (i) la ejecución de fórmulas XBRL, (ii) la generación de tablas XBRL, (iii) las transformaciones XBRL inline
- para ejecutar fórmulas XBRL, (i) el límite de seguimiento de las aserciones y (ii) si solo se debe hacer un seguimiento de las aserciones no satisfechas o también de las satisfechas
- las opciones de las etiquetas para la generación de tablas XBRL
- que se eliminen las filas y columnas al generar tablas XBRL

© 2016-2022 Altova GmbH



Nota: también se puede acceder al cuadro de diálogo "Opciones de procesamiento XBRL" desde los cuadros de diálogo "Ejecutar fórmula XBRL" 1528 y "Generar tabla XBRL" 1531.

Nota sobre la deduplicación

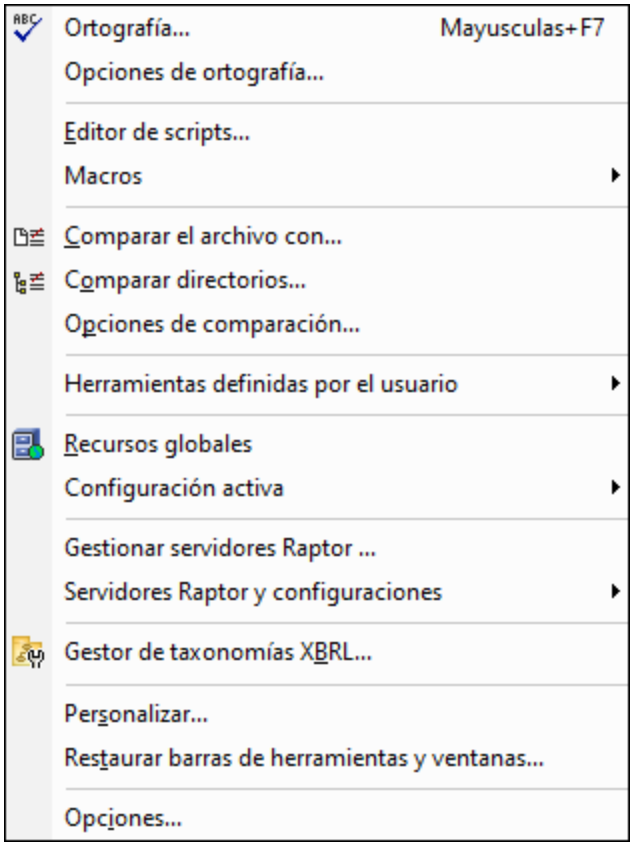
Solo se ignoran los siguientes tipos de hechos duplicados: (i) duplicados completos y (ii) duplicados constantes. El hecho (en los duplicados) que se selecciona (no ignorado) es el más preciso. Por ejemplo, entre duplicados constantes de un hecho numérico se selecciona el hecho con mayos precisión numérica.

Para más información sobre duplicados consulte la especificación Handling Duplicate Facts in XBRL and Inline XBRL 1.0.

28.17 Menú Herramientas

Con el menú **Herramientas** puede:

- Revisar la ortografía[1539] de sus documentos XML
- Acceder al entorno de scripting[1641] de XMLSpy para crear, gestionar y almacenar sus propios formularios, macros y controladores de eventos.
- Ver[1547] las macros asignadas actualmente.
- Comparar dos archivos y ver las diferencias que hay entre ellos.
- Comparar dos carpetas y ver las diferencias que hay entre ellas.
- Acceder a comandos personalizados que usan aplicaciones externas. Estos comandos se pueden crear en la pestaña *Herramientas*[1568] del cuadro de diálogo Personalizar[1568].
- Definir recursos globales[1557]
- Cambiar la configuración activa[1558] para los recursos globales de XMLSpy
- Agregar servidores RaptorXML[1560] para validar datos XML y XBRL y configurar las opciones de validación de RaptorXML[1561]
- Seleccionar una configuración de RaptorXML Server[1563] como configuración activa
- Administrar sus paquetes de taxonomías XBRL con la aplicación desarrollada para ello, el Gestor de taxonomías[1563]
- Personalizar[1564] su versión de XMLSpy: definiendo sus propias barras de herramientas, teclas de acceso rápido, macros y menús.
- Definir opciones de configuración[1583] globales de XMLSpy



 © 2016-2022 Altova GmbH

28.17.1 Ortografía

El corrector ortográfico de XMLSpy (y sus diccionarios integrados) está disponible en la vista Texto, en la vista Cuadrícula y la vista Authentic. Si está editando un documento en otra vista y desea revisar su ortografía, entonces debe cambiar a la vista Texto o Cuadrícula. Por ejemplo, si está editando un esquema XML en la vista Esquema, cambie a la vista Texto o Cuadrícula y revise la ortografía del documento.

Nota: los diccionarios integrados que vienen con el software de Altova no son un reflejo de las preferencias lingüísticas de Altova. Los diccionarios integrados del corrector ortográfico se seleccionaron porque su licencia permite su distribución con software comercial (MPL, LGPL o BSD, por ejemplo). Existen muchos otros diccionarios de código abierto, pero se distribuyen con licencias más restrictivas, como la licencia GPL. Estos diccionarios pueden descargarse e instalarse por separado desde la página http://www.altova.com/es/dictionaries. Elija los diccionarios que quiere usar en base a las licencias de estos y a sus propias necesidades.

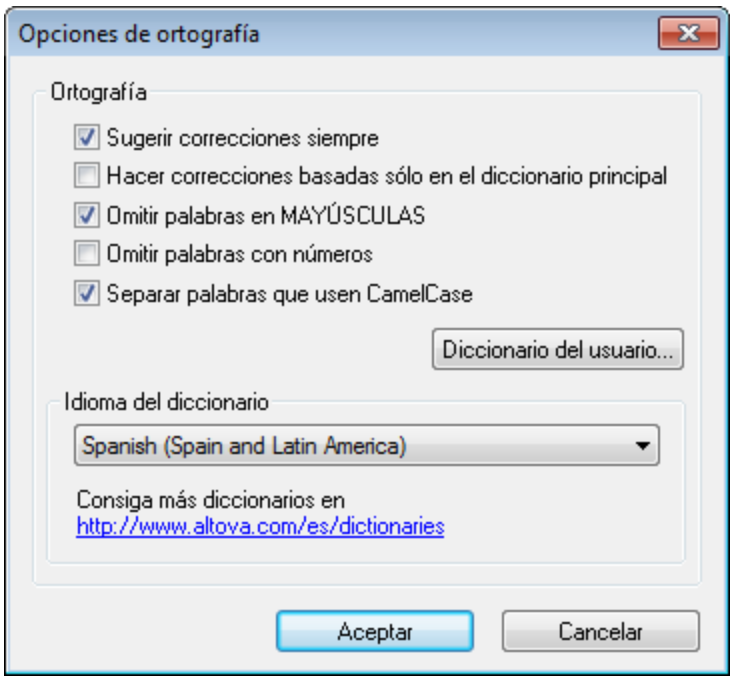
Este apartado describe el uso del corrector ortográfico y tiene tres partes:

- Seleccionar el idioma del corrector ortográfico 1539
- Definir el contexto de la revisión ortográfica 1540
- Ejecutar la revisión ortográfica 1540

Seleccionar el idioma del corrector ortográfico

Siga estos pasos para seleccionar el idioma del corrector ortográfico:

1. Haga clic en el comando de menú **Herramientas | Opciones de ortografía**.
2. En el cuadro de diálogo "Opciones de ortografía" que aparece seleccione uno de los diccionarios instalados de la lista desplegable del cuadro combinado *Idioma del diccionario*.

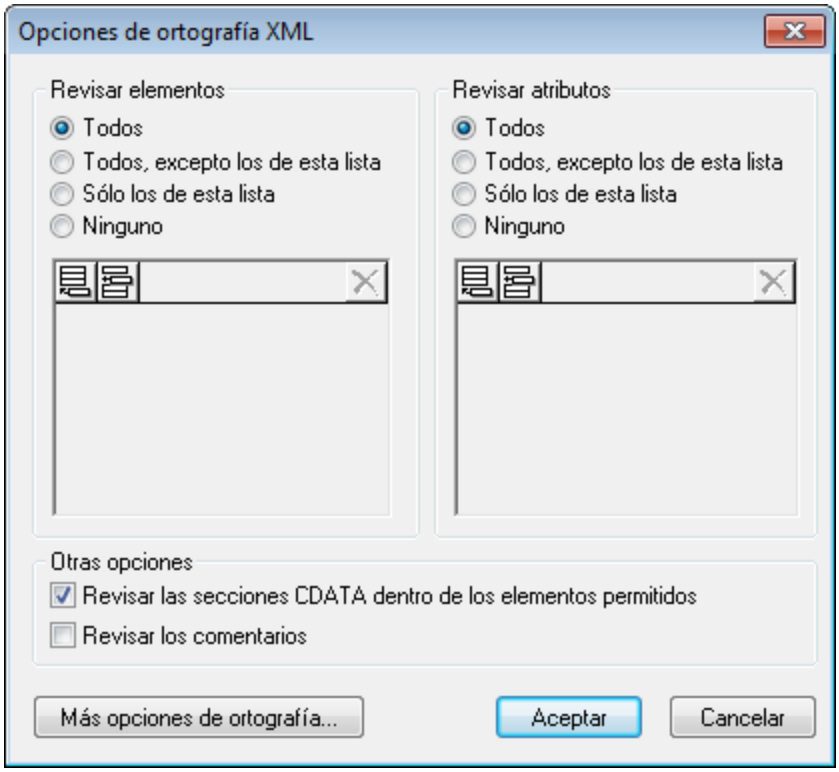


3. Para terminar haga clic en **Aceptar**.

El idioma del diccionario seleccionado será utilizado durante la revisión ortográfica. Si el idioma que desea utilizar no está instalado, puede descargar más diccionarios desde el sitio web de Altova. Para más información consulte el apartado Opciones de ortografía (añadir diccionarios para el corrector ortográfico)[1545].

Definir el contexto de la revisión ortográfica

Cuando el corrector ortográfico se utiliza en la vista Texto o Cuadrícula, puede definir el contexto de la revisión antes de ejecutar el corrector. Para ello seleccione el comando **Herramientas | Opciones de ortografía** y defina el contexto que desea utilizar en el cuadro de diálogo Opciones de ortografía XML[1543] que aparece (*imagen siguiente*).

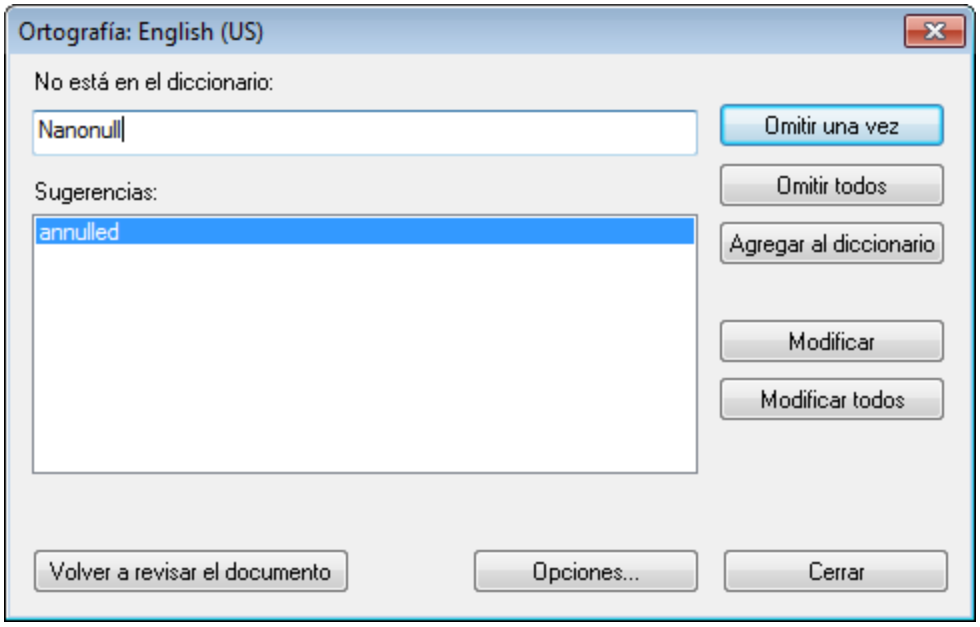


En este cuadro de diálogo puede seleccionar qué elementos y atributos se revisan y si las secciones CDATA y los comentarios se deben revisar o no. Consulte la descripción del comando **Opciones de ortografía**[1543] para más información.

Ejecutar la revisión ortográfica

El comando **Herramientas | Ortografía (Mayús+F7)** empieza a revisar el documento XML activo inmediatamente, según el contexto definido en el paso anterior[1540]. Si el corrector encuentra una palabra desconocida, aparece el cuadro de diálogo *Ortografía: No está en el diccionario* (*imagen siguiente*). De lo contrario, el corrector recorre el documento hasta el final.

© 2016-2022 Altova GmbH



A continuación describimos el cuadro de diálogo *Ortografía: No está en el diccionario* y las opciones disponibles en él.

No está en el diccionario
Este cuadro de texto contiene la palabra desconocida (es decir, una palabra que no está en el diccionario seleccionado ni en el diccionario del usuario). En caso de encontrar una palabra desconocida hay varias opciones:

- Puede editar la palabra en el cuadro de texto manualmente o seleccionar una de las opciones del panel *Sugerencias.* Después haga clic en el botón **Modificar** para reemplazar la palabra en el documento XML con la palabra nueva. (Si hace doble clic en una sugerencia, la palabra sugerida se inserta directamente en el documento XML.) La palabra que aparece en el cuadro *No está en el diccionario* también aparece resaltada en el documento XML para que pueda editarla directamente en el documento si lo prefiere. Si hace clic en el botón **Modificar todos** se reemplazan todas las instancias de la palabra en el documento con la palabra nueva.
- Puede decidir no realizar ningún cambio y omitir la advertencia del corrector (para la instancia actual de la palabra o para todas las instancias de la palabra).
- Puede añadir la palabra al diccionario del usuario para que a partir de ahora la palabra sea correcta.

Sugerencias
Este cuadro sugiere palabras que se parecen a la palabra desconocida (las palabras sugeridas provienen del diccionario integrado y de los diccionarios del usuario). Si hace doble clic en una palabra de la lista, la palabra elegida se inserta automáticamente en el documento y el proceso de revisión ortográfica continúa.

Omitir una vez
Este botón permite seguir revisando el documento e ignorar la primera aparición de la palabra desconocida. Si la misma palabra aparece otra vez en el documento, volverá a aparecer en el recuadro *No está en el diccionario.*

Omitir todos
Este botón permite seguir revisando el documento e ignorar todas las apariciones de la palabra desconocida en todo el documento.

Agregar al diccionario
Este botón añade la palabra desconocida al diccionario del usuario. Para tener acceso al diccionario del usuario (para poder editarlo) utilice el cuadro de diálogo Opciones de ortografía[1546].

Modificar
Este botón reemplaza la palabra desconocida en el documento XML con la palabra (nueva) del cuadro *No está en el diccionario*.

Modificar todos
Este botón reemplaza todas las instancias de la palabra desconocida en el documento XML con la palabra (nueva) del cuadro *No está en el diccionario*.

Volver a revisar el documento
Este botón vuelve a revisar el documento desde el principio.

Opciones
Dependiendo de la vista en la que esté trabajando, este botón abre:

- el cuadro de diálogo Opciones de ortografía[1543] (si está trabajando en la vista Authentic) o
- el cuadro de diálogo Opciones de ortografía XML[1542] (si está trabajando en la vista Texto o Cuadrícula.

Para más información sobre estos cuadros de diálogo consulte el apartado Opciones de ortografía[1542].

Cerrar
Este botón cierra el cuadro de diálogo Ortografía.

28.17.2 Opciones de ortografía

El comando **Herramientas | Opciones de ortografía** abre las opciones de configuración del corrector ortográfico[1543].Dependiendo de la vista activa, el comando **Herramientas | Opciones de ortografía** abre el cuadro de diálogo **Opciones de ortografía**[1543] directamente (en las vistas Esquema, WSDL, XBRL, Authentic y Explorador) o el cuadro de diálogo **Opciones de ortografía XML**[1542] (en las vistas Texto y Cuadrícula). El cuadro de diálogo "Opciones de ortografía XML" incluye el botón **Más opciones de ortografía**, con el que puede abrir el cuadro de diálogo "Opciones de ortografía".

Las opciones disponibles en estos dos cuadros de diálogo se describen en detalle a continuación:

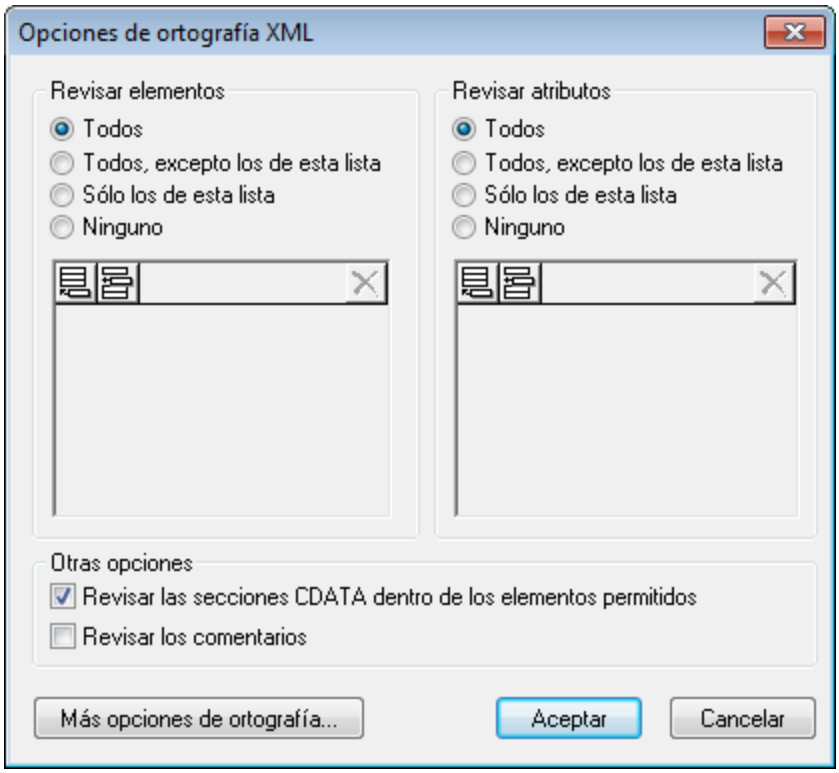
- Contexto de la revisión ortográfica[1542]
- Opciones de ortografía[1543]
- Agregar diccionarios para el corrector ortográfico[1545]
- Trabajar con el diccionario del usuario[1546]

Opciones de ortografía XML (cuadro de diálogo)

Si hace clic en el comando **Herramientas | Opciones de ortografía** en la vista Texto o Cuadrícula, aparece el cuadro de diálogo "Opciones de ortografía XML" (*imagen siguiente*), donde puede seleccionar el contexto de

 © 2016-2022 Altova GmbH

la revisión ortográfica. Se trata de seleccionar qué elementos y atributos se revisan y elegir si las secciones CDATA y los comentarios se revisan también.

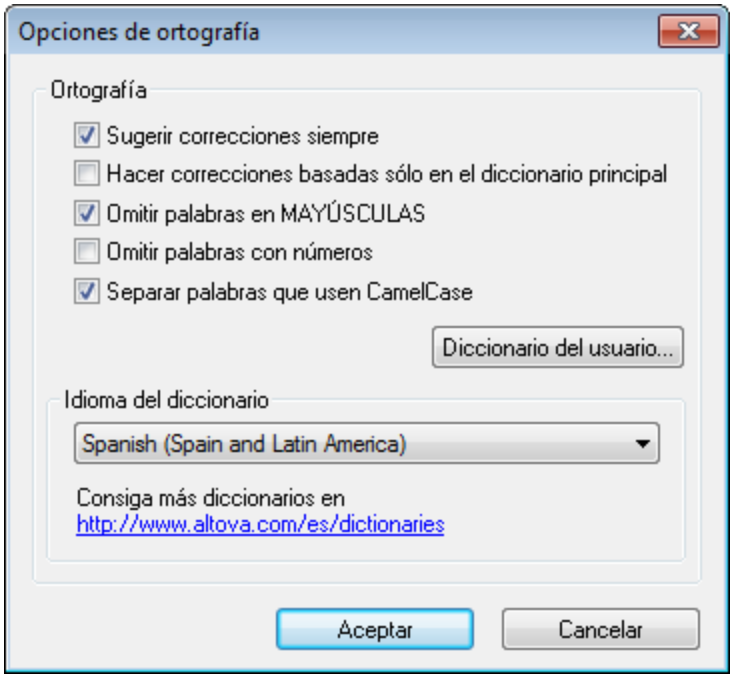


En este cuadro de diálogo puede crear una lista con los elementos y atributos que quiere incluir o excluir de la revisión ortográfica. El cuadro de diálogo también tiene una opción rápida para revisar todos los elementos/atributos o ninguno.

Si hace clic en el botón **Más opciones de ortografía**, aparece el cuadro de diálogo "Opciones de ortografía".

Opciones de ortografía

El cuadro de diálogo "Opciones de ortografía" (*imagen siguiente*) sirve para configurar el corrector ortográfico de forma general.



Sugerir correcciones siempre
Si marca esta casilla, el cuadro *Sugerencias* siempre mostrará opciones del diccionario integrado elegido y del diccionario del usuario. Si desactiva esta opción, el corrector no ofrecerá ninguna sugerencia.

Hacer correcciones basadas sólo en el diccionario principal
Si marca esta casilla, solamente se usan sugerencias del diccionario integrado elegido (diccionario principal). El diccionario del usuario no se utilizará para ofrecer sugerencias. Si marca esta casilla, se deshabilita el botón **Diccionario del usuario**. Es decir, mientras esté activa esta opción el diccionario del usuario no se podrá editar.

Omitir palabras en MAYÚSCULAS
Si marca esta casilla, el corrector ortográfico pasa por alto las palabras que están en mayúsculas.

Omitir palabras con números
Si marca esta casilla, el corrector ortográfico pasa por alto las palabras que contienen números.

Separar palabras que usen CamelCase
Las palabras que usan CamelCase están formadas por palabras que empiezan por mayúsculas, unidas unas a otras sin espacios. Por ejemplo, la palabra *CamelCase* está formada por las palabras *Camel* y *Case*, ambas en mayúsculas y unidas sin espacios. Este tipo de palabras no suele aparecer en los diccionarios y, por tanto, el corrector ortográfico las marcará como erróneas. Marque la casilla *Separar palabras que usen CamelCase* para que el corrector revise por separado cada una de las palabras que forman la palabra CamelCase. Esta opción está marcada por defecto.

Idioma del diccionario
Use este cuadro combinado para seleccionar el idioma del diccionario integrado del corrector ortográfico. La opción predeterminada es **US English** (inglés EE UU). Para descargar gratis más diccionarios en otros idiomas, visite el sitio web de Altova.

© 2016-2022 Altova GmbH

Agregar diccionarios para el corrector ortográfico

Por cada idioma hay dos archivos de diccionario Hunspell que funcionan conjuntamente: un archivo `.aff` y un archivo `.dic`. Los diccionarios se instalan en la carpeta `Lexicons` de este directorio: `C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons`.

Dentro de la carpeta `Lexicons` se crea una carpeta por idioma: `<nombre del idioma>\<archivos del diccionario>`. Por ejemplo, los archivos de los dos diccionarios de español (de España y de Hispanoamérica y España) se almacenan así:

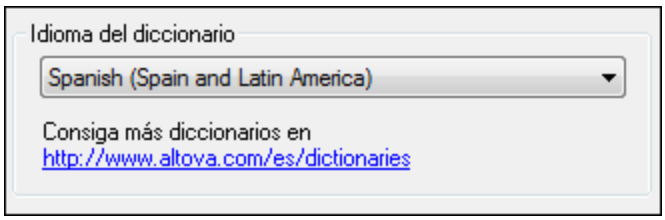
```
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)
\en_GB.aff
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)
\en_GB.dic
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US)\en_US.aff
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US)\en_US.dic
```

La lista desplegable *Idioma del diccionario* del cuadro de diálogo "Opciones de ortografía" muestra los diccionarios disponibles. Se trata de los diccionarios de la carpeta `Lexicons` y reciben el nombre de las subcarpetas de la carpeta `Lexicons`. Por ejemplo, los dos diccionarios de inglés del ejemplo anterior aparecerían así en el cuadro combinado: *Spanish (Spain)* y *Spanish (Spain and Latin America)*.

Todos los diccionarios instalados son compartidos por los diferentes usuarios del equipo y por las diferentes versiones de los productos de Altova (tanto en 64 como en 32 bits).

Hay dos maneras de agregar diccionarios nuevos para el corrector ortográfico. En ninguno de los dos casos es necesario registrar los archivos en el sistema:

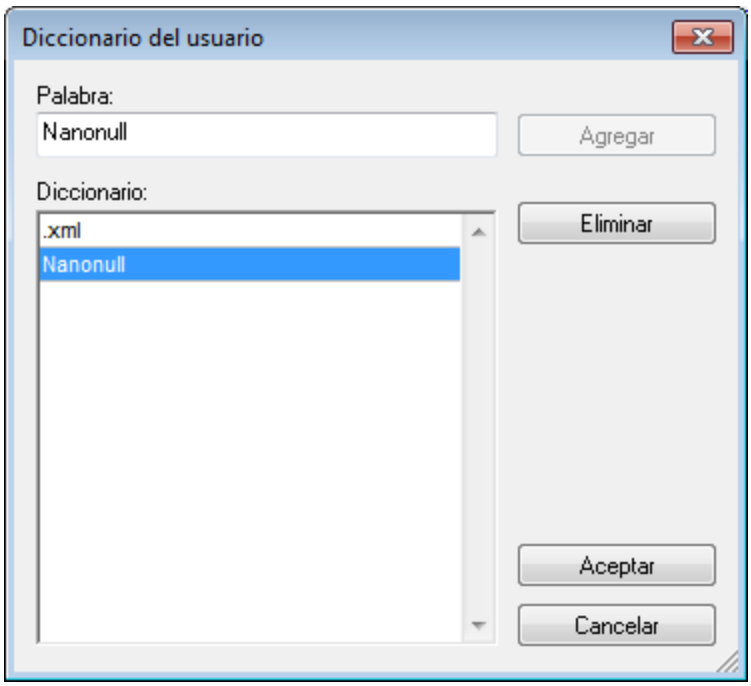
- Puede añadir diccionarios Hunspell a una subcarpeta nueva de la carpeta `Lexicons`. Los diccionarios Hunspell se pueden descargar desde https://wiki.services.openoffice.org/wiki/Dictionaries o desde https://extensions.services.openoffice.org/en/dictionaries, por ejemplo. (Recuerde que OpenOffice utiliza el formato comprimido `OXT`. Cambie la extensión a `.zip` y descomprima los archivos `.aff` y `.dic` en las subcarpetas correspondientes de la carpeta `Lexicons`. También puede usar diccionarios Myspell, ya que los diccionarios Hunspell están basados en Myspell.)
- Puede usar el instalador de diccionarios de Altova, que instala un paquete con varios diccionarios en el directorio adecuado del equipo. En el cuadro de diálogo "Opciones de ortografía", bajo el panel *Idioma del diccionario*, aparece un enlace a la página de Altova de descarga de diccionarios (*imagen siguiente*). Si no usa derechos de administrador para instalar los diccionarios, se producirá un error de instalación.



Nota: recuerde que es decisión suya aceptar o no las condiciones de uso de la licencia del diccionario elegido. También es responsabilidad suya comprobar si el diccionario puede utilizarse en su equipo o no.

Trabajar con el diccionario del usuario

Cada usuario tiene un diccionario del usuario propio, donde se almacenan las palabras aprobadas por el usuario. Durante la revisión ortográfica, el corrector compara la ortografía con una lista compuesta por palabras del diccionario integrado y del diccionario del usuario. Puede añadir o eliminar palabras del diccionario del usuario en el cuadro de diálogo "Diccionario del usuario" (*imagen siguiente*) Para abrir este cuadro de diálogo pulse el botón **Diccionario del usuario** del cuadro de diálogo "Opciones de ortografía" (*segunda imagen de este apartado*).



Para añadir una palabra al diccionario del usuario escriba la palabra en el recuadro *Palabra* y pulse el botón **Agregar**. La palabra se añade a la lista alfabética del panel *Diccionario.* Para eliminar una palabra del diccionario, seleccione la palabra del panel *Diccionario* y pulse el botón **Eliminar**. La palabra se elimina del panel *Diccionario*. Cuando termine de editar el cuadro de diálogo "Diccionario del usuario" haga clic en el botón **Aceptar**. Los cambios se guardan en el diccionario del usuario.

También puede añadir palabras al diccionario del usuario durante la revisión ortográfica. Si el corrector encuentra una palabra desconocida, aparece el cuadro de diálogo Ortografía[1539]. Pulse el botón **Agregar al diccionario** para añadir la palabra desconocida al diccionario del usuario.

El diccionario del usuario se encuentra en este directorio: `C:\Users\<usuario>\Documentos\Altova\SpellChecker\Lexicons\user.dic`

28.17.3 Editor de script

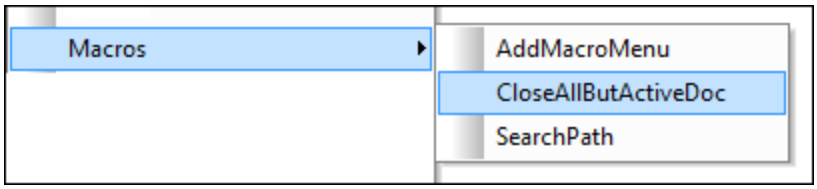
El comando **Editor de script** abre la ventana del editor de script. Para más información sobre cómo trabajar con el editor de script, consulte la sección Editor de scripts[1641] de la presente documentación.

© 2016-2022 Altova GmbH

Nota: es necesario tener instalado .NET Framework versión 2.0 (o superior) para poder ejecutar el editor de scripts.

28.17.4 Macros

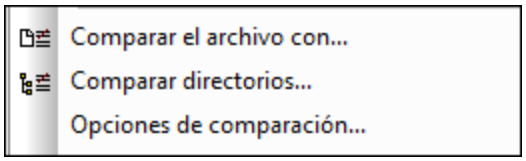
Al pasar el ratón por el comando **Macros** aparece un submenú con todas las macros definidas en el proyecto de script que está activo en XMLSpy (*imagen siguiente*).



Al hacer clic en una macro del submenú (*imagen anterior*) la macro se ejecuta.

28.17.5 Comparaciones

XMLSpy ofrece una función de comparación (o diferenciación) que sirve para comparar archivos XML y archivos de texto, así como carpetas, y ver las diferencias que existen entre ellos.



El menú **Herramientas** ofrece tres comandos con los que puede realizar tareas de comparación de archivos y carpetas:

- **Comparar el archivo con** 1547
- **Comparar directorios** 1551
- **Opciones de comparación** 1555

En los siguientes subapartados describimos estos comandos detalladamente.

28.17.5.1 Comparar el archivo con

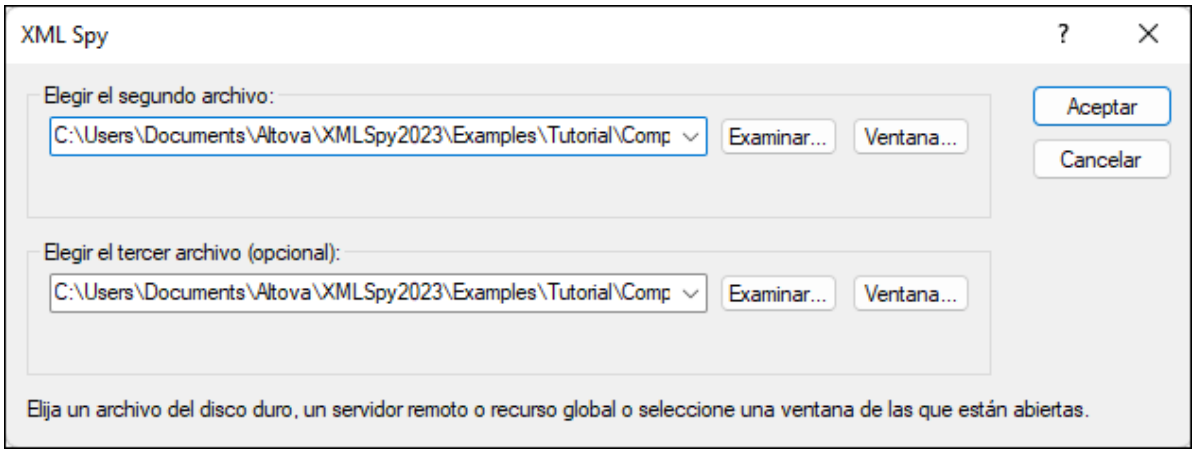
Este comando sirve para comparar el archivo que está abierto con otro archivo. La comparación presenta los archivos que se deben comparar en paralelo en la ventana principal y las diferencias detectadas aparecen resaltadas en ambos archivos. Si hay una diferencia entre dos archivos, se resalta en color verde. Si el contenido es distinto en tres archivos (***nota:** este tipo de comparación solo está disponible en la edición Enterprise Edition*), la diferencia se señala como conflicto y se resalta en color rosa. Los archivos se pueden

comparar como documentos XML (para tener en cuenta la estructura y la semántica de las etiquetas) o como documentos de texto.

Nota: las comparaciones de archivos a tres bandas solamente están disponibles en la edición **Enterprise Edition**.

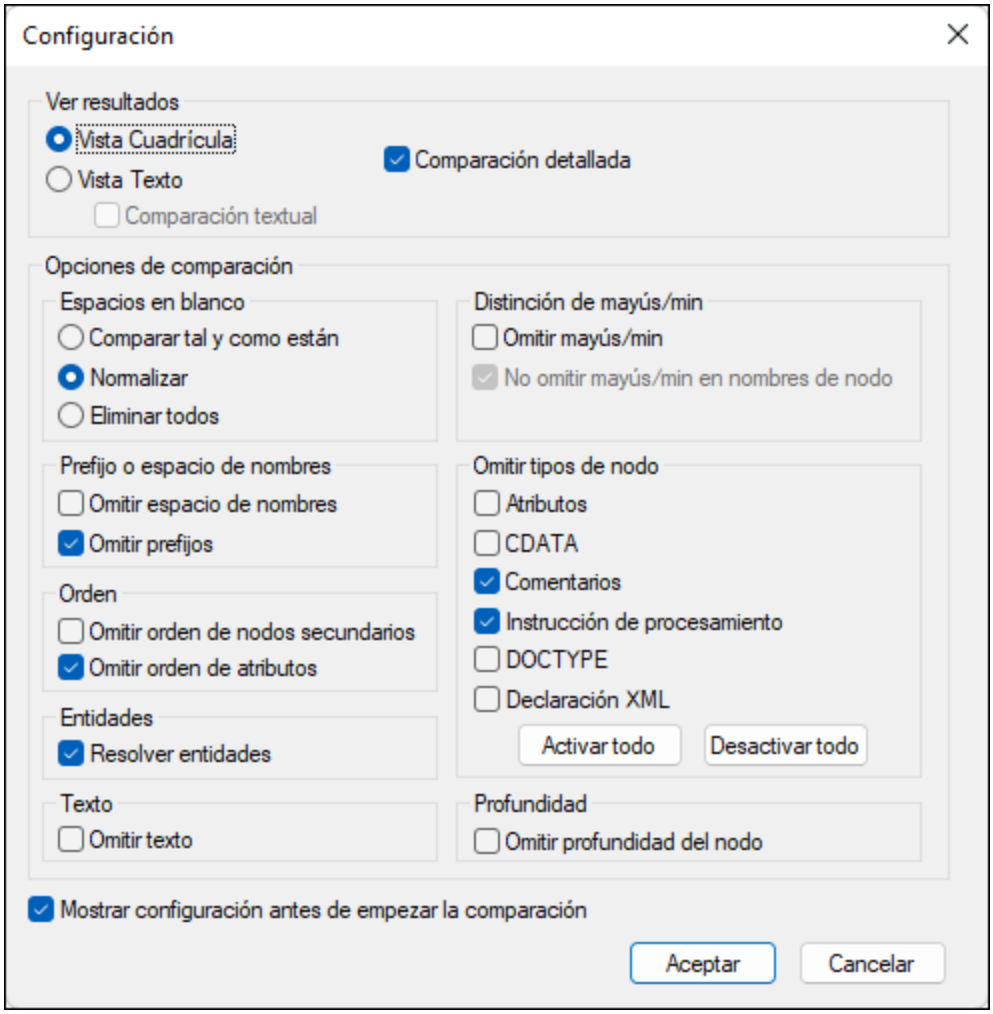
Para comparar el archivo activo con otro archivo:

1. Asegúrese de que el archivo que está activo en la ventana principal es el que desea comparar. Después haga clic en **Herramientas | Comparar el archivo con**. Aparece un cuadro de diálogo (*imagen siguiente*) donde puede navegar hasta el archivo o hasta los dos archivos que se deben comparar con el archivo activo.



2. Haga clic en el botón **Examinar...**[1262] para seleccionar un archivo en el Explorador de Windows, un recurso global o una URL. Haga clic en **Ventana...** para seleccionar un archivo que esté abierto en una de las ventanas de XMLSpy.
3. Haga clic en **Aceptar**. Ahora aparece el cuadro de diálogo "Configuración" (*imagen siguiente*). Estas opciones se describen en el apartado Opciones de comparación[1555]. Si prefiere que este cuadro de diálogo no aparezca cada vez que se inicia una sesión de comparación, desactive la casilla *Mostrar configuración antes de empezar la comparación* (al final del cuadro de diálogo) y haga clic en **Aceptar**.

© 2016-2022 Altova GmbH



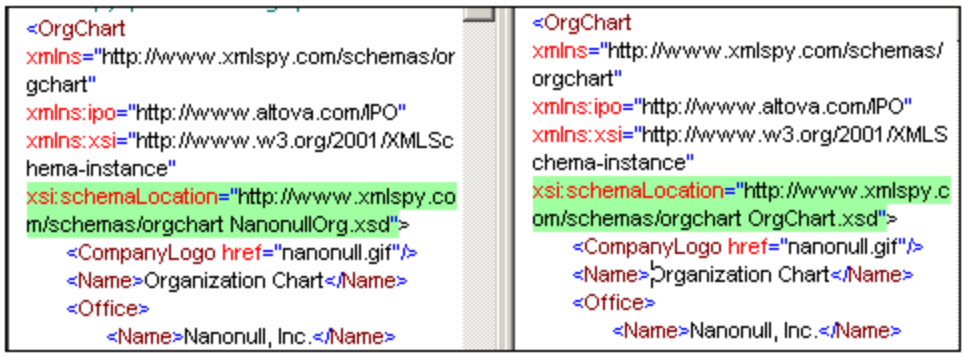
4. Seleccione las opciones correspondientes y haga clic en **Aceptar**. Ahora ocurren dos cosas: (i) en primer lugar, los archivos que van a compararse se presentan en paralelo en paneles diferentes y (ii) en segundo lugar, aparece la ventana de control "Comparar archivos" (*ver imágenes más abajo*[1549]).
5. Con los botones de la ventana de control "Comparar archivos" puede navegar por las diferencias y combinarlas (*encontrará más información más abajo*[1549]).
6. Cuando termine de revisar y combinar las diferencias haga clic en **Listo** en la ventana de control "Comparar archivos".

Nota: las opciones de comparación se pueden modificar durante la sesión de comparación (haciendo clic en **Herramientas | Opciones de comparación**[1555]), pero los cambios solo surtirán efecto a partir de la siguiente sesión de comparación (es decir, los cambios no afectarán a la sesión de comparación actual).

Comparaciones de archivos y combinación de diferencias

Hay dos tipos de comparación:

- *Comparación bidireccional:* se comparan dos archivos. El archivo que estaba activo cuando se inició la comparación aparece en el panel izquierdo y se compara con otro archivo (*imagen siguiente*).



- *Comparación a tres bandas (solo en la edición Enterprise Edition):* se comparan tres archivos. El archivo activo aparece en el panel izquierdo. El primero de los dos archivos seleccionados para compararse con el archivo activo aparece en el panel intermedio. Este archivo intermedio se denomina *archivo base*. Puede combinar diferencias entre archivo izquierdo y el archivo base o entre el archivo derecho y el archivo base.

Nota: evite mover los paneles de la ventana o de lo contrario el sentido de los botones **Copiar-> / <-Copiar** de la ventana de control "Comparar archivos" dejará de ser el correcto.

Diferencias y conflictos

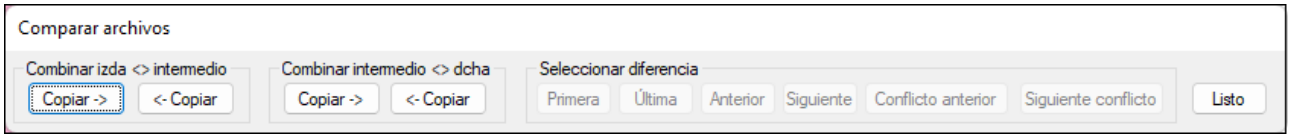
Las diferencias y los conflictos se resaltan en colores distintos y pueden examinarse por separado (*ver más abajo* 1551).

- *Diferencias:* se trata del contenido que es diferente en dos archivos. Se resalta en color verde.
- *Conflictos:* se trata del contenido que es diferente en tres archivos. Se resalta en color rosa.

Nota: en las comparaciones bidireccionales solo se muestran diferencias, no conflictos.

Navegar por las diferencias y conflictos

Puede navegar por el documento con ayuda de los botones del panel *Seleccionar diferencia* de la ventana de control "Comparar archivos" (*imágenes siguientes*). En las comparaciones bidireccionales este panel cuenta con cuatro botones, mientras que en las comparaciones a tres bandas cuenta con seis botones (*imágenes siguientes*).



La ventana de control "Comparar archivos" cuenta con estos botones:

- *Primera:* este botón conduce a la primera diferencia o al primero conflicto del documento.
- *Última:* este botón conduce a la última diferencia o al último conflicto del documento.
- *Anterior:* este botón conduce a la diferencia anterior a la diferencia que está activa o a la diferencia anterior al conflicto que está activo.
- *Siguiente:* este botón conduce a la diferencia siguiente a la diferencia que está activa o a la diferencia siguiente al conflicto que está activo.

 © 2016-2022 Altova GmbH

- *Conflicto anterior:* este botón conduce al conflicto anterior a la diferencia o conflicto que está activo.
- *Siguiente conflicto:* este botón conduce al conflicto siguiente a la diferencia o conflicto que está activo.

Combinar diferencias y conflictos
Puede usar los botones del panel *Combinar* para copiar el contenido resaltado en un panel en el otro panel. Use el botón **Copiar-> / <-Copiar** que corresponda para copiar el contenido resaltado de un panel a otro. Para habilitar la combinación de diferencias es necesario haber definido estas opciones de comparación[1555]:

- la casilla *Comparación detallada* debe estar marcada y
- la casilla *Omitir profundidad del nodo* debe estar desactivada.

Cuando termine de revisar y combinar las diferencias haga clic en el botón **Listo**.

Nota: si quiere deshacer una combinación, detenga la sesión de comparación, seleccione el archivo donde desea deshacer el cambio y seleccione **Edición | Deshacer** o pulse **Ctrl+Z**.

Nota: mientras está activa la sesión de comparación no se permite la edición de datos ni alternar entre vistas. Si intenta editar datos o cambiar de vista en cualquiera de los archivos, aparecerá un mensaje advirtiéndole de que la sesión de comparación se cerrará.

Colores utilizados para señalar diferencias y conflictos

Si el contenido es distinto en dos archivos, las líneas se resaltan en color verde en ambos archivos (pues se trata de una diferencia). Si el contenido es distinto en los tres archivos, las líneas se resaltan en color rosa en los tres archivos (pues se trata de un conflicto). Si selecciona una diferencia, la línea aparece señalada en verde oscuro en ambos archivos. Si selecciona un conflicto, la línea aparece señalada en rosa oscuro en los tres archivos.

Estos son los colores que se usan en las sesiones de comparación para señalar diferencias y conflictos:

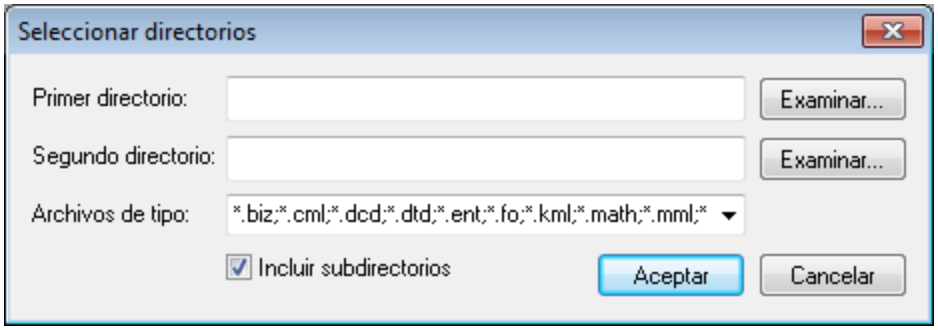
	Diferencia
	Diferencia actual
	Diferencia combinada
	Diferencia combinada actual
	Conflicto
	Conflicto actual

28.17.5.2 Comparar directorios

El comando **Comparar directorio** sirve para comparar dos directorios, con o sin sus respectivos subdirectorios. Los directorios se comparan para ver si faltan archivos en algún directorio y si los archivos del mismo nombre son diferentes o no.

Siga estos pasos para comparar dos directorios:

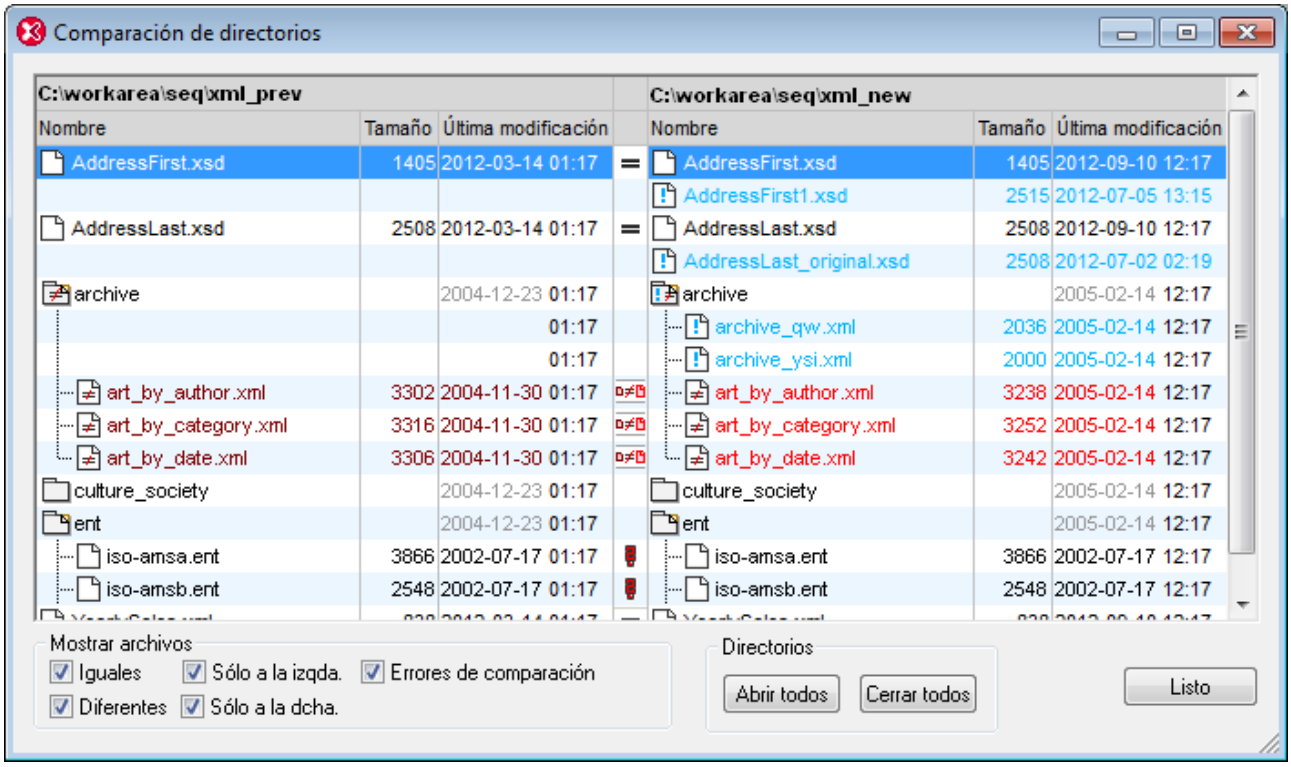
1. Seleccione el comando **Herramientas | Comparar directorios**. Aparece este cuadro de diálogo:



2. Busque los directorios que desea comparar y marque la casilla *Incluir subdirectorios* si desea incluir los subdirectorios en la sesión de comparación.
3. En el campo *Archivos de tipo*: debe seleccionar los tipos de archivo que desea comparar. El menú desplegable contiene tres opciones: (i) todos los archivos de tipo XML; (ii) los tipos de archivo definidos en XMLSpy 1587; y (iii) todos los tipos de archivo. Los archivos ZIP se comparan como si fueran directorios y los directorios y ficheros del archivo ZIP aparecen en la ventana de resultados. Para estar seguro de que los archivos ZIP se leen como directorios, compruebe que la extensión de archivo pertinente aparece en el campo *Archivos de tipo:* antes de iniciar la comparación.
4. Haga clic en **Aceptar**. Aparece el cuadro de diálogo "Configuración" (consulte el subapartado Opciones de comparación 1555).
5. Seleccione las opciones de comparación.
6. Haga clic en **Aceptar**. Aparece un cuadro de diálogo que indica el progreso de la comparación.

El resultado de la comparación aparece en una ventana como esta:

© 2016-2022 Altova GmbH



Símbolos para los directorios en la ventana de resultados

Los nombres de los directorios aparecen en negro.

El directorio está contraído y no se ve su contenido.

El directorio está expandido (la esquina de la carpeta está doblada) y se puede ver su contenido.

El directorio contiene archivos que no se pueden comparar o que no son diferentes de los archivos correspondientes del otro directorio.

El directorio contiene archivos que no existen en el otro directorio.

El directorio contiene archivos que son diferentes de los archivos correspondientes del otro directorio.

El directorio contiene archivos que no existen en el otro directorio y archivos que son diferentes de los archivos correspondientes del otro directorio.

Símbolos para los archivos en la ventana de resultados

El color del nombre de los archivos depende de su estado de comparación:



Este archivo no se puede comparar (con el archivo correspondiente del otro directorio). En la columna central aparece un signo de interrogación. El nombre del archivo aparece en negro.



Este archivo es igual que el archivo correspondiente del otro directorio. En la columna central aparece un signo de igual. El nombre del archivo aparece en negro.



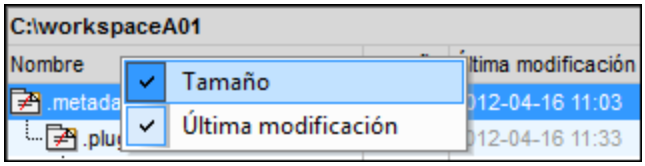
Este archivo no existe en el otro directorio. La columna central está vacía. El nombre del archivo aparece en azul.



Este archivo es diferente del archivo correspondiente del otro directorio y la última modificación del archivo es más reciente que la última modificación del archivo correspondiente. El archivo más reciente aparece en un rojo intenso. En la columna central aparece un icono con dos archivos y el más reciente de ellos aparece en un rojo intenso.

Opciones de presentación

- Elija qué archivos aparecen en la ventana de resultados activando o desactivando las opciones del panel *Mostrar archivos* situado en la parte inferior de la ventana.
- Abra o cierre todos los subdirectorios pulsando el botón correspondiente del panel *Directorios*.
- Expanda o contraiga los subdirectorios haciendo doble clic en el icono de carpeta.
- Para mostrar/ocultar las columnas *Tamaño* y *Última modificación*, haga clic con el botón derecho en la barra de título de la ventana de resultados y haga clic en la opción correspondiente.



- Para cambiar el ancho de las columnas arrastre el borde lateral de la columna.
- También puede maximizar o minimizar la ventana de resultados y ajustar su tamaño.

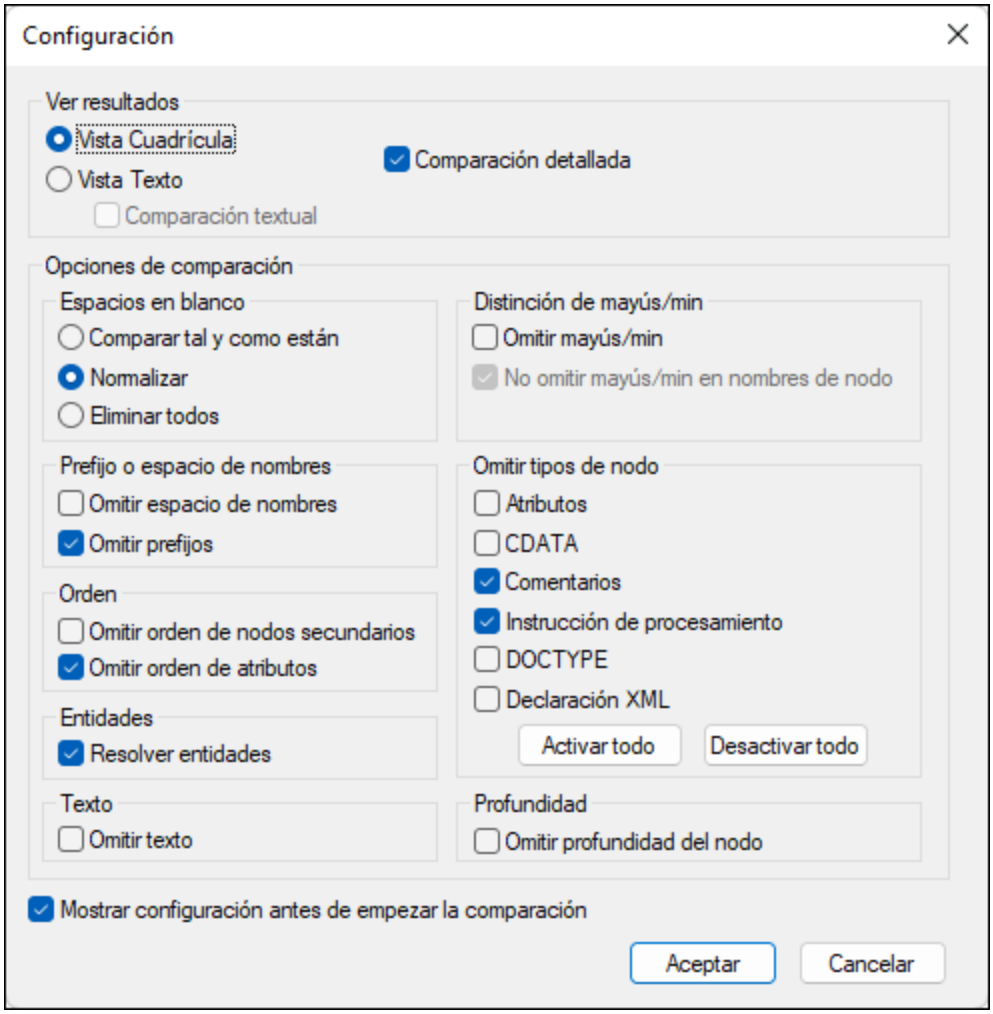
Comparar y combinar archivos

Al hacer doble clic en una línea se abren los dos archivos de esa línea en la ventana principal y se inicia la comparación de archivos automáticamente. Puede seguir comparando los archivos como en cualquier otra sesión de comparación (Comparar el archivo con... 1547).

© 2016-2022 Altova GmbH

28.17.5.3 Opciones de comparación

Haga clic en **Herramientas | Opciones de comparación** para abrir el cuadro de diálogo "Configuración" (*imagen siguiente*). En este cuadro de diálogo puede configurar la sesión de comparación. Las opciones que están activas en este diálogo cuando se inicia una sesión de comparación son las opciones que se aplican a la sesión de comparación.



Ver resultados

Seleccione en qué vista se muestran los resultados. Puede elegir una de estas opciones:

- Vista Cuadrícula (comparación XML)
- Vista Texto con la casilla *Comparación textual* desactivada (comparación XML)
- Vista Texto con la casilla *Comparación textual* activada (comparación textual)

Si selecciona una vista de comparación XML, los documentos se tratan como documentos XML y se habilitan las opciones de comparación XML. Si marca la opción *Comparación textual*, se habilitan solamente las

opciones de comparación válidas para la comparación textual (*Espacios en blanco* y *Distinción de mayús/min*) y las demás opciones se deshabilitan.

Nota: puede combinar las diferencias encontradas en los archivos en las vistas Texto y Cuadrícula, tanto en el modo de comparación XML como en el textual. Para deshacer la combinación de diferencias, detenga la sesión de comparación, seleccione el archivo donde desea deshacer los cambios y seleccione **Edición | Deshacer** o pulse **Ctrl + Z**.

Comparación detallada

Si se desactiva esta opción, las diferencias entre elementos inmediatos del mismo nivel se señalan como una sola diferencia y la opción de combinación se deshabilita. Si se activa esta opción, las diferencias entre elementos inmediatos del mismo nivel se señalan como diferencias distintas y se habilita la opción de combinación.

Nota: para poder realizar operaciones de combinación es necesario marcar la casilla *Comparación detallada*.

Espacios en blanco

Se consideran espacios en blanco los espacios, las tabulaciones, los retornos de carro y los saltos de línea. Si se normalizan los espacios en blanco, se sustituirán todos los caracteres de espacio en blanco consecutivos con un solo carácter de espacio en blanco. Sin embargo, recuerde que según la especificación XML, los espacios en blanco iniciales y finales de los valores de atributo se eliminan por completo cuando se normalizan los espacios en blanco. Puede elegir estas opciones de comparación: (i) comparar los espacios en blanco tal y como están, (ii) comparar los archivos con los espacios en blanco normalizados y (iii) eliminar todos los espacios en blanco antes de realizar la comparación. Estas opciones están disponibles tanto para la comparación XML como para la textual.

Distinción de mayúsculas y minúsculas

Si marca la casilla *Omitir mayús/min*, también tiene la opción de omitir o no las mayúsculas y minúsculas en los nombres de nodo (solamente en comparaciones XML). Las opciones de este panel están disponibles tanto para la comparación XML como para la textual.

Prefijo o espacio de nombres

Aquí puede elegir si los espacios de nombres y los prefijos se tienen en cuenta o no durante la comparación.

Orden

Si marca la casilla *Omitir orden de nodos secundarios*, no importa la posición de los nodos secundarios con respecto a la de los demás. La comparación toma todo el conjunto de nodos secundarios y, si la única diferencia entre los dos archivos es la posición relativa del conjunto de nodos, la diferencia se omite.

Nota: los nodos secundarios se identifican por su nombre, sus atributos y su posición. Si varios nodos del conjunto del mismo nivel tienen el mismo nombre, entonces se usa la posición de estos nodos para identificar su posición, aunque la opción *Omitir orden de nodos secundarios* esté marcada. Esta opción afecta a cada nivel por separado.

Si no marca la casilla *Omitir orden de nodos secundarios*, se señala como diferencias el orden diferente de los nodos.

© 2016-2022 Altova GmbH

También hay una opción para omitir el orden de los atributos en la comparación y afecta al orden de los atributos de cada elemento.

Entidades

Si marca la casilla *Resolver entidades,* se resuelven todas las entidades del documento. De lo contrario, los archivos se comparan con las entidades tal y como están.

Texto

Si marca la casilla *Omitir texto,* no se señalan las diferencias encontradas en los nodos de texto.

Omitir tipos de nodo

Aquí puede elegir qué tipos de nodos se comparan y cuáles no. Los tipos de nodos que se pueden omitir de la comparación son los atributos, las secciones CDATA, los comentarios, las instrucciones de procesamiento, las instrucciones DOCTYPE y la declaración XML.

Profundidad

Aquí puede elegir si se tiene en cuenta o no la profundidad adicional de cada elemento (es decir, si el elemento tiene más niveles de descendientes) en relación con la profundidad del elemento correspondiente del otro archivo. Para poder realizar operaciones de combinación de diferencias es necesario marcar la casilla *Omitir profundidad del nodo*.

Mostrar configuración antes de empezar la comparación

Si marca esta casilla, este cuadro de diálogo aparece antes de cada comparación de archivos/directorios (al hacer clic en los comandos **Comparar el archivo con...** y **Comparar directorios**). Esto puede ser práctico si desea repasar la configuración de la configuración o cambiarla en cada comparación.

Si desactiva esta casilla, la sesión de comparación se inicia nada más invocar la comparación.

28.17.6 Herramientas definidas por el usuario

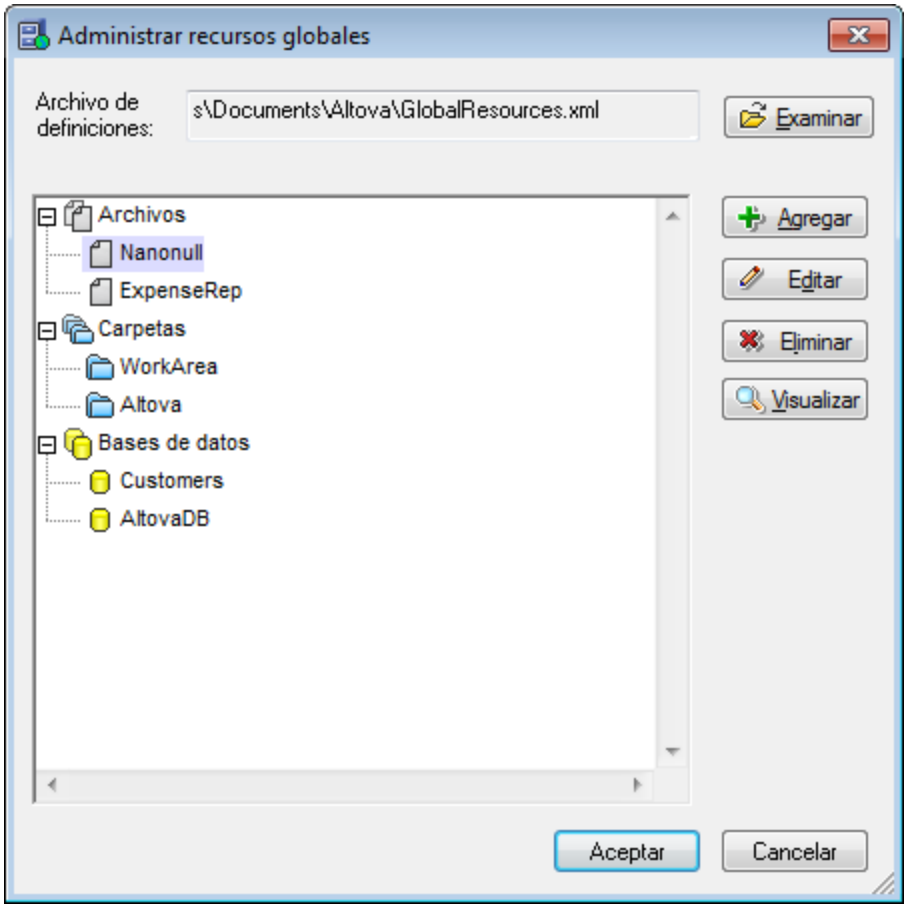
Al pasar el puntero del ratón por el comando **Herramientas definidas por el usuario** aparece un submenú con comandos hechos a medida que usan aplicaciones externas. Para crear estos comandos, use la pestaña *Herramientas* del cuadro de diálogo Personalizar. Al hacer clic en uno de estos comandos personalizados, se ejecuta la acción asociada al comando.

El comando **Herramientas definidas por el usuario | Personalizar** abre la pestaña *Herramientas* del cuadro de diálogo Personalizar (donde puede crear comandos hechos a medida para el submenú del comando **Herramientas definidas por el usuario**.)

28.17.7 Recursos globales

El comando **Recursos globales** abre el cuadro de diálogo "Administrar recursos globales" (*imagen siguiente*), donde puede:

- Seleccionar qué archivo XML de recursos globales se utiliza para los recursos globales.
- Añadir recursos globales (o alias) de tipo archivo, carpeta o base de datos.
- Especificar varias configuraciones para cada recurso global (alias). Cada configuración se asigna a un recurso concreto.



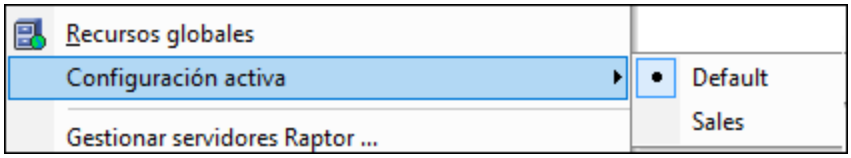
Para más información consulte la sección Definir recursos globales[1042].

Nota: también puede abrir el cuadro de diálogo "Administrar recursos globales" desde la barra de herramientas Recursos globales[1566] (**Herramientas | Personalizar | Barras de herramientas | Recursos globales**).

28.17.8 Configuración activa

Al pasar el puntero del ratón sobre el comando **Configuración activa** aparece un submenú con todas las configuraciones definidas en el archivo XML de recursos globales[1557] activo (*imagen siguiente*).

© 2016-2022 Altova GmbH

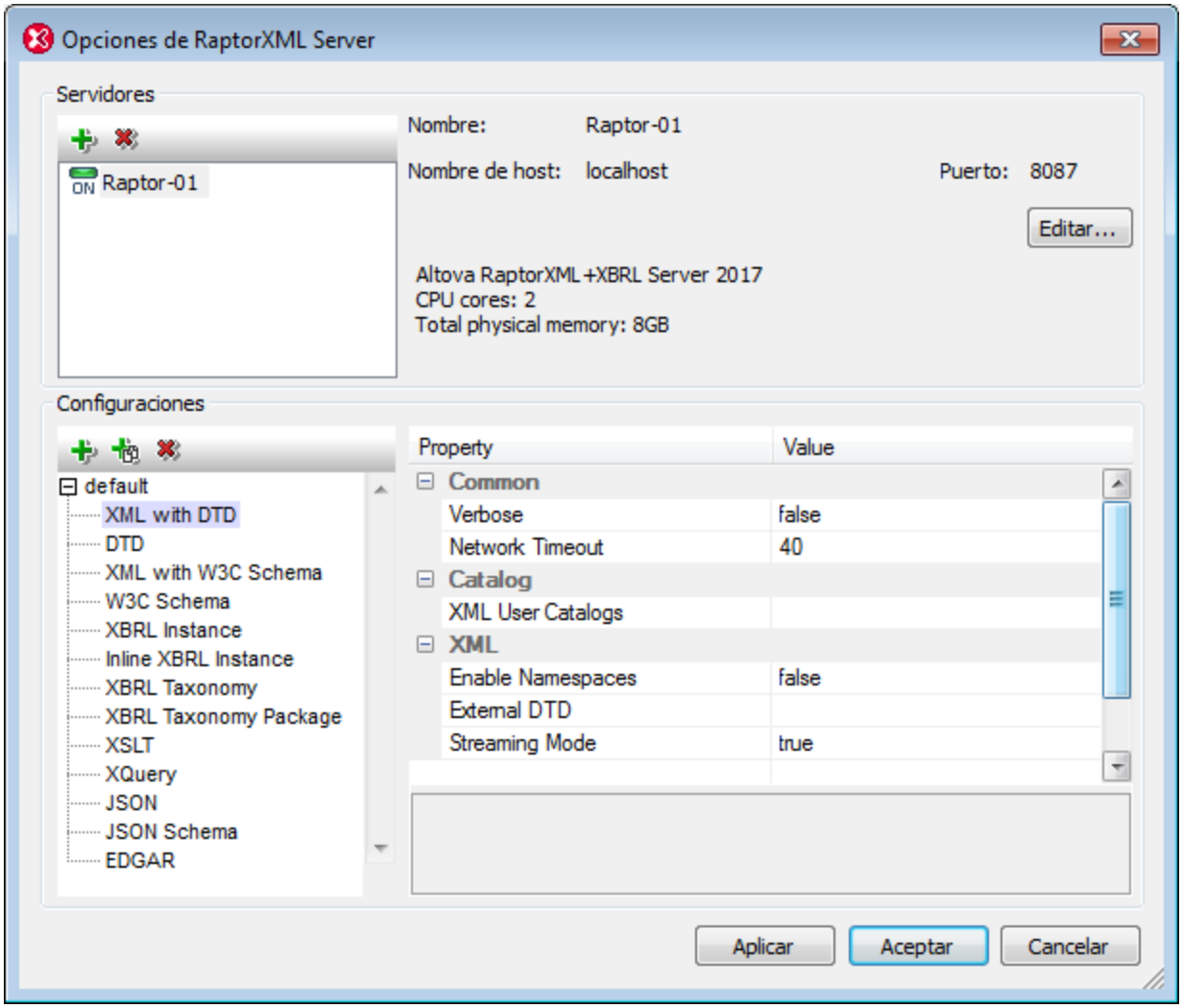


La configuración que está activa actualmente se identifica con un punto. En la imagen anterior, por ejemplo, la configuración activa es `Default`. Para cambiar de configuración simplemente haga clic en otra configuración.

Nota: también puede cambiar de configuración en la barra de herramientas Recursos globales[1566] (**Herramientas | Personalizar | Barras de herramientas | Recursos globales**).

28.17.9 Gestionar servidores Raptor

El comando **Gestionar servidores Raptor** permite añadir varios servidores RaptorXML al repertorio de servidores disponibles y definir configuraciones para cada uno de ellos. Para más información sobre validaciones de datos XML y XBRL con RaptorXML Server consulte la sección Validar con RaptorXML Server[1068].



Agregar un servidor RaptorXML

En el panel *Servidores* del cuadro de diálogo (*imagen siguiente*) haga clic en el icono **Agregar servidor** y escriba el nombre con el quiere identificar al servidor Raptor, el nombre de red del equipo donde está instalado Raptor (el nombre de host) y el puerto. Haga clic en **Aceptar** para guardar la configuración.



© 2016-2022 Altova GmbH

- *Nombre:* cualquier cadena de texto. Se usa en XMLSpy para identificar un servidor Raptor concreto.
- *Nombre de host:* el nombre o dirección IP del equipo de red donde está instalado el servidor Raptor. El procesamiento será más rápido si utiliza una dirección IP en lugar de un nombre de host. La dirección IP que corresponde a `localhost` (el equipo local) es `127.0.0.1`.
- *Puerto:* el puerto por el que se accede al servidor Raptor. Este puerto se especifica en el archivo de configuración de Raptor (llamado `server_config.xml`). El puerto debe ser fijo y debe conocerse para que las solicitudes se puedan dirigir correctamente al servicio. Para más información consulte los manuales del usuario de *RaptorXML Server* y *RaptorXML+XBRL Server*.

Haga clic en **Aceptar** cuando termine de introducir la información del servidor. El nombre de servidor aparecerá en la lista de servidores (en el panel izquierdo). Además, el icono verde situado junto al nombre del servidor indica que ese servidor Raptor está encendido y en ejecución. Los datos del servidor aparecen en el panel. Si el icono es de color rojo, esto significa que el servidor no está conectado. Por otra parte, si no se encuentra el servidor, entonces aparece un mensaje de error.

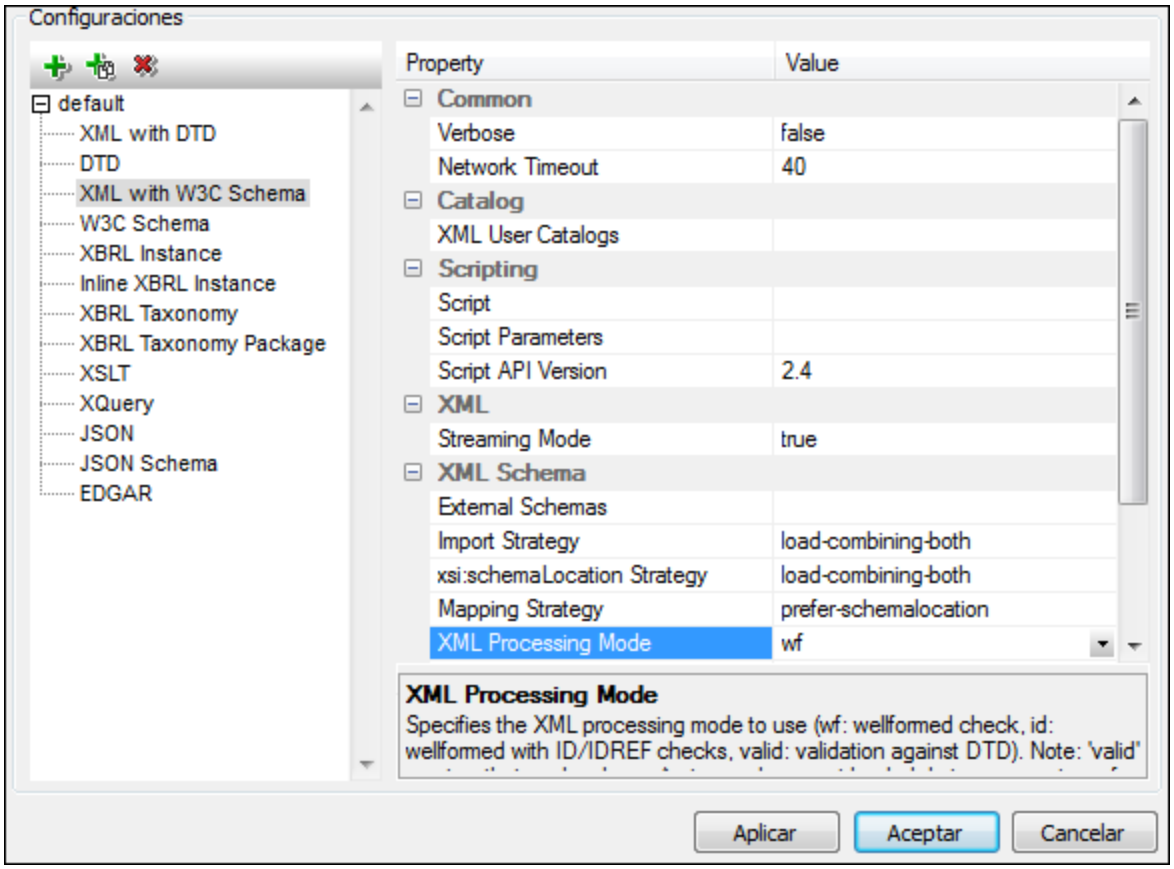
Nota: el servidor Raptor debe estar en ejecución cuando se añada. De lo contrario, XMLSpy no podrá obtener ni almacenar información sobre el servidor. Si, tras ser añadido, el servidor se desconecta o no se encuentra, el icono situado junto al nombre del servidor cambiará de color y aparecerá un mensaje de error.

Para editar el nombre, nombre de host o puerto de un servidor basta con seleccionar el servidor en el panel izquierdo y hacer clic en el botón **Editar**. Esto abre un cuadro de diálogo donde puede editar los datos. Para quitar un servidor del repertorio basta con seleccionarlo y hacer clic en el icono **Quitar servidor seleccionado**.

Configuraciones de servidor

Una configuración es un conjunto de opciones de validación de RaptorXML. Cuando se añade un servidor, se añade con la configuración predeterminada `default`. Se trata de un conjunto de opciones RaptorXML que tienen valores predeterminados. Estos valores pueden editarse y también se pueden añadir nuevas configuraciones con otros valores. Tras definir las configuraciones de servidor podrá seleccionar una configuración como configuración activa. Esta configuración activa es la que se utilizará cuando se ejecute el comando **Validar en servidor**.

El panel *Configuraciones* está dividido en dos partes: (i) el lateral izquierdo muestra las configuraciones y los tipos de documentos que se pueden validar y (ii) el lateral derecho muestra todas las opciones del tipo de validación que está seleccionada en el lateral izquierdo. En la parte inferior del cuadro de diálogo encontrará una descripción de la opción seleccionada en el lateral derecho.



Agregar una configuración
Haga clic en **Agregar una configuración** en el panel *Configuraciones* del cuadro de diálogo "Opciones de RaptorXML Server". Esto añade una configuración nueva cuyas opciones tienen valores predeterminados. También puede crear una configuración nueva haciendo clic en Copiar configuración seleccionada. Esto crea una configuración nueva cuyas opciones tienen los mismos valores que la configuración que se copió. Las configuraciones nuevas se crean con nombres predeterminados como `config<X>`. Para cambiar el nombre de una configuración haga doble clic en ella y escriba el nombre nuevo. Después puede editar los valores de las opciones de la configuración.

Editar valores de las opciones de una configuración
Primero seleccione el documento que se debe validar en el panel izquierdo. Esto muestra las opciones para ese tipo de validación en el panel derecho. Siga estos pasos para editar el valor de una opción:

- Si el valor puede ser un valor de un conjunto de valores predefinidos, seleccione el valor en el cuadro combinado de esa opción.
- Si el valor no está restringido, haga clic en el campo de la opción y escriba el valor que desea utilizar.
- Si el valor es una ruta de acceso de archivo, no solo puede introducir el valor sino que además puede buscar el archivo que desea usar haciendo clic en el botón **Examinar** de la opción.

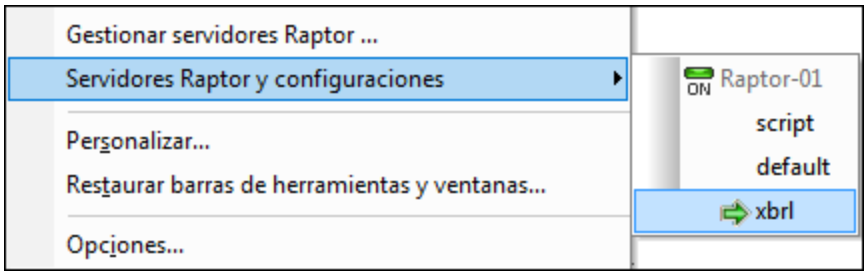
Cuando seleccione una opción su descripción aparecerá en el recuadro situado al final del cuadro de diálogo. Consulte los manuales de RaptorXML Server y RaptorXML(+XBRL) Server para obtener más información sobre las diferentes opciones de RaptorXML.

© 2016-2022 Altova GmbH

Quitar una configuración
Seleccione la configuración en el panel izquierdo y después haga clic en **Quitar configuración seleccionada**.

28.17.10 Servidores Raptor y configuraciones

Si definió varias configuraciones en varios servidores, podrá seleccionar un servidor y una de sus configuraciones como configuración activa. La configuración activa se usará para todas las validaciones siguientes. Pase el puntero por encima del comando **Herramientas | Servidores Raptor y configuraciones** (*imagen siguiente*) para abrir el submenú que contiene todos los servidores que se añadieron, así como sus configuraciones. Seleccione la configuración de servidor que desea usar como configuración activa. Por ejemplo, en la imagen siguiente, se seleccionó la configuración `xbrl` del servidor `Raptor-01`. La fecha verde indica que se trata de la configuración activa.



Consulte la sección Servidor RaptorXML(+XBRL)[1068] para leer un resumen de cómo usar Raptor desde XMLSpy.

28.17.11 Gestor de taxonomías XBRL

Este comando abre el cuadro de diálogo del Gestor de taxonomías, que permite administrar paquetes de taxonomías XBRL.

Para instalar una taxonomía XBRL marque la casilla que hay junto a la taxonomía o taxonomías que quiere instalar y haga clic en **Aplicar**. También puede desinstalar y actualizar taxonomías, así como comprobar si hay versiones nuevas, y en general administrarlas, desde un punto central.

Para más información consulte Gestor de taxonomías[818].

28.17.12 Gestor de esquemas XML

Este comando abre el cuadro de diálogo del Gestor de esquemas, que permite administrar paquetes de XML Schema.

Para instalar un esquema XML marque la casilla que hay junto al esquema que quiere instalar y haga clic en **Aplicar**. También puede desinstalarlos, actualizarlos, comprobar si hay esquemas nuevos y, en general, administrar todos los esquemas desde un punto central.

Para más información consulte Gestor de esquemas[426].

28.17.13 Personalizar

El comando **Personalizar** sirve para adaptar los menús y barras de herramientas de la aplicación a sus requisitos. Al hacer clic en este comando aparece el cuadro de diálogo "Personalizar", que está compuesto por estas pestañas:

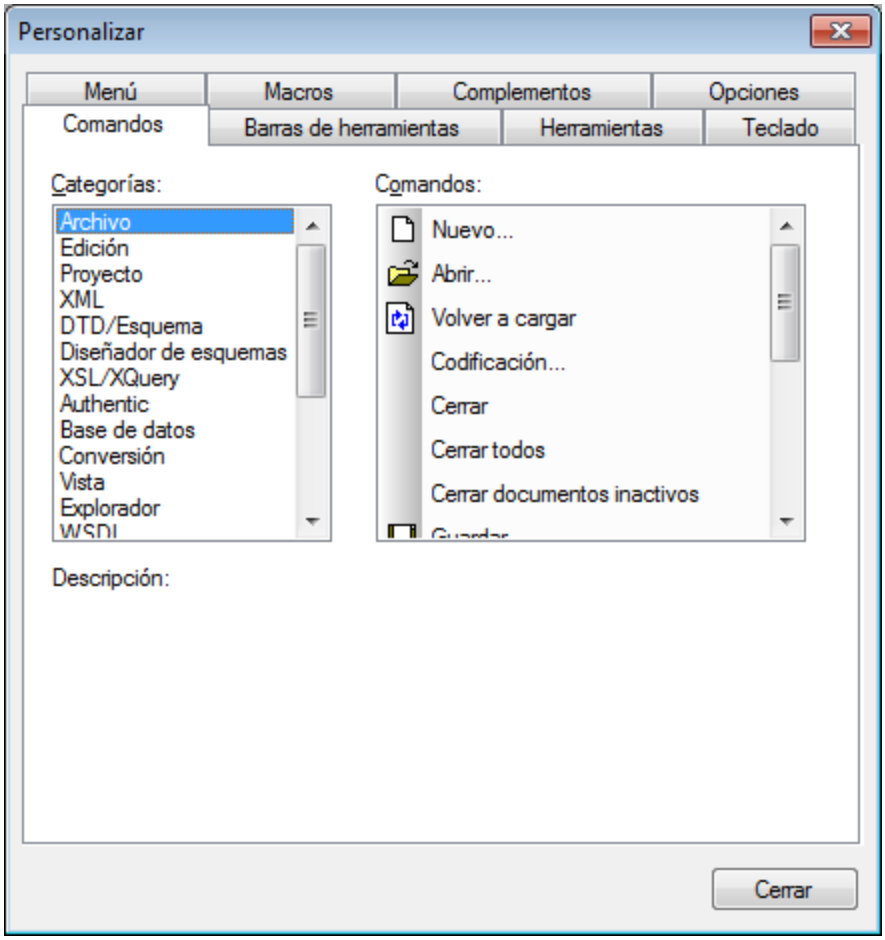
- *Comandos*[1564]: desde esta pestaña puede arrastrar comandos de la aplicación y de macros hasta barras de menú, menús y barras de herramientas.
- *Barras de herramientas*[1566]: aquí puede activar, desactivar y restaurar cada una de las barras de herramientas.
- *Herramientas*[1538]: aquí puede añadir comandos que abren programas externos desde dentro de la interfaz.
- *Teclado*[1569]: aquí puede crear teclas de acceso directo para cada comando de la aplicación y de macros.
- *Menú*[1573]: aquí puede seleccionar y activar las barras de menú y los menús contextuales que desea personalizar. Esta pestaña funciona conjuntamente con la pestaña *Comandos*.
- *Macros*[1575]: aquí puede asignar comandos nuevos a las macros.
- Complementos[1577]: aquí puede activar complementos e integrarlos a la interfaz.
- Opciones[1583]: aquí puede configurar la presentación de las barras de herramientas.

En este apartado también describimos el menú contextual[1580] que aparece si hace clic con el botón derecho en una barra de menú, en un menú o en una barra de herramientas de la aplicación mientras está abierto el cuadro de diálogo "Personalizar".

28.17.13.1 Comandos

En la pestaña ***Comandos*** puede personalizar sus menús y barras de herramientas y añadir comandos a los menús y a las barras de herramientas, dependiendo de lo que necesite. No obstante, tenga en cuenta que no puede crear comandos ni menús nuevos para la aplicación.

© 2016-2022 Altova GmbH



Para añadir un comando a una barra de herramientas o menú:

1. Seleccione el comando **Herramientas | Personalizar**. Se abre el cuadro de diálogo "Personalizar".
2. Seleccione la pestaña ***Comandos***. En el cuadro de lista *Categorías* seleccione la opción **Todos los comandos**. Todos los comandos disponibles aparecen en el cuadro de lista *Comandos*.
3. Haga clic en un comando del cuadro de lista *Comandos* y arrástrelo a un menú o barra de herramientas ya existente. Al pasar el puntero por encima de una posición donde se puede colocar el comando aparece el icono **I**.
4. Cuando encuentre la posición donde desea colocar el comando, suelte el botón del ratón.

Tenga en cuenta que:

- Mientras arrastra el comando, aparece un pequeño botón al final del puntero del ratón. Esto indica que el comando está siendo arrastrado.
- Si el comando no se puede colocar en la posición actual del cursor, debajo del puntero aparece una **X**.
- Si el cursor está en una posición donde se puede colocar el comando (en una barra de herramientas o en un menú), la **X** desaparece y el icono **I** indica que la posición es válida.
- Los comandos se pueden colocar en menús o barras de herramientas. Si creó una barra de herramientas nueva[1566], puede usar este mecanismo de personalización para rellenar la barra de herramientas con comandos.

- Si pasa el cursor por un menú que está cerrado, el menú se abre y puede insertar el comando en cualquier parte del menú.

Agregar comandos a menús contextuales

También puede añadir comandos a menús contextuales arrastrando comandos del cuadro de lista *Comandos* hasta el menú contextual:

1. Haga clic en la pestaña *Menú*[1573] del cuadro de diálogo "Personalizar".
2. En el cuadro combinado del panel *Menús contextuales* seleccione un menú contextual. El menú contextual seleccionado aparece en pantalla.
3. Vuelva a la pestaña *Comandos* del cuadro de diálogo "Personalizar".
4. Seleccione un comando en el cuadro de lista *Comandos* y arrástrelo hasta la posición deseada del menú contextual.

Eliminar un comando o menú

Por último, puede eliminar un comando de un menú, menú contextual (ver párrafo anterior) o barra de herramientas o eliminar un menú entero:

1. Abra el cuadro de diálogo "Personalizar" (**Herramientas | Personalizar**).
2. Seleccione cualquier pestaña del cuadro de diálogo "Personalizar". Haga clic con el botón derecho en un menú o comando de menú y seleccione **Eliminar** en el menú contextual que aparece. Si lo prefiere, también puede arrastrar el menú o comando de menú hasta que aparezca el icono **X** debajo del puntero del ratón y suelte el menú o comando de menú. Como resultado se elimina el menú o comando de menú.

Para volver a instalar los comandos de menú eliminados, utilice los mecanismos descritos en este apartado. Para restablecer un menú eliminado, seleccione **Herramientas | Personalizar | Menú** y pulse el botón **Restaurar** del panel *Menús del marco de la aplicación*. Otra opción es seleccionar **Herramientas | Personalizar | Barras de herramientas**, hacer clic en la barra de herramientas pertinente y pulsar el botón **Restaurar**.

28.17.13.2 Barras de herramientas

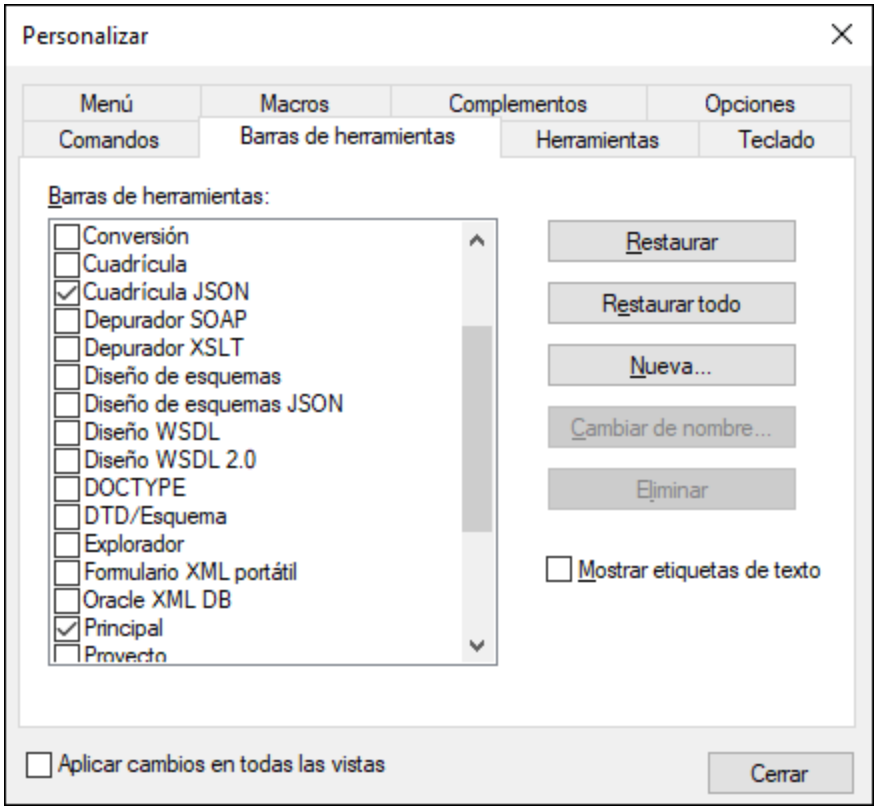
En la pestaña ***Barras de herramientas*** puede: (i) activar o desactivar barras de herramientas (es decir, decidir qué barras de herramientas aparecen en la interfaz), (ii) definir qué iconos aparecen en cada barra de herramientas y (iii) crear barras de herramientas personalizadas.

Las barras de herramientas incluyen iconos para los comandos de menú más utilizados. Además, al pasar el puntero sobre un icono, se ofrece información rápida sobre el icono en un mensaje emergente y en la barra de estado de la aplicación. Las barras de herramientas se pueden colocar en cualquier posición de la pantalla, donde aparece como ventana flotante.

Nota: para añadir un comando a una barra de herramientas, arrastre un comando del cuadro de lista *Comandos* de la pestaña ***Comandos***[1564] hasta la barra de herramientas. Para eliminar un comando de una barra de herramientas, abra el cuadro de diálogo Personalizar, seleccione cualquier pestaña y arrastre el comando fuera de la barra de herramientas (para más información consulte el apartado Comandos[1564]).

© 2016-2022 Altova GmbH

Nota: la configuración de una barra de herramientas definida en una vista determinada afecta solamente a dicha vista. Para que la configuración afecte a todas las vistas, marque la casilla situada al final de la pestaña *Barras de herramientas* del cuadro de diálogo Personalizar.



En esta pestaña puede llevar a cabo estas operaciones:

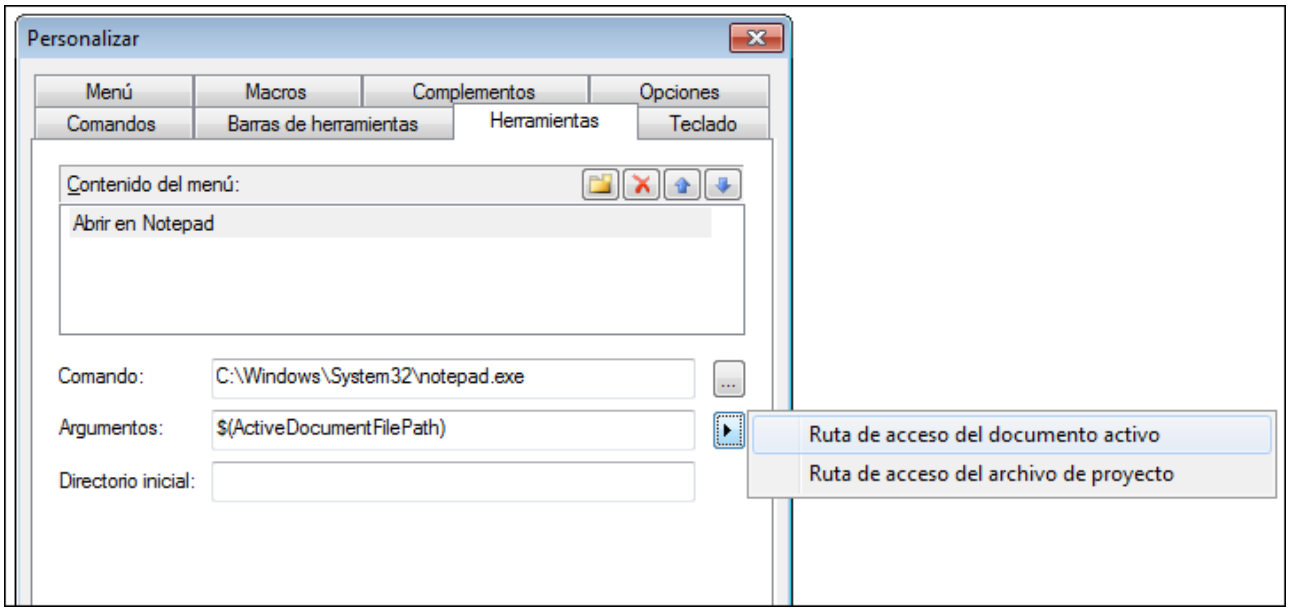
- *Para activar/desactivar una barra de herramientas:* marque su casilla en el cuadro de lista *Barras de herramientas*.
- *Para aplicar los cambios a todas las vistas:* marque la casilla situada al final de la pestaña. De lo contrario, los cambios realizados afectan solamente a la vista activa. Recuerde que los cambios realizados **después** de marcar la casilla *Aplicar cambios en todas las vistas* afectarán a todas las vistas.
- *Para añadir una barra de herramientas nueva:* pulse el botón **Nueva...** y escriba el nombre de la barra de herramientas nuevas en el cuadro de diálogo "Nombre de la barra de herramientas" que aparece. Arrastre comandos desde la pestaña *Comandos* 1564 hasta la barra de herramientas nueva.
- *Para cambiar el nombre de una barra de herramientas nueva:* seleccione la barra de herramientas en el panel *Barra de herramientas* y pulse el botón **Cambiar de nombre**. Edite el nombre en el cuadro de diálogo "Nombre de la barra de herramientas" que aparece.
- *Para restaurar la barra de menús:* seleccione `Barra de menús` en el panel *Barras de herramientas* y pulse el botón **Restaurar**. La barra de menús vuelve a su estado original de instalación.
- *Para restaurar todas las barras de herramientas y comandos de menú:* pulse el botón **Restaurar todo**. Todas las barras de herramientas y menús vuelven a su estado original de instalación.

- *Para eliminar una barra de herramientas:* seleccione la barra de herramientas en el panel *Barras de herramientas* y pulse el botón **Eliminar**.

- *Para mostrar las etiquetas de texto de una barra de herramientas:* seleccione la barra de herramientas y marque la casilla *Mostrar etiquetas de texto.* Recuerde que debe activar las etiquetas de texto de cada barra de herramientas por separado.

28.17.13.3 Herramientas

En la pestaña ***Herramientas*** puede crear comandos para poder usar aplicaciones externas desde XMLSpy directamente. Estos comandos se añaden al menú **Herramientas | Herramientas definidas por el usuario**. Por ejemplo, el archivo activo de la ventana principal de XMLSpy se puede abrir en una aplicación externa como Notepad haciendo clic en un comando del menú **Herramientas | Herramientas definidas por el usuario** creado previamente.



Para crear un comando y usarlo con una aplicación externa siga estos pasos:

1. En el panel *Contenido del menú* (*imagen anterior*), haga clic en el icono **Nuevo** de la barra de título del panel y aparece una línea nueva. Escriba el nombre del comando de menú que desea crear. En el ejemplo de la imagen anterior introdujimos el comando de menú **Abrir en Notepad**. La idea es usar este comando para abrir el documento activo en la aplicación externa Notepad. Puede añadir más comandos a la lista de comandos haciendo clic en el icono **Nuevo**. Para subir o bajar un comando de la lista, use los iconos **Subir** y **Bajar** respectivamente. Para eliminar un comando, selecciónelo y haga clic en el icono **Eliminar**.
2. Para asignar una aplicación externa al comando, seleccione el comando en el panel *Contenido del menú*. En el campo *Comando* escriba la ruta de acceso del archivo ejecutable de la aplicación externa o búsquelo. En el ejemplo de la imagen anterior introdujimos la ruta de acceso de la aplicación Notepad en el campo *Comando*.

© 2016-2022 Altova GmbH

3. Las acciones que se pueden llevar a cabo con la aplicación externa aparecen al hacer clic en el botón desplegable del campo *Argumentos* (*imagen anterior*). Estas acciones se describen más abajo. Seleccione una acción y en el campo *Argumentos* aparece una cadena de código para la acción elegida.
4. Si desea especificar un directorio de trabajo actual, introdúzcalo en el campo *Directorio inicial*.
5. Haga clic en **Cerrar** para finalizar.

Los comandos creados en esta pestaña aparecen en el menú **Herramientas | Herramientas definidas por el usuario** y en el menú contextual de los archivos y carpetas de la ventana Proyecto (en el submenú **Herramientas definidas por el usuario)**.

Al hacer clic en uno de los comandos de **Herramientas | Herramientas definidas por el usuario**, se ejecuta la acción asignada al comando. El comando del ejemplo de la imagen hace esto: abre en Notepad el documento que está activo en la ventana principal de XMLSpy. El comando de la aplicación externa también está disponible en el menú contextual de los archivos de la ventana Proyecto (clic con el botón derecho en un archivo de la ventana Proyecto). En la ventana Proyecto también puede abrir varios archivos (si la aplicación externa lo permite) seleccionando varios archivos y haciendo clic en el comando del menú contextual.

Argumentos

El campo *Argumentos* especifica la acción que debe ejecutar el comando de la aplicación externa. Hay dos argumentos posibles:

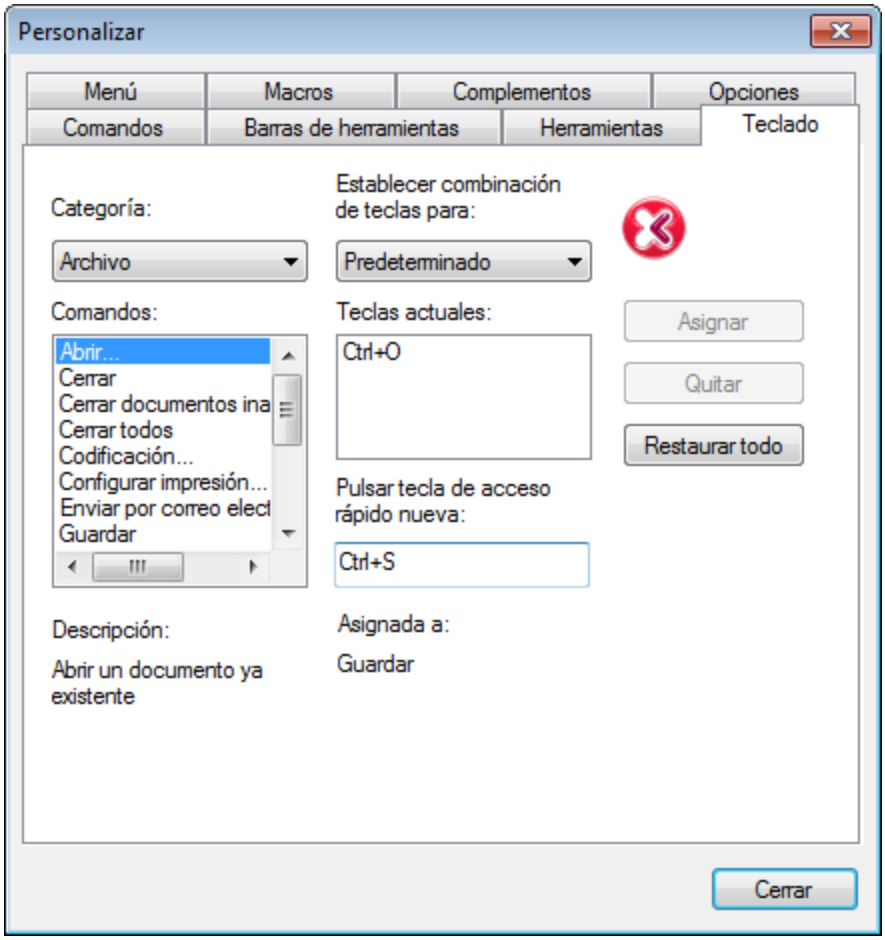
- *Ruta de acceso del documento activo:* el comando de **Herramientas | Herramientas definidas por el usuario** abre en la aplicación externa el documento que está activo en XMLSpy. El comando del menú contextual de un archivo de la ventana Proyecto abre el archivo seleccionado en la aplicación externa
- *Ruta de acceso del archivo de proyecto:* abre en la aplicación externa el archivo de proyecto de XMLSpy (el archivo `.spp`)

Directorio inicial

El campo *Directorio inicial* es opcional y es la ruta de acceso que se usa como directorio actual.

28.17.13.4 Teclado

En la pestaña ***Teclado*** puede crear teclas de acceso rápido nuevas o cambiar las teclas de acceso rápido ya existentes para cualquier comando de la aplicación.



Siga estos pasos para asignar una tecla de acceso rápido nueva a un comando o cambiar una tecla de acceso rápido ya existente:

1. En el cuadro combinado *Categoría* seleccione la opción *Todos los comandos*. Recuerde que si seleccionó una macro como comando asociado[1575], la macro también estará disponible en el cuadro combinado *Categoría* y podrá crear una tecla de acceso rápido para ella.
2. En el cuadro de lista *Comandos* seleccione el comando al que desea asignar una tecla de acceso rápido nueva o el comando cuya tecla de acceso rápido desea cambiar.
3. Haga clic dentro del cuadro *Pulsar tecla de acceso rápido nueva* y pulse la tecla de acceso rápido que desea asignar al comando. La tecla de acceso rápido aparece en el cuadro *Pulsar tecla de acceso rápido nueva.* Si la tecla de acceso rápido no se asignó todavía a ningún comando, se habilita el botón **Asignar**. Si la tecla ya se asignó a un comando, el comando aparece debajo del cuadro y el botón **Asignar** está deshabilitado. (Para borrar el contenido del cuadro *Pulsar tecla de acceso rápido nueva* pulse **Ctrl**, **Alt** o **Mayús**).
4. Haga clic en el botón **Asignar**. La tecla de acceso rápido aparece ahora en el cuadro de lista *Teclas actuales*. Puede asignar varias teclas de acceso rápido al mismo comando si lo desea.
5. Para confirmar los cambios pulse el botón **Cerrar**.

© 2016-2022 Altova GmbH

Eliminar una tecla de acceso rápido.

No se puede asignar la misma tecla de acceso rápido a comandos diferentes. Si desea eliminar una tecla de acceso rápido, selecciónela en el cuadro *Teclas actuales* y después pulse el botón **Quitar**.

Establecer combinación de teclas para

Actualmente solo se pueden establecer combinaciones de teclas predeterminadas.

Teclas de acceso rápido predeterminadas

La tabla que aparece a continuación enumera las teclas de acceso rápido asignadas a los comandos más utilizados en el momento de la instalación. En la aplicación puede ver una lista de los comandos, su descripción y sus teclas de acceso rápido en el cuadro de diálogo "Mapa de teclado" (**Ayuda | Mapa de teclado** 1632).

Teclas de función (también para validación y transformación)

F1	Menú Ayuda
F1 + Alt	Abrir último archivo
F3	Buscar siguiente
F4 + CTRL	Cerrar ventana activa
F4 + Alt	Cerrar XMLSpy
F5	Actualizar
F6 + CTRL	Recorrer ventanas abiertas
F7	Comprobar formato XML
F8	Validar
F10	Transformación XSL
F10 + CTRL	Transformación XSL:FO

Comandos de archivo y de aplicación

Alt + F1	Abrir último archivo
CTRL + O	Abrir archivo
CTRL + N	Archivo nuevo
CTRL + P	Imprimir archivo
CTRL + S	Guardar archivo
CTRL + F4	Cerrar ventana activa
CTRL + F6	Recorrer ventanas abiertas
CTRL + TAB	Cambiar de documento abierto
Alt + F4	Cerrar XMLSpy

Otras teclas

Teclas de flecha arriba/abajo	Mover el cursor o la barra de selección
Esc	Abandonar cambios o cerrar cuadro de diálogo
Entrar	Confirmar selección
Supr	Eliminar carácter o elemento seleccionado
Mayús + Supr	Cortar

- *Comandos de edición*

CTRL + A	Seleccionar todo
CTRL + F	Buscar
CTRL + G	Ir a la línea o carácter
CTRL + H	Reemplazar
CTRL + V	Pegar
CTRL + X	Cortar
CTRL + Y	Rehacer
CTRL + Z	Deshacer

- *Comandos de la vista Texto*

CTRL + E	Pasar de la etiqueta de inicio a la etiqueta final y viceversa
CTRL + Mayús + E	Seleccionar elemento que contiene el cursor
CTRL + Alt + E	Ir al elemento primario
CTRL + "+"	Acercarse
CTRL + "-"	Alejarse
CTRL + 0	Restaurar nivel de zoom
CTRL + rueda de desplazamiento hacia adelante	Acercarse
CTRL + rueda de desplazamiento hacia atrás	Alejarse

- *Comandos de la vista Cuadrícula*

CTRL + D	Anexar CDATA
CTRL + E	Anexar Elemento
CTRL + I	Anexar Atributo
CTRL + M	Anexar comentario
CTRL + T	Anexar texto
CTRL + Mayús + D	Insertar CDATA

© 2016-2022 Altova GmbH

CTRL + Mayús + E	Insertar Elemento
CTRL + Mayús + I	Insertar Atributo
CTRL + Mayús + M	Insertar comentario
CTRL + Mayús + T	Insertar texto
CTRL + Alt + D	Agregar secundario CDATA
CTRL + Alt + E	Agregar secundario Elemento
CTRL + Alt + I	Agregar secundario Atributo
CTRL + Alt + M	Agregar secundario comentario
CTRL + Alt + T	Agregar secundario texto

- *Comandos de la vista Esquema*

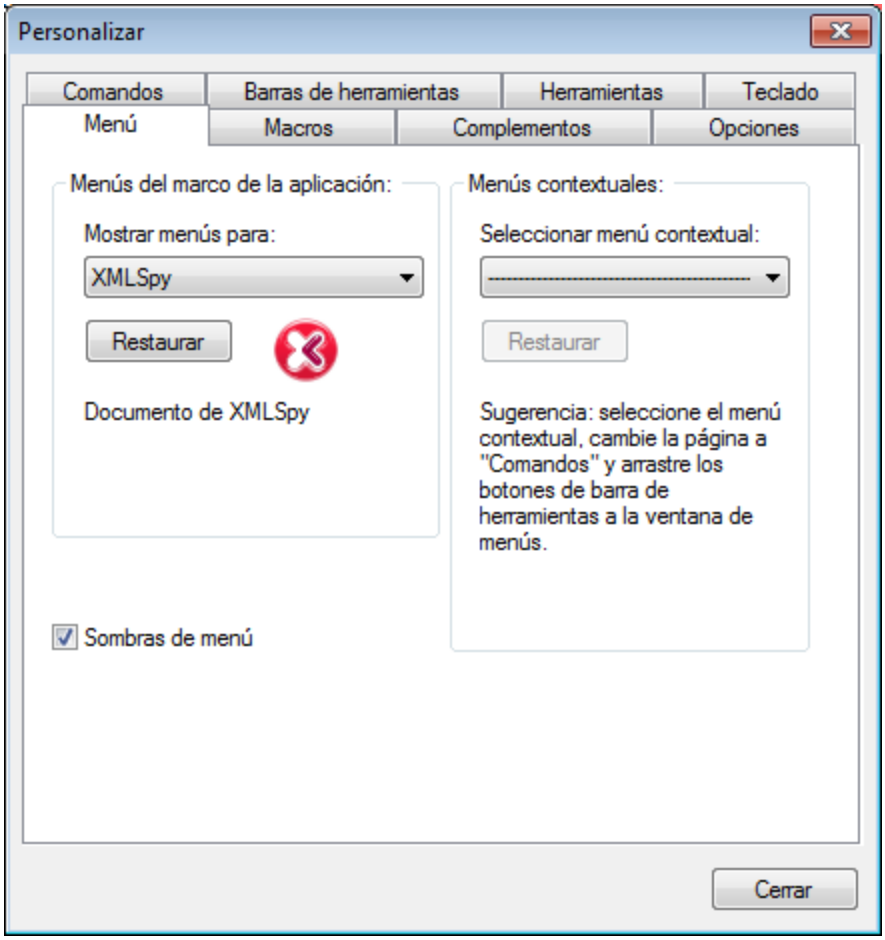
CTRL + doble clic en elemento	Ver definición del elemento

- *Comandos del depurador*

F9	Insertar o quitar punto de interrupción
F9 + Mayús	Insertar o quitar punto de seguimiento
F9 + CTRL	Habilitar o deshabilitar punto de interrupción
F9 + Mayús + CTRL	Habilitar o deshabilitar punto de seguimiento
F11	Paso a paso por instrucciones
F11 + Mayús	Paso a paso para salir
F11 + CTRL	Paso a paso por procedimientos
F11 + Alt	Iniciar el depurador

28.17.13.5 Menú

En la pestaña ***Menú*** puede personalizar las dos barras de menú principales (la barra de menú predeterminada y la barra de menú de la aplicación) así como los menús contextuales de la aplicación.



Personalizar la barra de menú predeterminada y la barra de menú de la aplicación

La barra de menú predeterminada es la barra de menú que aparece cuando no hay ningún documento abierto en la ventana principal. La barra de menú de la aplicación es la barra que aparece cuando hay un documento abierto en la ventana principal. Cada una de estas barras de menú se puede personalizar y los cambios realizados en una de las barras de menú no afecta a la otra.

Para personalizar una barra de menú, selecciónela en el cuadro combinado *Mostrar menús para:* de la pestaña *Menú* (*imagen anterior*). Después cambie a la pestaña *Comandos* del cuadro de diálogo Personalizar y arrastre comandos desde el cuadro de lista *Comandos* hasta la barra de menú.

Eliminar comandos de menús y restaurar las barras de menú

Para **eliminar** un menú entero o un comando de un menú:

1. En el panel *Menús del marco de la aplicación* seleccione *Menú predeterminado* (que muestra los menús disponibles aunque no haya ningún documento abierto) o *XMLSpy* (que muestra los menús disponibles si hay al menos un documento abierto).
2. Con el cuadro de diálogo "Personalizar" abierto, seleccione (i) el menú que quiere eliminar de la barra de herramientas de la aplicación o (ii) el comando que quiere eliminar de uno de estos menús.

© 2016-2022 Altova GmbH

3. Ahora puede (i) arrastrar el menú para eliminarlo de la barra de herramientas o arrastrar el comando de menú para eliminarlo del menú, o bien (ii) hacer clic con el botón derecho en el menú o en el comando de menú y seleccionar **Eliminar**.

Para restaurar estas dos barras de menú (la barra de menú predeterminada y la de la aplicación) a su estado original de instalación seleccione el menú en el cuadro combinado *Mostrar menús para:* y haga clic en el botón **Restaurar** situado bajo el cuadro combinado.

Personalizar los menús contextuales de la aplicación

Los menús contextuales son los menús que aparecen cuando se hace clic con el botón derecho en determinados objetos de la interfaz de la aplicación. Siga estos pasos para personalizar un menú contextual:

1. Seleccione el menú contextual en el cuadro combinado *Seleccionar menú contextual*. Aparece el menú contextual.
2. Pase a la pestaña *Comandos* del cuadro de diálogo Personalizar.
3. Arrastre un comando del cuadro de lista *Comandos* al menú contextual.
4. Si desea eliminar un comando del menú contextual, haga clic en él con el botón derecho y seleccione **Eliminar**. También puede seleccionar el comando y arrastrarlo fuera del menú contextual.

Para restaurar un menú contextual a su estado original de instalación seleccione el menú en el cuadro combinado *Seleccionar menú contextual* y después pulse el botón **Restaurar**, situado bajo el cuadro combinado.

Sombras de menú

Marque la casilla *Sombras de menú* para dar sombra a todos los menús.

28.17.13.6 Macros

En la pestaña ***Macros*** puede crear comandos de aplicación para macros creadas con el editor de script de XMLSpy. Más tarde puede incluir estos comandos de aplicación (que ejecutan las macros asociadas a ellos) en menús y barras de herramientas, bien desde la pestaña *Macros* directamente o bien con los mecanismos disponibles en la pestaña *Comandos* del cuadro de diálogo Personalizar. Al ser comandos de aplicación, también pueden tener asignadas teclas de acceso rápido (desde la pestaña *Teclado* del cuadro de diálogo Personalizar.)

Cómo funcionan las macros en XMLSpy

En XMLSpy las macros funcionan así:

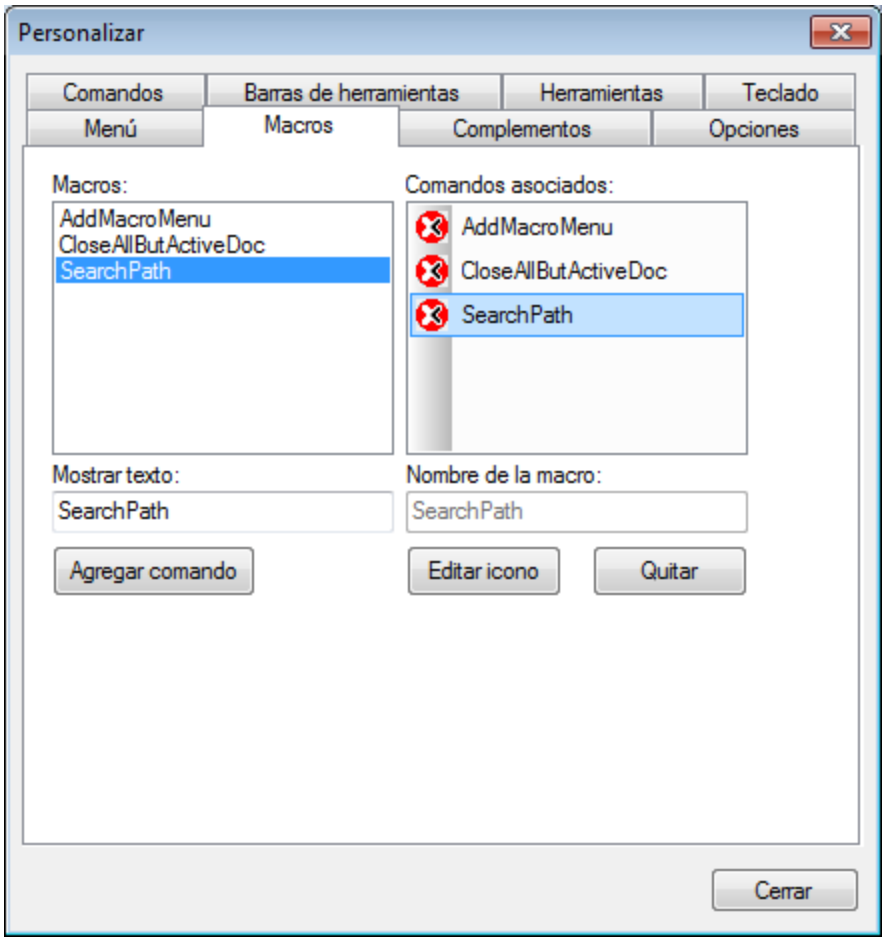
- Los proyectos de script de Altova (archivos `.asprj`) se crean en el editor de script de XMLSpy. Estos proyectos de script contienen las macros que se utilizan en XMLSpy.
- En XMLSpy puede haber dos proyectos de script activos al mismo tiempo: (i) un proyecto de script de aplicación, que se especifica en la sección *Scripting* del cuadro de diálogo "Opciones" y (ii) el proyecto de script del proyecto de XMLSpy, que se especifica en el cuadro de diálogo Script (**Proyecto | Configuración de script**).
- Las macros de estos dos proyectos de script están disponibles en la aplicación en el menú **Herramientas | Macros** (desde donde puede ejecutarlas) y en la pestaña *Macros* del cuadro de diálogo Personalizar (*imagen siguiente*), donde se pueden configurar como comandos de aplicación.

Después de configurar una macro como comando de aplicación, el comando se puede colocar en un menú o barra de herramientas.

Cómo crear un comando de aplicación para una macro

En el editor de script[1641] (**Herramientas | Editor de script**[1641]) cree la macro que desea utilizar y guárdela en un proyecto de script. Después defina esta proyecto de script como proyecto de script de aplicación (en la sección *Scripting*[1625] del cuadro de diálogo "Opciones"[1625]) o como proyecto de script del proyecto activo (en el cuadro de diálogo Script[1328] del proyecto (**Proyecto | Configuración de script**[1328]). A continuación aparecen las macros del proyecto de script en el panel *Macros* de la pestaña *Macros* (*imagen siguiente*).

Para crear un comando de aplicación para una macro, seleccione la macro en el panel *Macros*, escriba el texto del comando en el cuadro *Mostrar texto* y pulse el botón **Agregar comando** (*imagen siguiente*). A continuación se añade a la lista *Comandos asociados* un comando para la macro seleccionada.



Para editar el icono de un comando asociado, seleccione el comando y pulse el botón **Editar icono**. Para eliminarlo, seleccione el comando asociado y pulse **Quitar**.

Cómo poner un comando asociado a una macro en un menú o barra de herramientas

Hay dos maneras de poner un comando asociado a una macro en un menú o en una barra de herramientas:

© 2016-2022 Altova GmbH

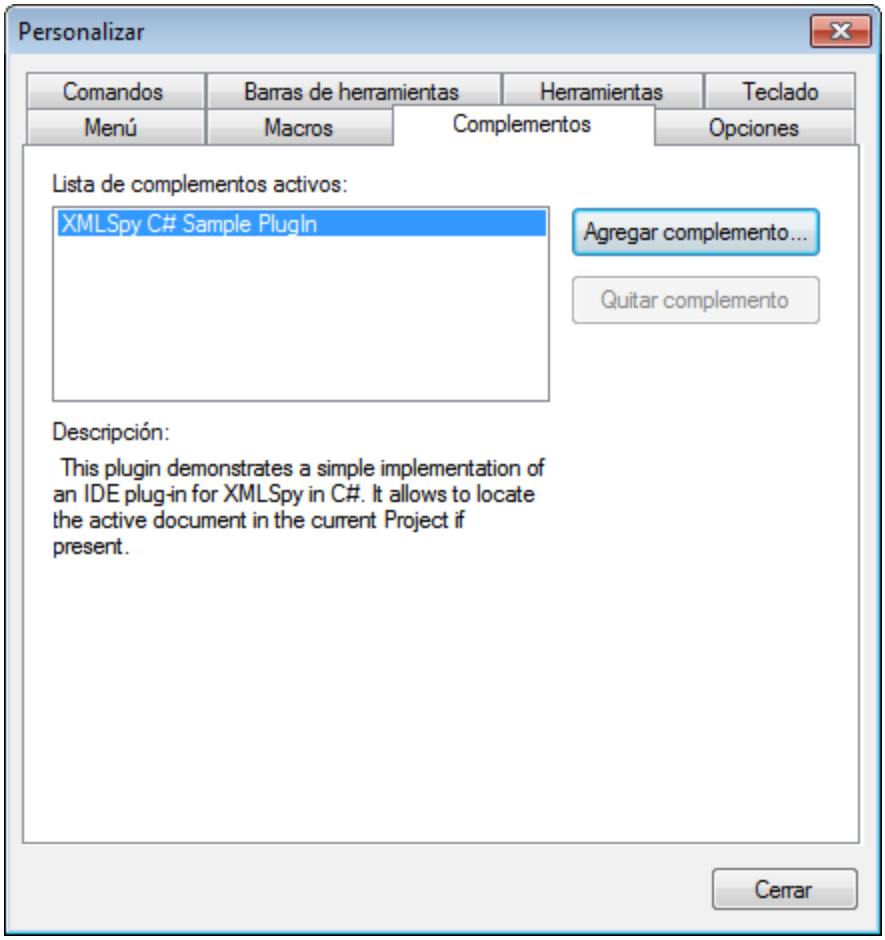
- Arrastrando el comando desde el cuadro de lista *Comandos asociados* hasta la posición deseada en el menú o barra de herramientas de destino.
- Usando los mecanismos disponibles en la pestaña *Comandos* del cuadro de diálogo Personalizar.

En ambos casos se crea el comando en la posición deseada del menú o de la barra de herramientas y al hacer clic en el comando se ejecuta la macro.

Nota: si definió una macro como comando asociado, también puede asignarle una tecla de acceso rápido. En la pestaña *Teclado* del cuadro de diálogo Personalizar, seleccione la opción *Macros* en el cuadro combinado *Categoría.* Después seleccione la macro y defina su tecla de acceso rápido. Para poder asignarle la tecla antes debe definir la macro como comando asociado.

28.17.13.7 Complementos

Desde la pestaña ***Complementos*** puede integrar complementos y colocar comandos (si se han programado) en un menú o en una barra de herramientas de la aplicación. Pulse el botón **Agregar complemento** de la pestaña *Complementos* (*imagen siguiente*) y busque el archivo DLL del complemento (ver más abajo el párrafo *Cómo crear complementos*). Haga clic en **Aceptar** para agregar el complemento. Si quiere, puede agregar varios complementos.



En el cuadro de diálogo aparece una descripción de los complementos añadidos y se habilita el botón **Quitar complemento**. Si el código del complemento crea barras de herramientas y menús, estos aparecen inmediatamente en la interfaz de la aplicación. Si desea quitar un complemento, selecciónelo y pulse el botón **Quitar complemento**.

Cómo crear complementos

En la carpeta `Examples\IDEPlugin` de la carpeta (Mis) Documentos [36] de la aplicación encontrará código fuente para crear ejemplos de complementos en XMLSpy. Siga estas instrucciones para crear un complemento a partir de este código fuente:

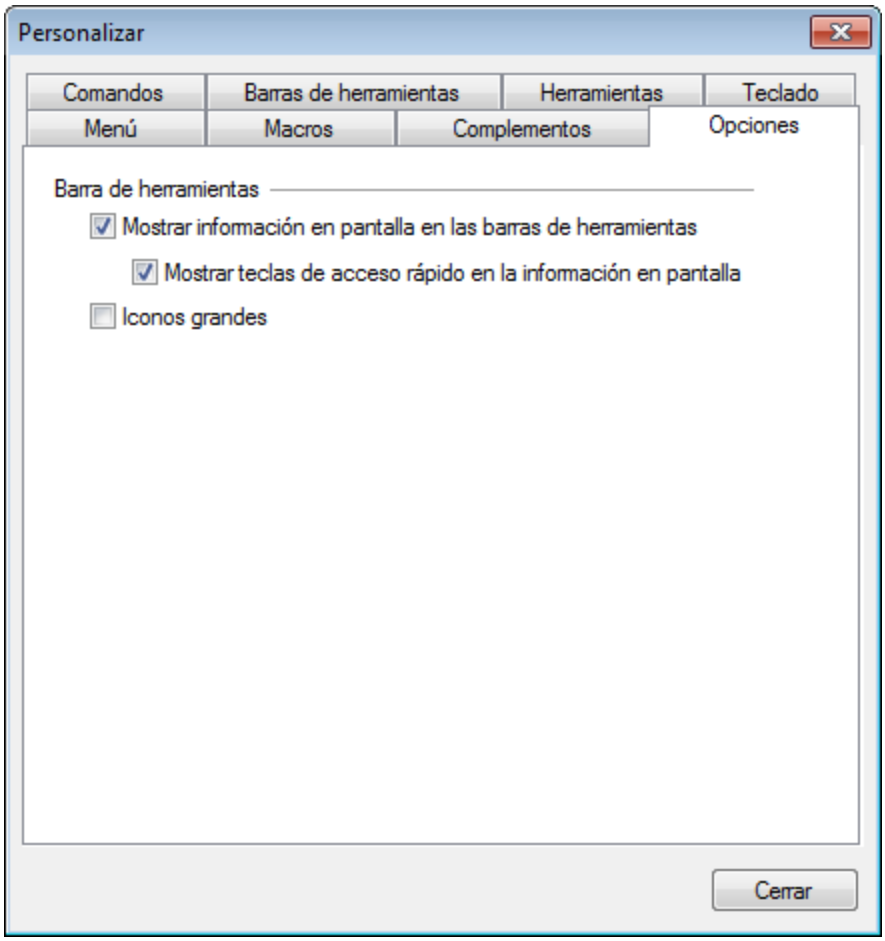
1. Abra la solución que desea crear como complemento en Visual Studio.
2. Cree el complemento con el comando del menú **Compilación**.
3. El archivo DLL del complemento se creará en la carpeta `Bin` o `Debug`. Este archivo DLL es el que se debe añadir como complemento (ver instrucciones más arriba).

Para más información consulte la sección Complementos para entornos IDE [1669].

© 2016-2022 Altova GmbH

28.17.13.8 Opciones

En la pestaña ***Opciones*** puede definir la configuración general del entorno.

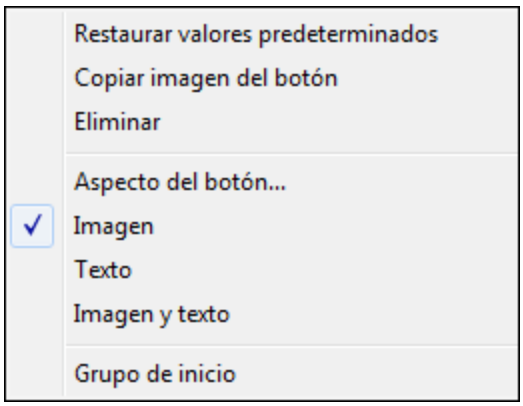


Marque las casillas para activar estas opciones:

- *Mostrar información en pantalla en las barras de herramientas:* al pasar el puntero del ratón sobre los iconos de las barras de herramientas aparece un mensaje emergente con una breve descripción de la función del icono, así como su tecla de acceso rápido (si le asignó una y si marcó la casilla *Mostrar teclas de acceso rápido*).
- *Mostrar teclas de acceso rápido en la información en pantalla:* la información en pantalla de los iconos de las barras de herramientas incluye la tecla de acceso rápido del comando.
- *Iconos grandes*: alterna el tamaño de los iconos de las barras de herramientas entre tamaño estándar y tamaño grande.

28.17.13.9 Menú contextual Personalizar

El menú contextual **Personalizar** (*imagen siguiente*) es el menú que aparece al hacer clic con el botón derecho en un menú, un comando de menú o en un icono de barra de herramientas de la aplicación mientras está abierto el cuadro de diálogo Personalizar.



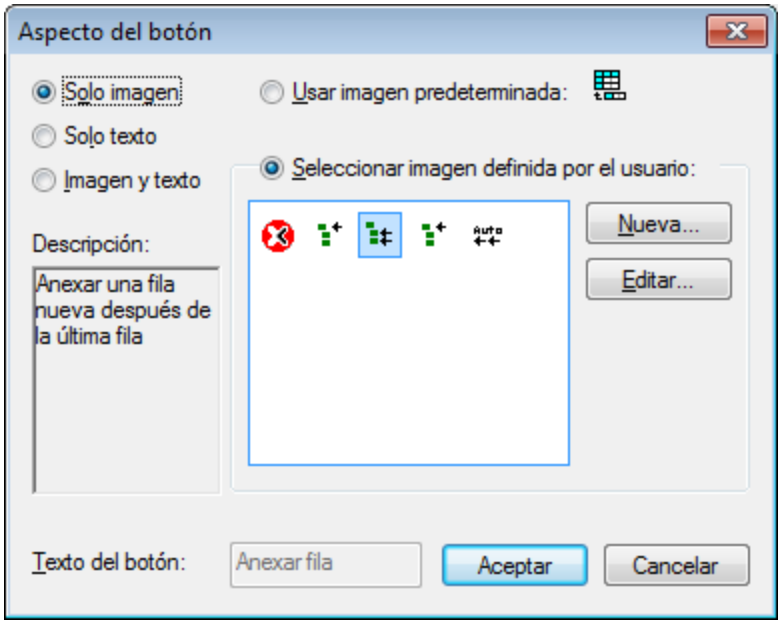
Estas son las opciones del menú contextual **Personalizar**:

- *Restaurar valores predeterminados* (por ahora no tiene función alguna).
- *Copiar imagen del botón:* copia la imagen del icono en el portapapeles.
- *Eliminar:* elimina el menú, comando o icono seleccionado. Para más información sobre cómo restaurar los componentes eliminados lea el último párrafo de este apartado.
- *Aspecto del botón:* abre el cuadro de diálogo "Aspecto del botón" (*imagen siguiente*), donde puede configurar las propiedades que definen el aspecto del icono seleccionado. Consulte el resto de este apartado para más información.
- *Imagen, Texto, Imagen y texto:* estas tres opciones son mutuamente excluyentes y determinan si el icono seleccionado debe ser solamente icono, solamente texto o ambos. Esto también se puede definir en el cuadro de diálogo "Aspecto del botón".
- *Grupo de inicio:* inserta un divisor de grupo vertical a la izquierda del icono seleccionado. Como resultado el icono seleccionado se convierte en el primer icono del grupo.

El cuadro de diálogo "Aspecto del botón"

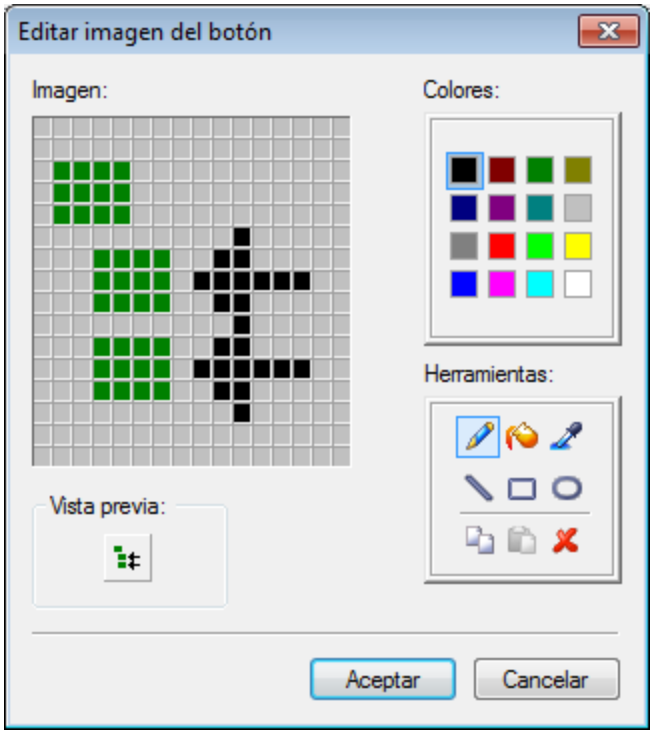
Si hace clic con el botón derecho en un icono de una barra de herramientas (mientras está abierto el cuadro de diálogo Personalizar) y después selecciona **Aspecto del botón**, aparece el cuadro de diálogo "Aspecto del botón" (*imagen siguiente*). En este cuadro de diálogo puede editar la imagen del icono y su texto. Actualmente sólo puede editar iconos de macros y complementos en este cuadro de diálogo.

© 2016-2022 Altova GmbH



Estas son las funciones de edición disponibles en este cuadro de diálogo para el icono seleccionado:

- *Solo imagen, Solo texto* o *Imagen y texto:* seleccione uno de estos tres botones de opción para definir el formato del icono de la barra de herramientas.
- *Edición de la imagen:* si selecciona la opción *Solo imagen* o *Imagen y texto*, se habilitan las opciones de edición de la imagen. Pulse el botón **Nueva** para crear una imagen nueva, que se añadirá al panel de imágenes definidas por el usuario. Después seleccione una imagen y pulse **Editar**.



- *Selección de la imagen:* seleccione una imagen del panel de imágenes definidas por el usuario y después pulse **Aceptar** para usar la imagen seleccionada como icono de la barra de herramientas.
- *Edición y selección del texto:* si selecciona la opción *Solo texto* o *Imagen y texto*, se habilita el cuadro de texto *Texto del botón.* Escriba el texto y haga clic en **Aceptar** para usar este texto con el icono de la barra de herramientas.

Nota: el cuadro de diálogo "Aspecto del botón" también se puede usar para editar el texto de los comandos de menú. Haga clic con el botón derecho en el comando de menú (cuando está abierto el cuadro de diálogo Personalizar), seleccione **Aspecto del botón** y edite el texto del comando de menú en el cuadro de texto *Texto del botón*.

Restaurar menús, comandos e iconos de barras de herramientas eliminados

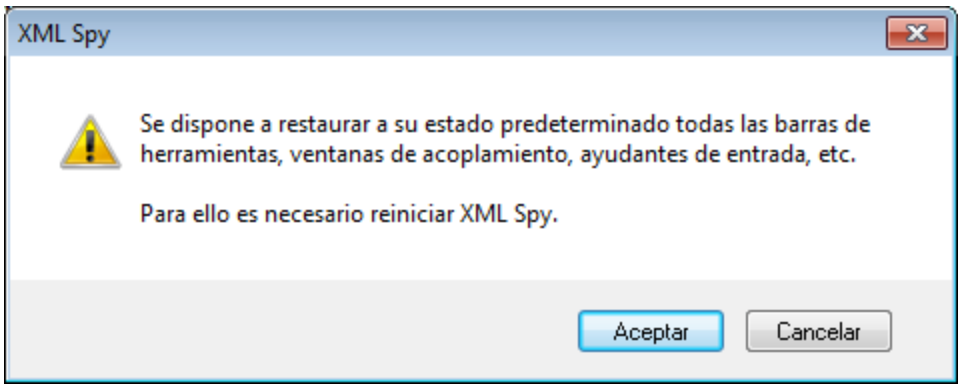
Si elimina un menú, comando o icono de barra de herramientas con el comando **Eliminar** del menú contextual Personalizar, puede restaurarlos siguiendo estos pasos:

- *Menús:* seleccione **Herramientas | Personalizar | Menú**[1573] y haga clic en el botón **Restaurar** del panel *Menús del marco de la aplicación.* Otra opción es seleccionar **Herramientas | Personalizar | Barras de herramientas**[1566], marcar la casilla `Barra de menús` y pulsar el botón **Restaurar**.
- *Comandos de menú:* seleccione **Herramientas | Personalizar | Comandos**[1564] y arrastre el comando desde el cuadro de lista *Comandos* hasta el menú.
- *Iconos de barra de herramientas:* seleccione **Herramientas | Personalizar | Comandos**[1564] y arrastre el comando desde el cuadro de lista *Comandos* hasta la barra de herramientas.

© 2016-2022 Altova GmbH

28.17.14 Restaurar barras de herramientas y ventanas

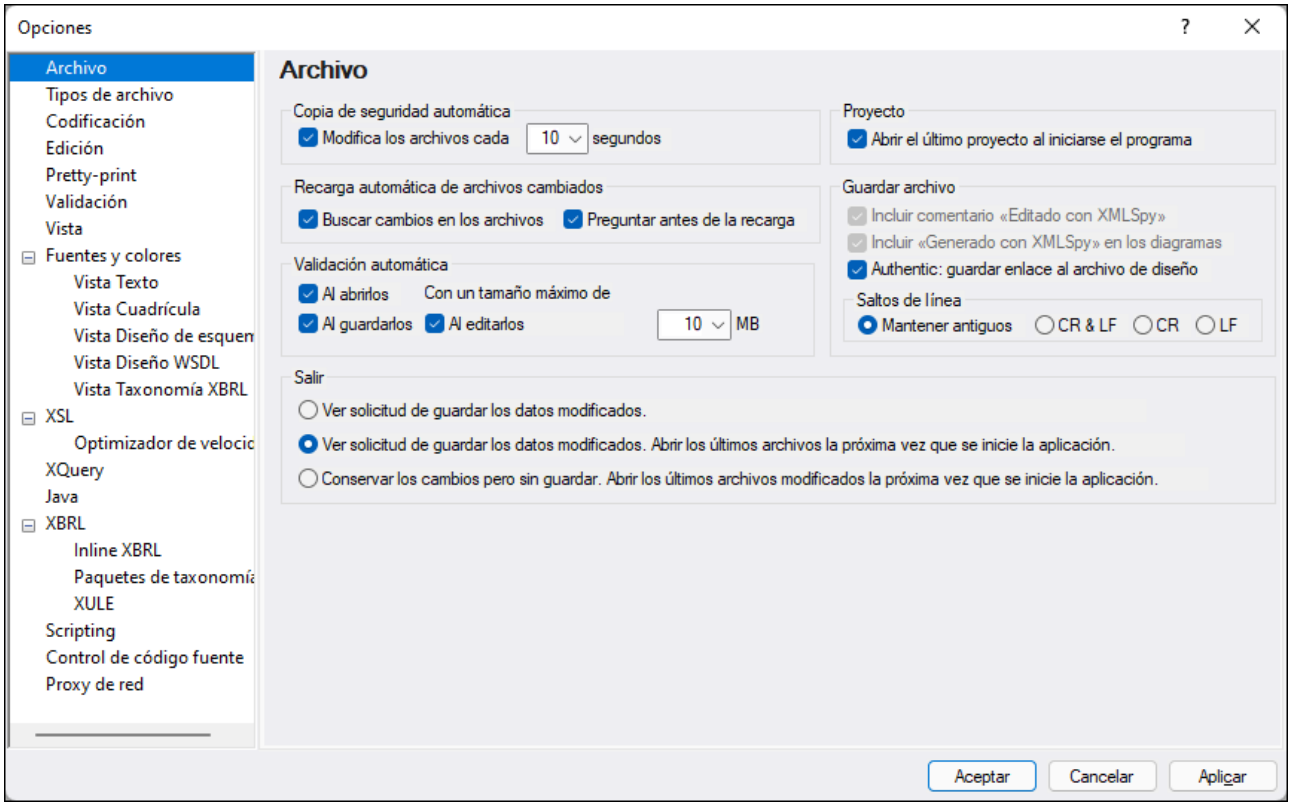
El comando **Restaurar barras de herramientas y ventanas** cierra XMLSpy y lo reinicia con su configuración predeterminada. Antes de cerrarse, XMLSpy le pregunta si desea cerrar o no la aplicación (*imagen siguiente*).



Este comando es muy práctico si movió ventanas o barras de herramientas de sitio, si las ocultó o si ajustó su tamaño y desea poner todas estas barras de herramientas y ventanas como estaban en un principio.

28.17.15 Opciones

El comando **Herramientas** | **Opciones** sirve para definir opciones de configuración globales de la aplicación. Estas opciones están organizadas en secciones (*ver panel izquierdo de la imagen siguiente*). Por ejemplo, la sección *Archivo* 1584 (*imagen siguiente*) contiene opciones que especifican cómo se deben abrir y guardar archivos en XMLSpy. Para definir opciones de configuración de una determinada sección, haga clic en dicha sección en el panel izquierdo del cuadro de diálogo y especifique los valores correspondientes. Al hacer clic en el botón **Aceptar** se guardan los cambios en el registro y se cierra el cuadro de diálogo. El botón **Aplicar** efectúa los cambios en los documentos que están abiertos.

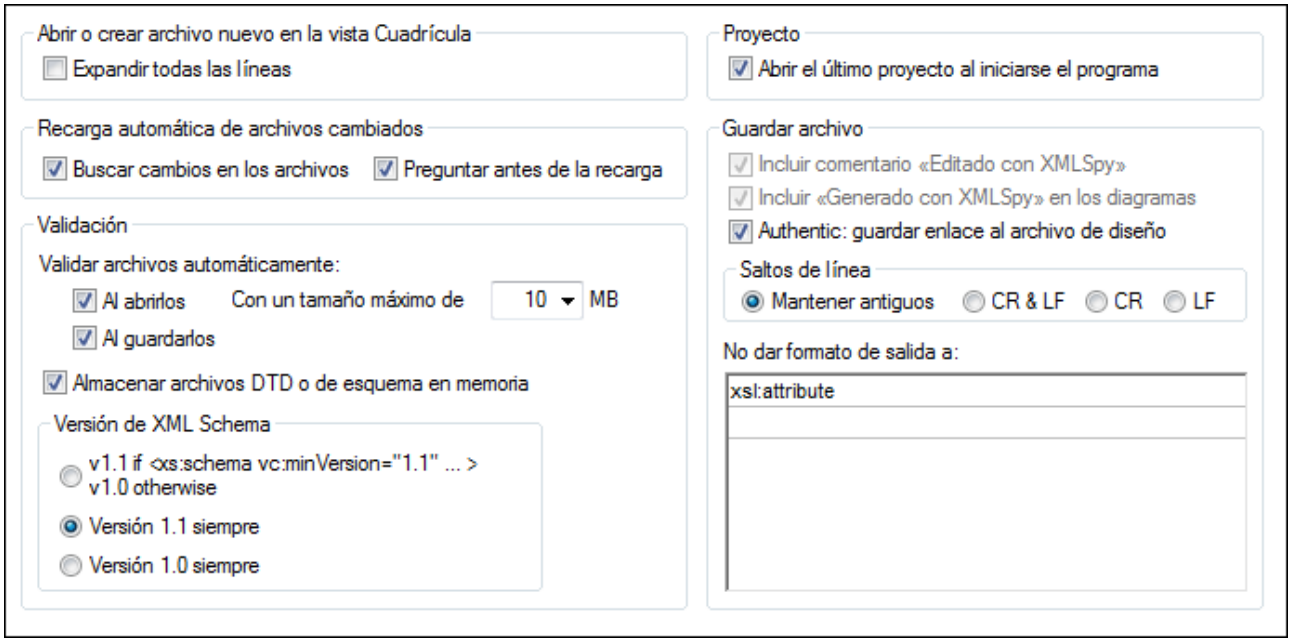


En los apartados siguientes se describen todas las secciones del cuadro de diálogo "Opciones".

28.17.15.1 Archivo

La sección ***Archivo*** define cómo se abren y guardan los documentos en XMLSpy. La sección *Codificación* 1589 incluye opciones de configuración relacionadas.

© 2016-2022 Altova GmbH



Copia de seguridad automática

Si esta opción está activada, la aplicación guarda automáticamente una copia de seguridad de los archivos que edite. Puede seleccionar en el cuadro combinado con qué frecuencia se guarda la copia de seguridad de entre 5 y 60 segundos o introducir un valor de hasta 300 segundos. Para más información consulte la sección *Copia de seguridad automática*[143].

Recarga automática de archivos cambiados

Si está trabajando en un entorno de varios usuarios o si trabaja con archivos que se generan de forma dinámica en un servidor, puede vigilar si se realizan cambios en los archivos que están abiertos en la interfaz. Cada vez que XMLSpy detecte un cambio en un documento de los que están abiertos, la aplicación le pregunta si desea volver a cargar el archivo en el que se realizaron los cambios.

Validación automática

Si usa archivos DTD o esquemas para definir la estructura de sus documentos XML, puede comprobar automáticamente la validez del documento:

- Cada vez que lo abra, si el archivo no supera el límite de MB que haya establecido
- Cada vez que lo guarde
- Mientras edita el archivo.Si selecciona esta opción, el documento se validará conforme teclee tanto en la Vista Texto[145] como en la Vista Cuadrícula JSON[689].Para más información consulte Validación XML en la Vista Texto[335].

Si el documento no es válido, aparece un mensaje de error. Si es válido no aparece ningún mensaje y la operación de lleva a cabo sin ninguna notificación.

Proyecto

Cuando inicie XMLSpy, puede abrir automáticamente el último proyecto que utilizó.

Guardar archivo

Cuando guarda un documento XML, XMLSpy incluye un breve comentario (`<!-- Edited with XMLSpy http://www.altova.com -->`) casi al principio del archivo. Esta opción solamente se puede deshabilitar si el usuario compró una licencia del producto y afecta a los archivos que se editan o guardan en las vistas Cuadrícula y Esquema.

Cuando guarda un diagrama del modelo de contenido (**Diseño de esquemas | Generar documentación**), XMLSpy incluye el logotipo de XMLSpy en la documentación. Esta opción solamente se puede deshabilitar si el usuario compró una licencia del producto.

Si el archivo XML tiene un archivo SPS asociado y marca la casilla *Authentic: guardar enlace al archivo de diseño,* el enlace al archivo SPS se guardará en el archivo XML.

Saltos de línea

Cuando abra un archivo, la codificación de caracteres para los saltos de línea del archivo se mantiene si selecciona la opción *Mantener antiguos*. Si lo prefiere, puede codificar los saltos de línea con una de estas tres codificaciones: **CR&LF** (para PC), **CR** (para MacOS) o **LF** (para Unix).

Modo de salida

Aquí puede definir qué ocurre con los archivos abiertos al cerrar XMLSpy. Hay varias opciones:

- *Ver solicitud de guardar los datos modificados:* si un archivo abierto contiene cambios sin guardar, el programa le preguntará si quiere guardarlos. Según la respuesta que dé, el archivo se guarda o no y después la aplicación se cierra.
- *Ver solicitud de guardar los datos modificados. Abrir los últimos archivos la próxima vez que se inicie la aplicación:* aparece el cuadro de diálogo "Guardar" para los archivos abiertos que contienen cambios sin guardar. Puede elegir si quiere o no guardar esos cambios. Cuando vuelva a abrir la aplicación, los archivos que estaban abiertos al cerrarla se volverán a abrir. (Si no se guardan los cambios realizados antes de salir de la aplicación, estos se perderán.)
- *Conservar los cambios pero sin guardar. Abrir los últimos archivos modificados la próxima vez que se inicie la aplicación:* la aplicación se cierra directamente sin guardar ningún cambio. Al volver a abrirla, todos los archivos que estaban abiertos se vuelven abrir y contienen los cambios no guardados. Básicamente es como si retomara el trabajo donde lo dejó.

Al salir de la aplicación por primera vez, verá el cuadro de diálogo del modo de salida, donde podrá escoger el comportamiento que mejor le convenga. En adelante puede cambiar estas opciones en la sección Archivo del cuadro de diálogo "Opciones".

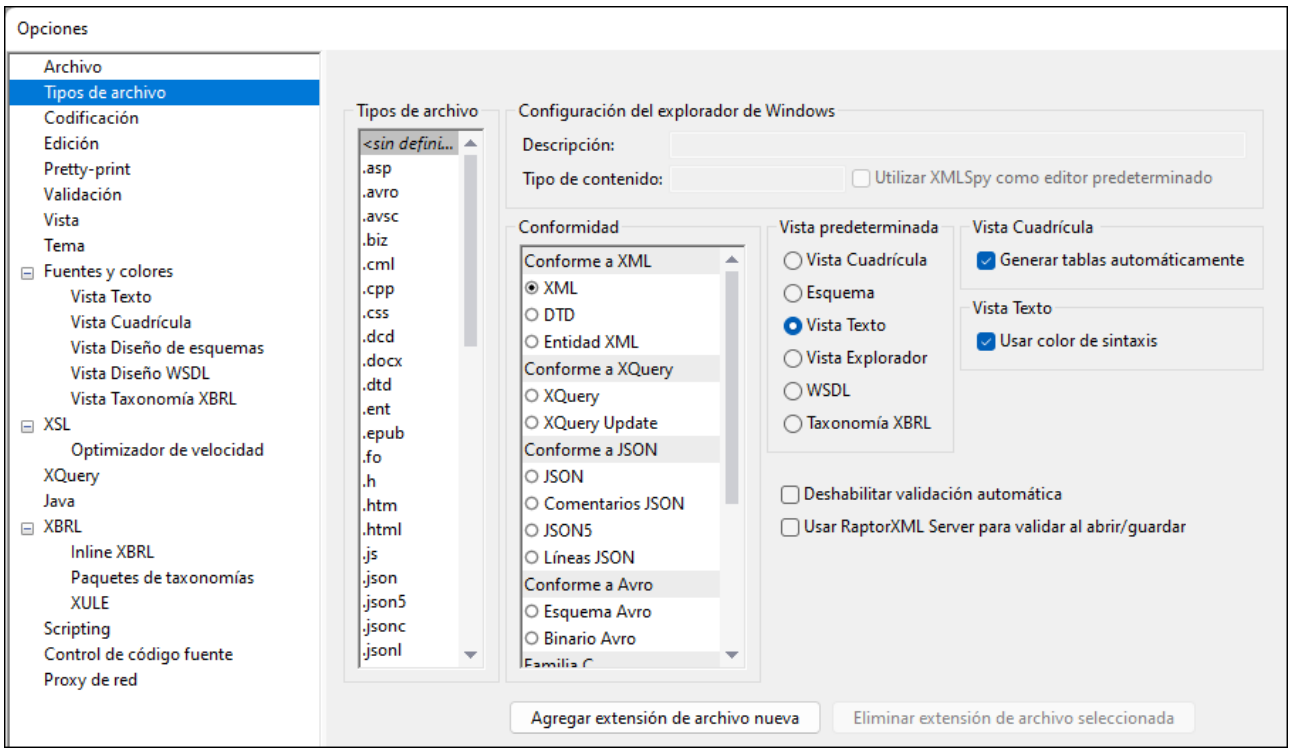
Guardar y salir

Cuando termine de modificar las opciones de configuración, haga clic en **Aceptar** para terminar.

© 2016-2022 Altova GmbH

28.17.15.2 Tipos de archivo

En la sección ***Tipos de archivo*** puede adaptar a sus requisitos el comportamiento de XMLSpy dependiendo del tipo de archivo.



Elija un tipo de archivo en el panel Tipos de archivo y después personalice las funciones para ese tipo de archivo concreto tal y como se explica a continuación. Observe que hay dos entradas especiales en la lista Tipos de archivo:

- *<predeterminado>* se puede usar para indicar qué tratamiento quiere dar a los archivos que tengan una extensión que no esté incluida en la lista de tipos de archivos.
- *<ninguno>* se puede usar para indicar qué tratamiento quiere dar a los archivos que no tengan ninguna extensión.

Configuración del explorador de Windows

Puede definir la descripción del tipo de archivo y el tipo de contenido MIME utilizado por Windows Explorer. También puede elegir si XMLSpy es el editor predeterminado para los archivos de este tipo.

Conformidad

XMLSpy ofrece funciones de edición inteligente y otras funciones para distintos tipos de archivo. XMLSpy determina las características propias para cada tipo de archivo dependiendo de la configuración de conformidad elegida en este grupo de opciones. Por ejemplo, en la imagen anterior, puede ver que los archivos con extensión `.xqu` se configuraron como archivos conformes a XQuery Update. Esto significa que XMLSpy abrirá todos los archivos `.xqu` con funciones de edición compatibles con XQuery Update. En XMLSpy puede elegir entre estas opciones de conformidad XML, XQuery 505, ZIP 943, JSON 665, Avro 767 y otros formatos. Además,

la opción de conformidad XML incluye varias subcategorías: XML, DTD y entidad XML. Los archivos conforme a JSON también pueden ser archivos JSON planos o Avro Schema. La opción *Conforma a Avro* invoca todas las funciones compatibles con archivos binarios Avro. La configuración predeterminada de XMLSpy abarca un gran número de tipos de archivo. Por eso no recomendamos cambiar esta configuración predeterminada a no ser que sea para añadir tipos de archivo nuevos o si necesita asignar otra clase de conformidad a un tipo de archivo determinado.

Vista predeterminada

En este panel puede definir la vista predeterminada de cada tipo de archivo. Si un tipo de archivo conforme con determinada especificación solamente se puede ver en una vista, entonces dicha vista estará seleccionada por defecto y la selección de vista se deshabilita. Asimismo, los documentos conformes a Avro (binarios Avro) solamente se pueden ver en la vista Cuadrícula.

Vista Cuadrícula

Aquí puede elegir si la vista Cuadrícula genera tablas automáticamente o no.

Vista Texto

Aquí puede elegir si se usa color de sintaxis o no.

Deshabilitar validación automática

Si marca esta casilla, se deshabilita la validación automática del tipo de archivo seleccionado. Por lo general, la validación automática se produce al abrir o guardar el archivo o cuando se cambia de vista.

Usar RaptorXML Server para validar al abrir/guardar

Con esta casilla puede indicar si se debe usar RaptorXML Server para validar archivos del tipo seleccionado cuando el archivo se abre y guarda. Para que esta característica funcione es necesario agregar un servidor RaptorXML y configurarlo[1559].

Agregar extensión de archivo nueva

Añade un tipo de archivo nuevo a la lista *Tipos de archivo*. Después debe definir la configuración de este tipo de archivo nuevo usando las demás opciones de esta pestaña.

Eliminar extensión de archivo seleccionada

Elimina el tipo de archivo seleccionado y todas sus opciones de configuración.

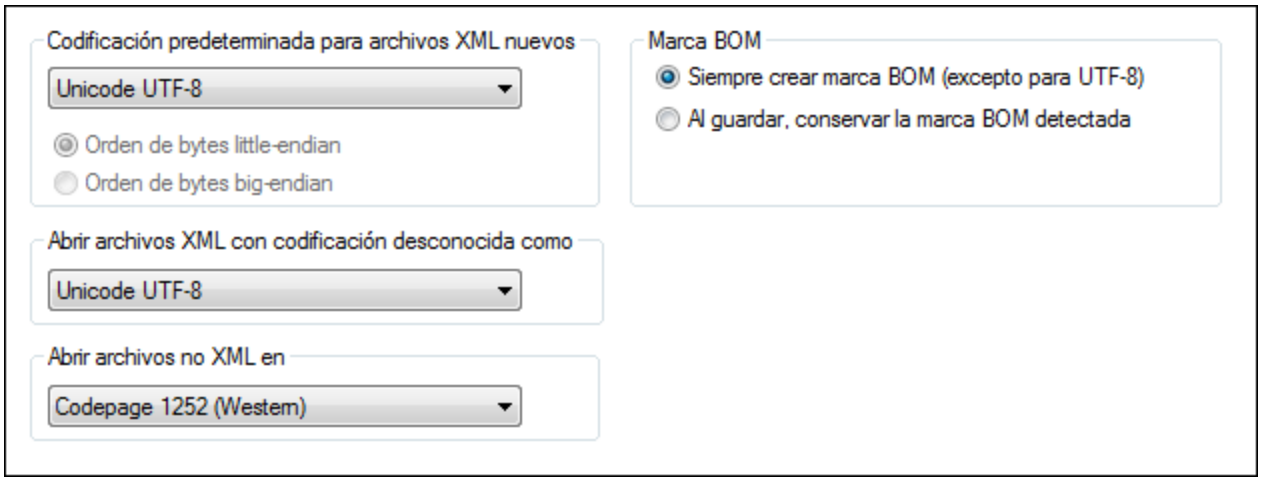
Guardar y salir

Cuando la configuración esté lista, haga clic en **Aceptar** para terminar.

© 2016-2022 Altova GmbH

28.17.15.3 Codificación

En la sección ***Codificación*** puede configurar las opciones de codificación de archivos.



Codificación predeterminada para archivos XML nuevos

En esta lista desplegable puede elegir la codificación predeterminada para los archivos XML nuevos. Cada documento nuevo se crea con una declaración XML que contiene el valor de codificación elegido aquí. Si selecciona una codificación de dos o cuatro bytes como codificación predeterminada (es decir, UTF-16, UCS-2 o UCS-4) también puede elegir entre el orden de bytes little-endian o big-endian.

La codificación de los archivos XML ya existentes se conserva y solamente se puede cambiar con el comando **Archivo | Codificación**[1268].

Abrir archivos XML con codificación desconocida como

Si no se puede determinar la codificación de un archivo XML o si el documento XML no especifica ninguna codificación, el archivo se abre con la codificación que elija en este cuadro combinado.

Abrir archivos no XML en

Los archivos no XML nuevos y los ya existentes se abren con la codificación que elija en este cuadro combinado. Puede cambiar la codificación del documento con el comando **Archivo | Codificación**[1268].

Marca BOM (orden de bytes)

Los documentos que tienen una codificación de caracteres de dos o cuatro bytes se pueden guardar:

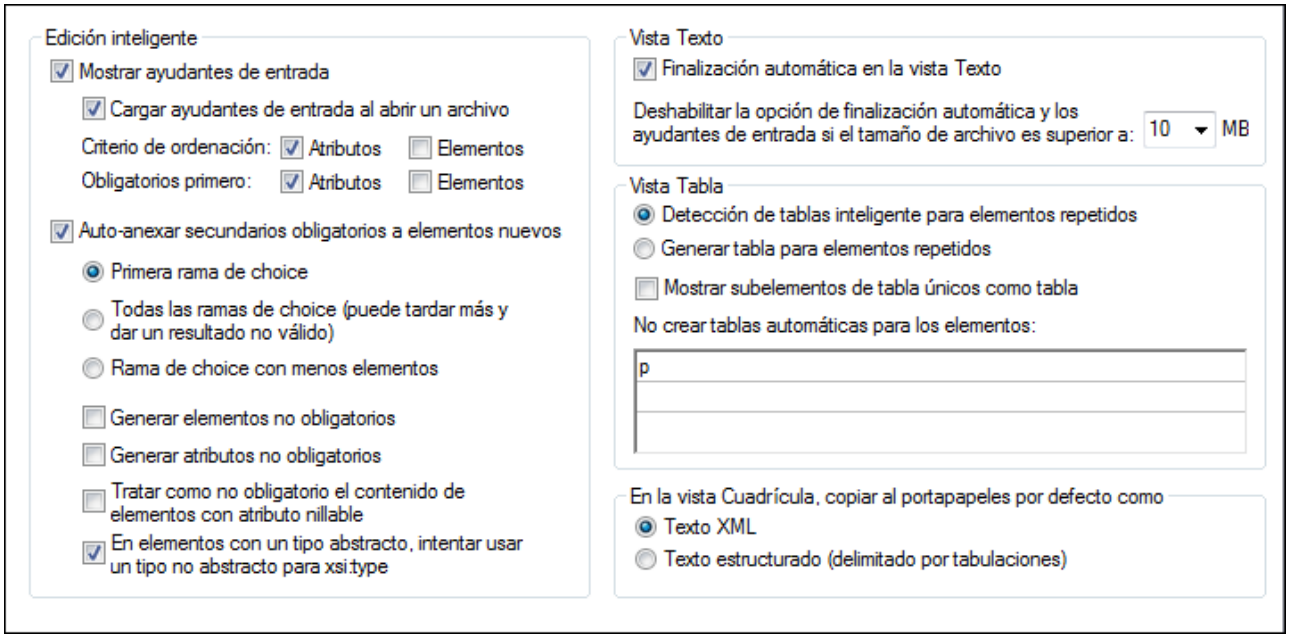
- con un orden de bytes little-endian y una marca BOM little-endian (opción *Siempre crear marca BOM (excepto para UTF-8)*) o
- con el orden de bytes detectado y la marca BOM detectada (opción *Al guardar, conservar la marca BOM detectada*).

Guardar y salir

Cuando termine de modificar las opciones de configuración, haga clic en **Aceptar** para terminar.

28.17.15.4 Edición

En la sección ***Edición*** puede configurar las opciones de edición de XMLSpy.



Ayudantes de entrada

Mientras edita un documento, XMLSpy puede ofrecerle opciones de edición inteligente a través de los denominados ayudantes de entrada. En este panel también puede personalizar algunos aspectos del comportamiento de estos ayudantes, como el orden en que aparecen los componentes de los ayudantes. Las opciones elegidas en esta pestaña se aplicarán cuando se edite el tipo de archivo pertinente. Por ejemplo, la opción de cargar los ayudantes al abrir un archivo y de ordenarlo por atributos no afecta a los documentos DTD ni XQuery.

Crear una estructura XML a partir de un esquema XML

Cuando se crea un documento XML nuevo y este está basado en un esquema XML, el documento se genera con una estructura derivada de las definiciones de ese esquema. Las opciones que describimos a continuación determinan ciertos aspectos ambiguos relacionados con la creación de esta estructura.

© 2016-2022 Altova GmbH

Auto-anexar secundarios obligatorios
Los elementos secundarios de los grupos `choice` del esquema se anexan automáticamente en base a la configuración de este panel. Puede elegir si se generan (i) la primera rama (elemento) del grupo `choice`, (ii) todas las ramas o (iii) la rama con el menor número de elementos descendientes. Tenga en cuenta que la opción Todas las ramas podría generar un documento no válido, ya que solo se permite una rama del grupo `choice`.

Nodos y elementos de tipo abstracto no obligatorios
Para añadir elementos o atributos no obligatorios seleccione la opción correspondiente. Si no lo hace se añadirán solamente los nodos obligatorios. También puede (i) definir el contenido de los elementos nillable (que puede tomar valores nulos) como no obligatorio y (ii) usar un tipo no abstracto como el `xsi:type` de un elemento de un tipo abstracto.

Vista Texto

Si marca la casilla *Finalización automática*, XMLSpy añade automáticamente componentes estructurales inequívocos. Por ejemplo, cuando introduzca el corchete angular de cierre de la etiqueta inicial de un elemento, se añade automáticamente la etiqueta final del elemento.

En la vista Texto puede deshabilitar la función de finalización automática y los ayudantes de entrada si el archivo tiene un tamaño mayor al indicado en este panel (cuadro combinado *Deshabilitar la opción de finalización automática y los ayudantes...*). Esto es muy útil si desea acelerar la edición de archivos grandes y no necesita la función de finalización automática ni los ayudantes de entrada. Si el tamaño del archivo es mayor al indicado en este cuadro combinado, el menú contextual de la vista Texto contiene un conmutador para activar/desactivar el uso de la finalización automática y de los ayudantes de entrada. Es decir, puede volver a activar estas funciones siempre que quiera durante la edición (si los archivos superan el tamaño máximo indicado en el cuadro combinado). Si el valor indicado en este cuadro combinado es menor al tamaño del archivo que está abierto, las posiciones de los mensajes de error no se corresponden al 100% con las posiciones en la vista Texto.

Guardar y salir

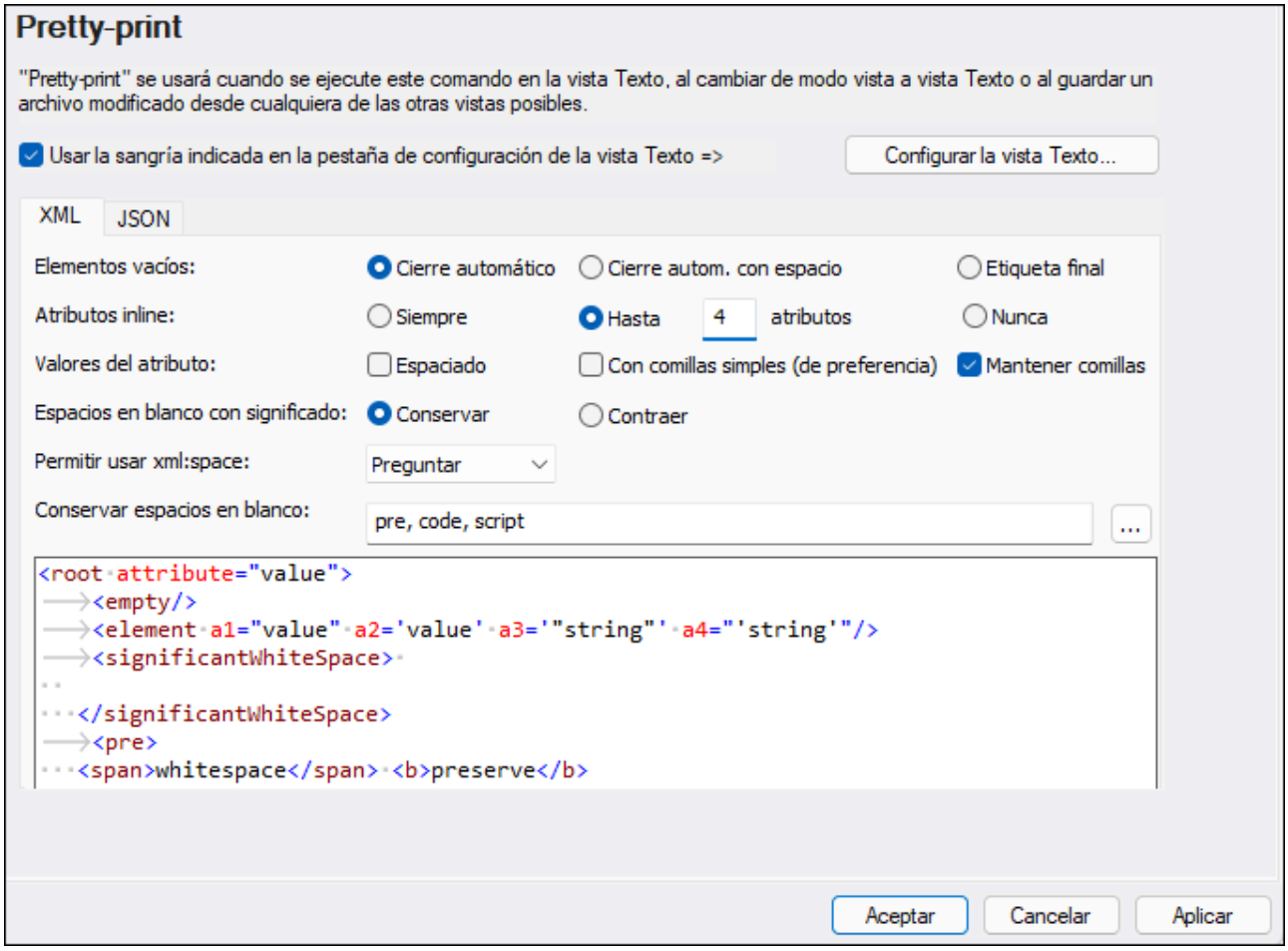
Para terminar haga clic en **Aceptar**.

28.17.15.5 Pretty Printing

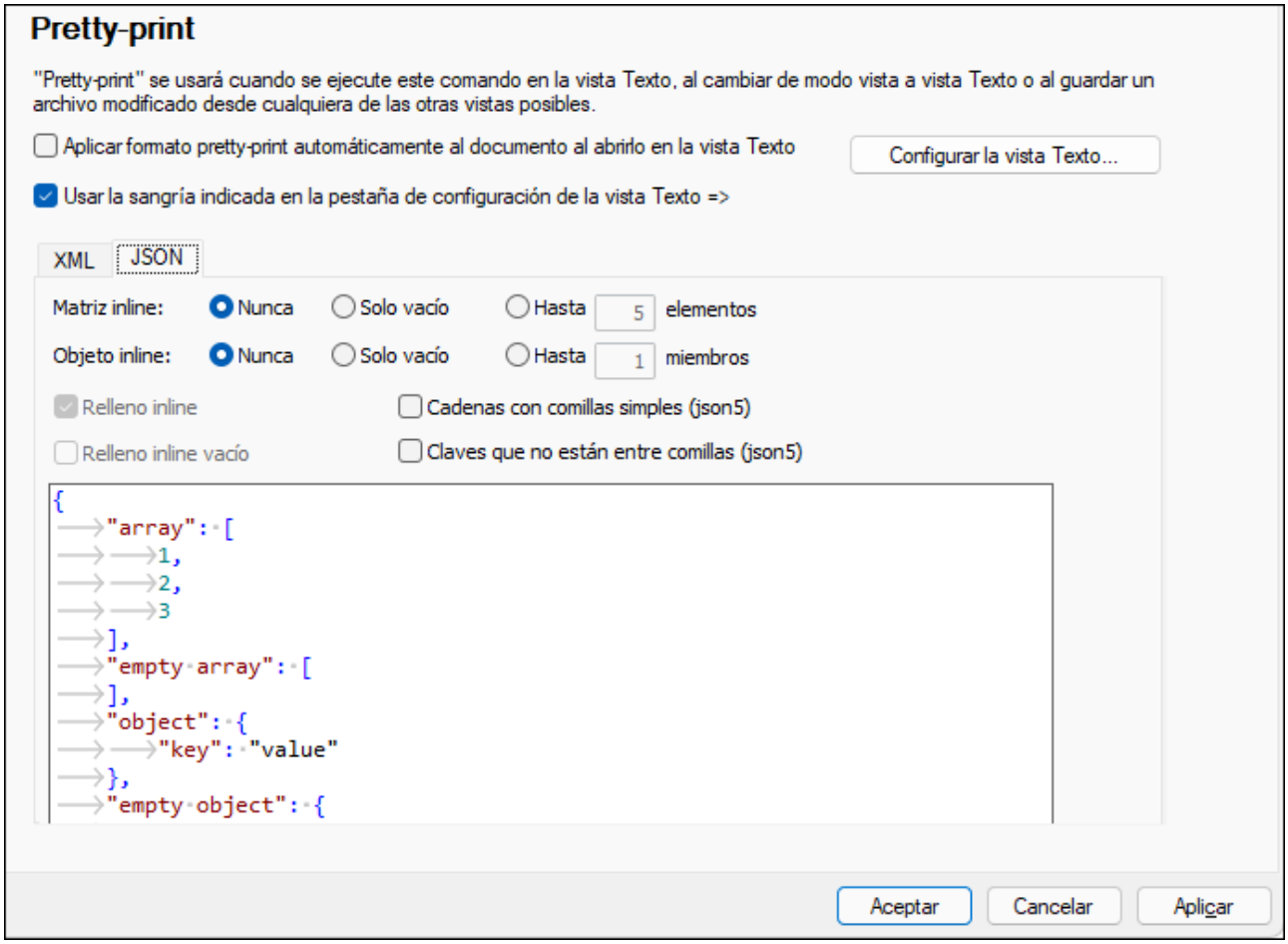
La sección **Pretty Print** (*imágenes siguientes*) permite indicar cómo aparece el texto en la Vista Texto. Las definiciones de esta sección se agrupan por categorías:

- Opciones XML (seleccione la pestaña XML para acceder a estas opciones; *imagen de la izquierda*)
- Opciones JSON (seleccione la pestaña JSON para acceder a estas opciones; *imagen de la derecha*)
- Opciones de la Vista Texto (haga clic en el botón correspondiente)

La casilla *Usar la sangría indicada* activa o desactiva el formato pretty-printing[146]. Si marca la casilla *Aplicar formato pretty-print automáticamente* este formato se aplicará automáticamente al cargar el documento.



 © 2016-2022 Altova GmbH



Opciones XML

Estas son las opciones disponibles:

- Cómo se escriben y aparecen en el documento los elementos vacíos: con una etiqueta (*cierre automático*) o dos (*etiqueta final*).
- Si los atributos aparecen en la misma línea que su elemento de nivel superior (inline) o no. Los atributos aparecen inline si selecciona *Siempre* o si selecciona *Hasta X atributos* y el número de atributos no es mayor que X. Los atributos aparecen en líneas nuevas si selecciona *Nunca* o si selecciona *Hasta X atributos* y el número de atributos excede X.
- Cómo se escriben los valores de los atributos: (i) con espacios a ambos lados del signo igual o sin ellos, (ii) si los valores usan comillas simples o dobles, (iii) si las comillas del texto de origen se conservan tal y como las escribió o si otras opciones las sobrescriben (como *Comillas simples de preferencia*); tenga en cuenta que si selecciona *Mantener comillas*, estas solo se aplican tanto como sea posible sin invalidar el documento.
- Definir si los espacios en blanco se conservan tal y como están o si se contraen. Los caracteres de espacio en blanco son: espacio, tabulador, salto de línea y retorno de línea. Consulte *Espacios en blanco*[337] para más detalles.
- Si los elementos de un documento contienen el par atributo-valor **`xml:space="preseve"`** puede indicar cómo se considera este par al aplicar pretty-print con la opción *Permitir usar xml:space*. La opción

Siempre indica que se debe seguir la intención del atributo al aplicar pretty-print a cualquier documento: los espacios en blanco de los elementos correspondientes se conserva y no se les aplica pretty-print. La opción *Nunca* indica que se debe ignorar el atributo *xml:space* y aplicar pretty-print a los elementos correspondientes. Si elige la opción *Preguntar*, XMLSpy le preguntará qué debe hacer cada vez que vaya a aplicar pretty-print a un documento que contenga **`xml:space="preseve"`**.

- Definir qué elementos conservan espacios en blanco
- Si se usa la sangría definida en el cuadro de diálogo Configurar la vista Texto[1489] o si no se usa ninguna (esto se indica con la casilla de encima del panel)

Al seleccionar una opción, el resultado aparece en el panel de vista previa en la parte inferior del cuadro de diálogo, lo que permite controlar qué cambios quiere aplicar antes de confirmarlos haciendo clic en **Aceptar**.

Opciones JSON

Al seleccionar cualquiera de estas opciones el resultado aparece inmediatamente en el panel de vista previa en la parte inferior del cuadro de diálogo.

- *Matriz inline:* presenta los elementos de una matriz en una sola línea (inline). De lo contrario los elementos pueden aparecer en líneas distintas. Puede elegir no aplicar nunca el formato inline a las matrices o aplicarlo solamente a matrices vacías o a matrices hasta un tamaño en concreto.
- *Objeto Inline:* presenta las propiedades de un objeto en una sola línea (inline). De lo contrario las propiedades pueden aparecer en líneas distintas. Puede elegir no aplicar nunca el formato inline a los objetos o aplicarlo solamente a objetos vacíos u objetos con cierto número de propiedades.
- *Relleno inline:* si activa esta opción se añade espacio entre los elementos de las matrices inline que no estén vacías y los objetos inline que no estén vacíos. Esta opción solo se activa si primero se activan una de las dos anteriores.
- *Relleno inline vacío:* esta opción añade espacio dentro de los delimitadores de las matrices inline vacías y de los objetos inline vacíos. La opción se habilita si define una matriz o un objeto como inline (estén vacíos o no).
- *Cadenas entre comillas (JSON5):* si se activa, esta opción convierte todas las comillas de los documentos JSON5 en comillas simples.
- *Cadenas que no están entre comillas:* si se activa, esta opción elimina las comillas de todas las claves (de los pares `clave:valor`) de los documentos JSON5.

Puede usar la casilla de encima del panel para indicar si se debe usar la sangría indicada en el cuadro de diálogo Configurar la vista Texto[1489] o si no se usa ninguna.

Opciones de la vista Texto

Haga clic para abrir el cuadro de diálogo Opciones de la vista Texto, donde puede habilitar propiedades de la vista Texto como la sangría, los márgenes de marcadores o el resaltado automático. También puede acceder al cuadro de diálogo Configurar la vista Texto desde el comando de menú **Vista | Configurar la vista Texto**[1489]. El cuadro de diálogo se explica en el apartado al que se accede si sigue el enlace.

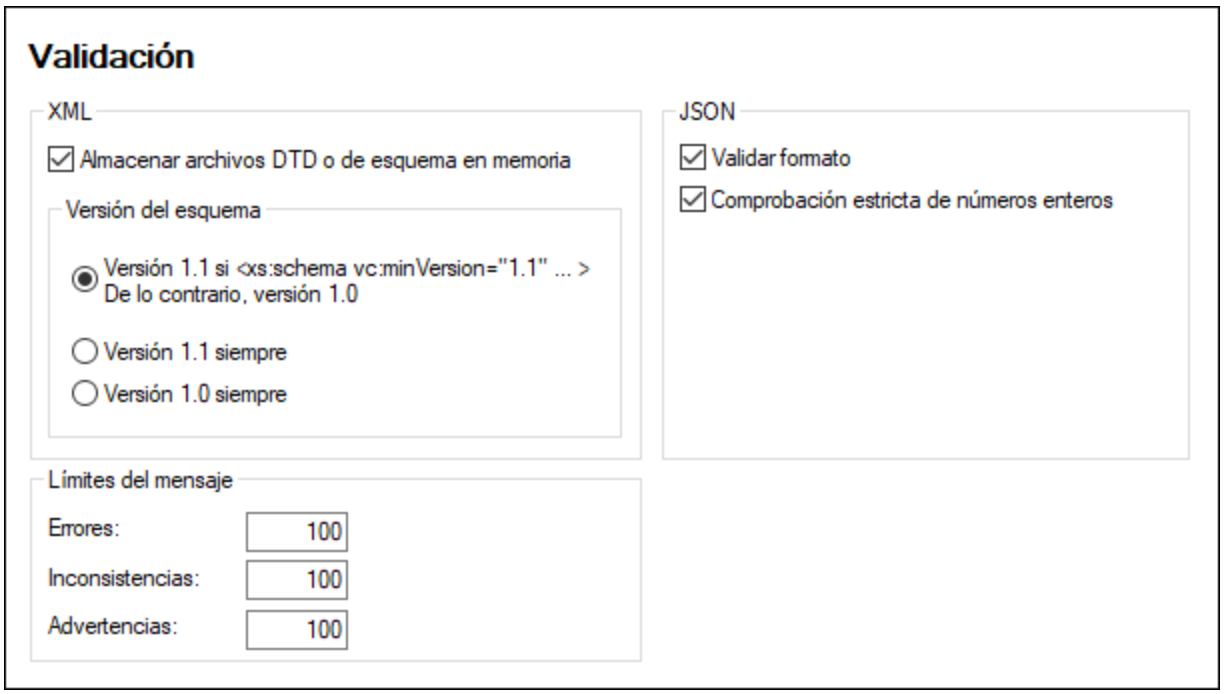
Guardar y salir

Cuando la configuración esté lista, haga clic en **Aceptar** para terminar.

© 2016-2022 Altova GmbH

28.17.15.6 Validación

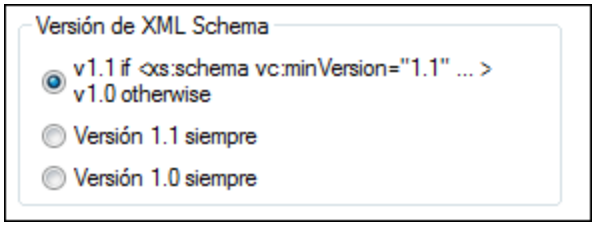
En la opción **Validación** puede configurar las opciones de validación de documentos XML y JSON.



XML

XMLSpy puede almacenar en su memoria caché archivos DTD y XML Schema para no tener que cargarlos continuamente (por ejemplo, si el esquema no es local sino que se accede a él con un URL). Sin embargo, recuerde que si usa las versiones de la memoria caché de los esquemas y hace cambios en un esquema, estos no se reflejan de inmediato en la validación, sino que necesita volver a cargar el archivo XML o reiniciar XMLSpy.

Versión del esquema
El modo XSD que se habilita en la vista Esquema depende tanto de (i) la presencia/ausencia del atributo `/xs:schema/vc:minVersion` del documento XSD (y, si está presente, de su valor), como de (ii) la opción seleccionada en la sección *Archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**) (*imagen siguiente*).



En definitiva pueden darse todas las situaciones que aparecen en la siguiente tabla. Con *Versión de XML Schema* nos referimos a la opción seleccionada en el panel de opciones Versión de XML Schema. La segunda columna (atributo `vc:minVersion`) corresponde al valor del atributo `xs:schema/@vc:minVersion` del esquema XML. Para más información consulte el apartado Vistas de edición | Vista Esquema | Modo XSD[173].

Versión de XML Schema	Atributo `vc:minVersion`	Modo XSD
Versión 1.0 siempre	Ausente (o presente con cualquier valor)	1.0
Versión 1.1 siempre	Ausente (o presente con cualquier valor)	1.1
Valor de @vc:minVersion	El atributo tiene el valor `1.1`	1.1
Valor de @vc:minVersion	`Falta el atributo o está presente y tiene cualquier valor menos 1.1`	1.0

Límite de los mensajes

Estas opciones permiten establecer distintos límites para los errores, las inconsistencias XBRL y las advertencias que aparecen. El número predeterminado para cada una de estas categorías es 100. Puede cambiarlo al número que quiera.

JSON

Estas son las opciones de validación disponibles para la validación de documentos JSON:

- *Validar formato:* Se valida el formato de los tipos de cadena[738] en documentos de instancia JSON.
- *Comprobación estricta de números enteros:* Hay dos tipos numéricos en JSON: `número` y `número entero`. Esta opción comprueba que los números enteros son de tipo número entero (y no, por ejemplo, números de coma flotante, como `7.0`, firmados, como `+7`, o cadenas de texto, como `"7"`).

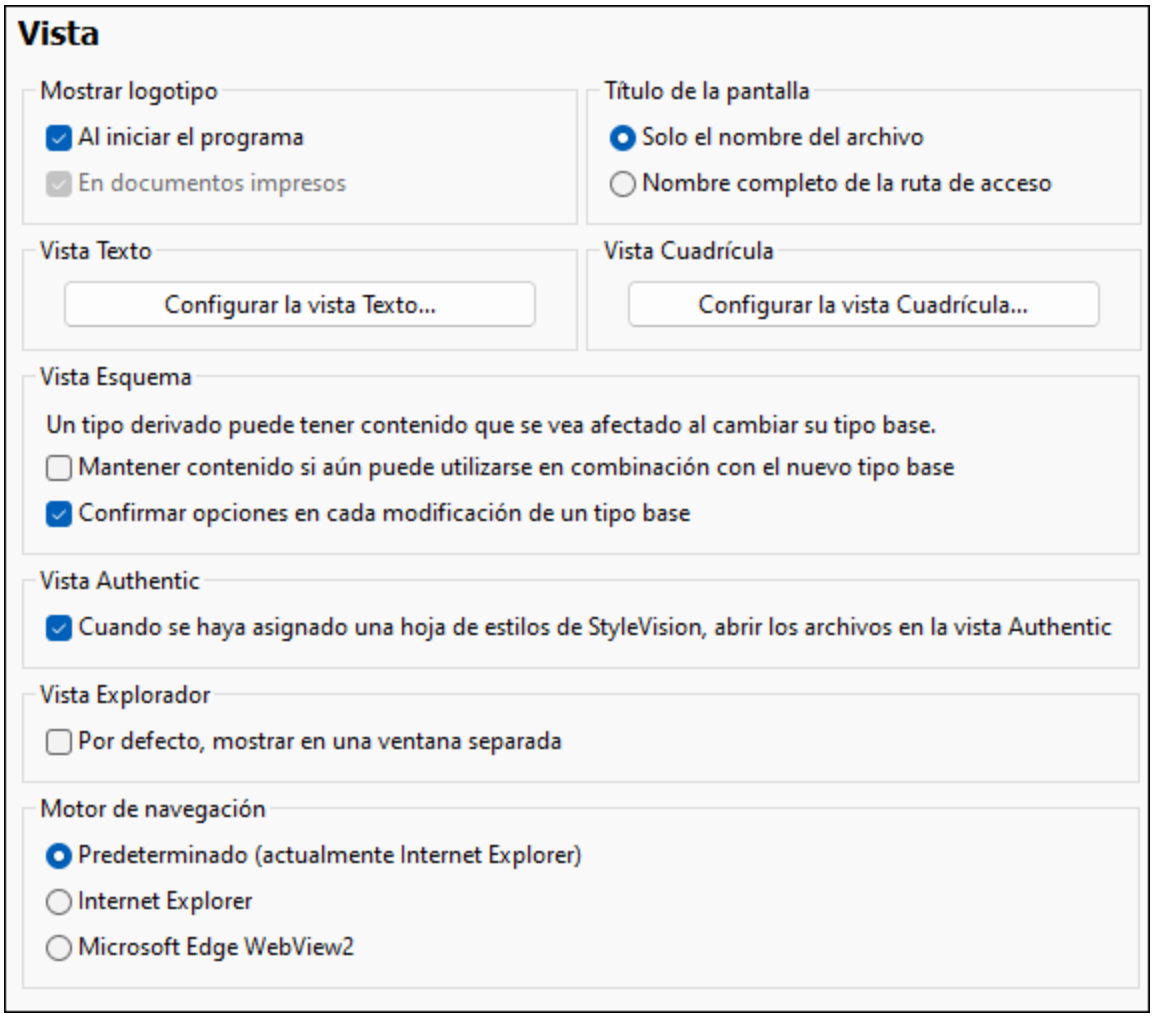
Guardar y salir

Cuando la configuración esté lista, haga clic en **Aceptar** para terminar.

© 2016-2022 Altova GmbH

28.17.15.7 Vista

En la sección ***Vista*** puede personalizar la presentación de los documentos XML en XMLSpy.



Mostrar logotipo

Si desea que la aplicación se inicie más rápido, puede desactivar la pantalla de presentación que aparece al iniciarse el programa. Además, si compró una licencia del producto, tiene la opción de no incluir el logotipo del programa, el aviso de copyright ni la información de registro en los documentos que imprima desde XMLSpy.

Título de la pantalla

El título de la pantalla de cada ventana de documento puede llevar el nombre del archivo solamente o el nombre completo de la ruta de acceso del archivo.

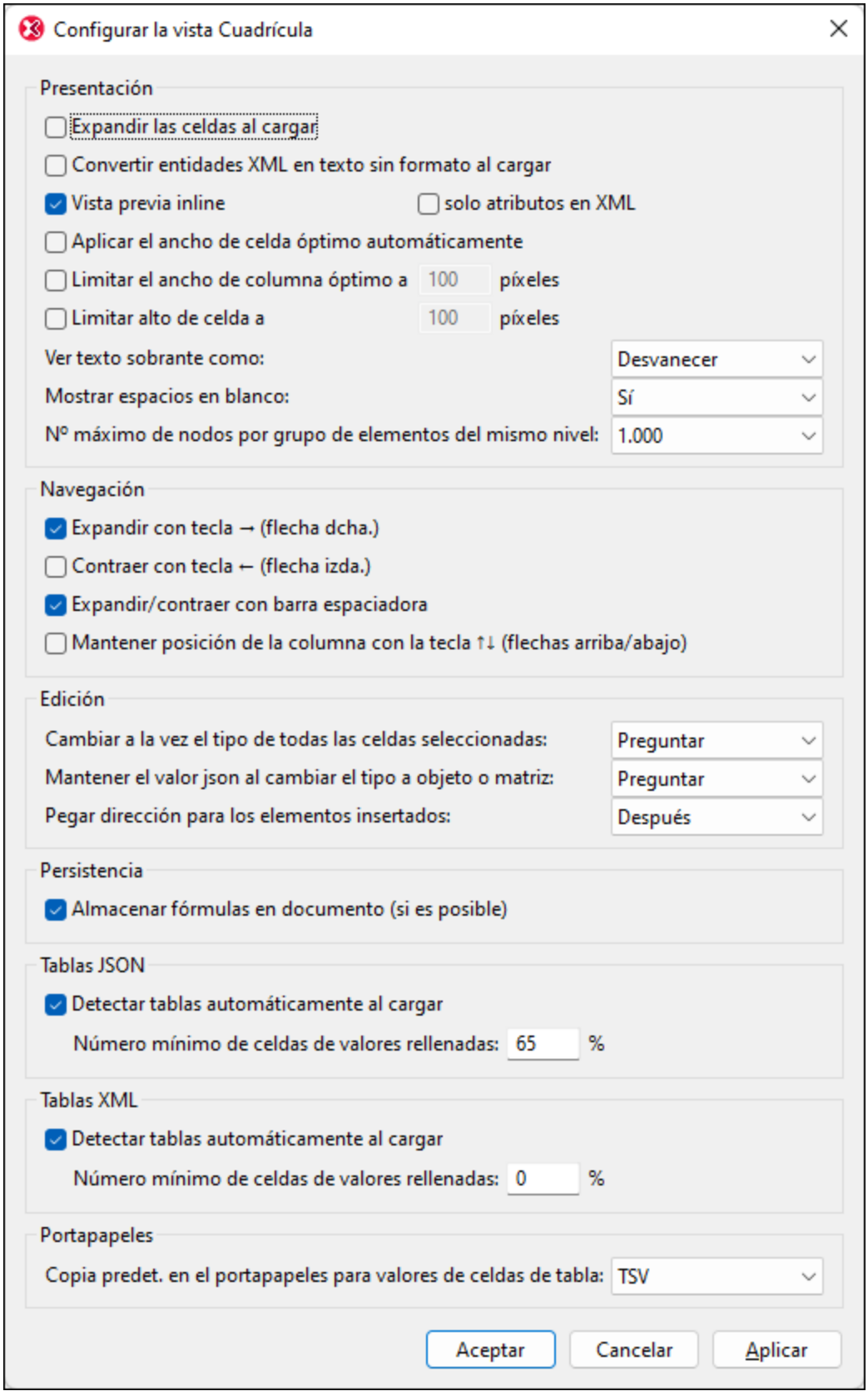
Configurar la vista Texto

Haga clic en **Configurar la vista Texto** para abrir el cuadro de diálogo correspondiente, donde puede habilitar propiedades de la vista Texto como la sangría, los márgenes de marcadores o el resaltado automático. También puede acceder al cuadro de diálogo Configurar la vista Texto desde el comando de menú **Vista | Configurar la vista Texto** 1489. El cuadro de diálogo se explica en el apartado al que se accede si sigue el enlace.

Configurar la vista Cuadrícula

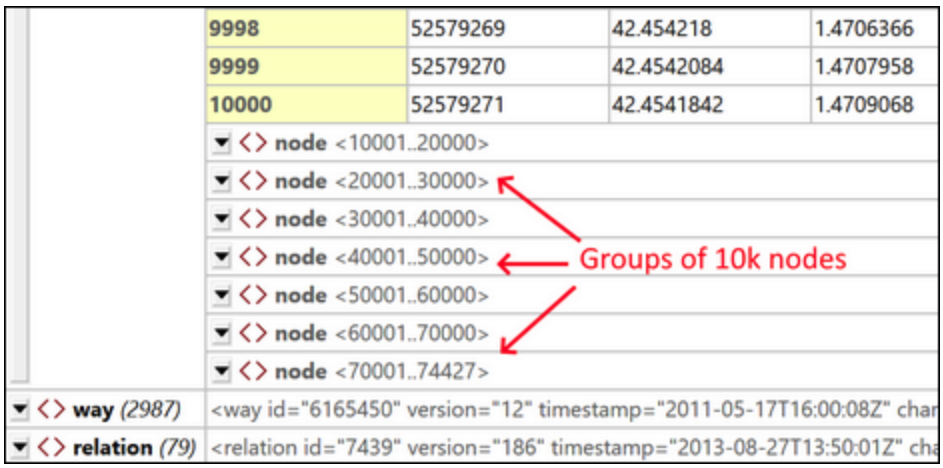
Haga clic en **Configurar la vista Cuadrícula** para abrir el cuadro de diálogo correspondiente. Estas son las opciones que puede configurar en él:

© 2016-2022 Altova GmbH


Configurar la vista Cuadrícula
Presentación
Expandir las celdas al cargar
Convertir entidades XML en texto sin formato al cargar
Vista previa inline
solo atributos en XML
Aplicar el ancho de celda óptimo automáticamente
Limitar el ancho de columna óptimo a 100 píxeles
Limitar alto de celda a 100 píxeles
Ver texto sobrante como: Desvanecer
Mostrar espacios en blanco: Sí
Nº máximo de nodos por grupo de elementos del mismo nivel: 1.000
Navegación
Expandir con tecla → (flecha dcha.)
Contraer con tecla ← (flecha izda.)
Expandir/contraer con barra espaciadora
Mantener posición de la columna con la tecla ↑↓ (flechas arriba/abajo)
Edición
Cambiar a la vez el tipo de todas las celdas seleccionadas: Preguntar
Mantener el valor json al cambiar el tipo a objeto o matriz: Preguntar
Pegar dirección para los elementos insertados: Después
Persistencia
Almacenar fórmulas en documento (si es posible)
Tablas JSON
Detectar tablas automáticamente al cargar
Número mínimo de celdas de valores rellenadas: 65 %
Tablas XML
Detectar tablas automáticamente al cargar
Número mínimo de celdas de valores rellenadas: 0 %
Portapapeles
Copia predet. en el portapapeles para valores de celdas de tabla: TSV
Aceptar
Cancelar
Aplicar

Presentación
Las casillas de la sección Presentación son bastante intuitivas. De todas formas incluimos unas breves aclaraciones a continuación.

- Si no se expanden todas las celdas al cargar los datos es porque el nodo raíz y sus secundarios están contraídos. En este caso necesita expandir cada uno de los nodos conforme va avanzando por el documento.
- El ancho óptimo se puede activar automáticamente (i) si el documento se carga en la vista Cuadrícula y (ii) si cambia un conjunto de elementos de la vista en forma de tabla a la vista Cuadrícula normal.
- Cuando se activa el ancho óptimo se amplía el tamaño de las celdas para que el texto quepa en una sola línea. En este caso pueden ocurrir varias cosas: (i) la tabla se expande y acaba siendo mayor que la ventana principal, por lo que aparece una barra de desplazamiento en la parte inferior de la ventana; (ii) el texto de algunas celdas es más largo que el límite derecho de la celda (o el izquierdo, según qué dirección de escritura se use); el texto sobrante se indica con texto atenuado o puntos suspensivos (*véase el punto siguiente*).
- Si la celda no se ajusta al tamaño del texto y este es mayor que el ancho de la celda, la opción *Ver texto sobrante como* permite escoger si quiere ver el texto sobrante atenuado o como puntos suspensivos.
- Los nodos hermanos pueden aparecer en grupos del mismo nivel de 100, 1.000 o 10.000 nodos (*imagen siguiente).* Esta opción es útil si quiere ahorrar espacio al mostrar los datos en archivos con un gran número de nodos del mismo nivel.



Edición
Las casillas de la sección Edición son bastante intuitivas. No obstante incluimos algunas aclaraciones a continuación.

- Al cambiar el tipo de varias de las celdas seleccionadas puede escoger entre varias opciones que permiten o no que la acción siga su curso: *Siempre*, *Nunca* o *Preguntar* (para usuarios).
- Al cambiar un tipo JSON de un tipo atómico a un objeto o una matriz, el programa también sugiere varias opciones que permiten o no que la acción siga su curso: (i) *Preguntar* (si se debe conservar el valor del tipo atómico como valor de un par **`clave:valor`** secundario sin nombre o si se debe descartar), (ii) *Siempre* (conserva el valor de un par **`clave:valor`** secundario sin nombre) y (iii) *Nunca* (no conserva el valor nunca, es decir, lo descarta).
- La opción *Pegar dirección* determina si la selección del portapapeles se pega sobre o junto a la celda seleccionada.

 © 2016-2022 Altova GmbH

Persistencia
Los filtros y las fórmulas se pueden almacenar en los metadatos de los documentos XML y JSON. Las fórmulas se almacenan como instrucciones de procesamiento en documentos XML y como comentarios en algunos tipos de documento JSON. Estas opciones indican cómo proceder con esos filtros y esas fórmulas almacenados. En el caso de las fórmulas, la terminología "si es posible" hace referencia al hecho de que las fórmulas no se pueden guardar como comentarios en todos los documentos JSJON (solo en documentos JSON5 y JSONC).

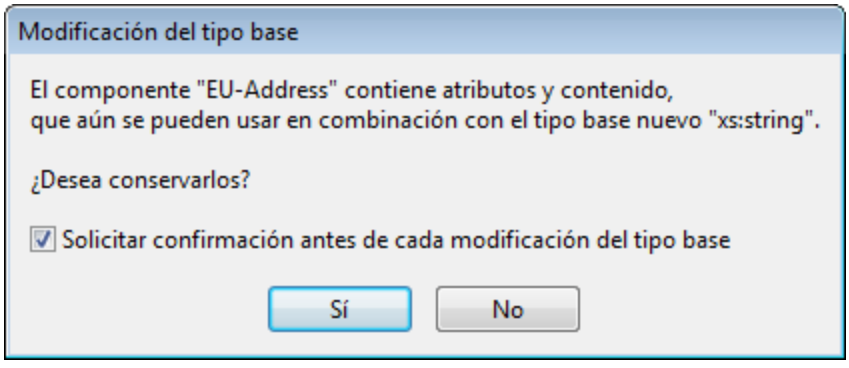
Tablas
Estas son las opciones disponibles:

- Si activa la opción para que las tablas se detecten automáticamente al cargar los datos, entonces puede seleccionar el porcentaje mínimo de celdas rellenadas que tiene que tener una tabla para ser reconocida como tal. Si el número de celdas de tabla rellenadas no sobrepasa este nivel, entonces la estructura se visualiza como una cuadrícula normal con los elementos que se repiten en una lista unos junto a otros.
- También puede decidir si el contenido del portapapeles se debe almacenar como valores separados por tabulador (TSV) o como XML/JSON (en función del tipo de documento). Otra opción muy útil: si quiere pegar una tabla desde el portapapeles en otro documento, esta opción permite escoger si la tabla copiada se almacena como TSV o con marcado. (Para ver la diferencia entre estas dos opciones intente pegar una tabla en un editor de texto después de copiarla en los distintos formatos mencionados).

Vista Esquema

Los tipos de datos XML Schema se pueden derivar de otros tipos de datos. Por ejemplo, el tipo de datos para los elementos `E-Mail` puede derivarse del tipo de datos base `xs:string` (restringiendo el tipo de datos `xs:string` a determinado conjunto de caracteres, por ejemplo). Si más adelante se cambia el tipo de datos base, tiene dos opciones:

- *Mantener el contenido:* si las definiciones utilizadas para definir el tipo derivado se pueden usar con el tipo base nuevo, marque esta casilla y las definiciones se mantendrán automáticamente.
- *Confirmar en cada modificación:* después de cambiar el tipo base aparece un cuadro de diálogo (*imagen siguiente*) que pregunta si desea mantener o no las definiciones anteriores y usarlas con el tipo base nuevo.



Vista Authentic

Si marca esta casilla, los archivos que están basados en un archivo SPS se abren automáticamente en la vista Authentic.

Motor del navegador

El motor del navegador que se utiliza en la vista Authentic y la vista Explorador es actualmente Internet Explorer (IE), por lo que IE es el motor del navegador por defecto para estas dos vistas. Como alternativa, puede utilizar Microsoft Edge Web View 2 como motor para la vista Explorador. Si Edge no está instalado en su equipo, vaya a la página de descarga de WebView2, desde donde puede instalar el Evergreen Bootstrapper. Esto le permitirá utilizar Microsoft Edge WebView2 como motor para la vista Explorador.

Consulte el tema Vista Explorador 279 para obtener más información.

Vista Explorador

Aquí puede elegir si la vista Explorador se abre en una ventana separada, lo cual permite ver la vista de edición y la vista explorador en paralelo.

Guardar y salir

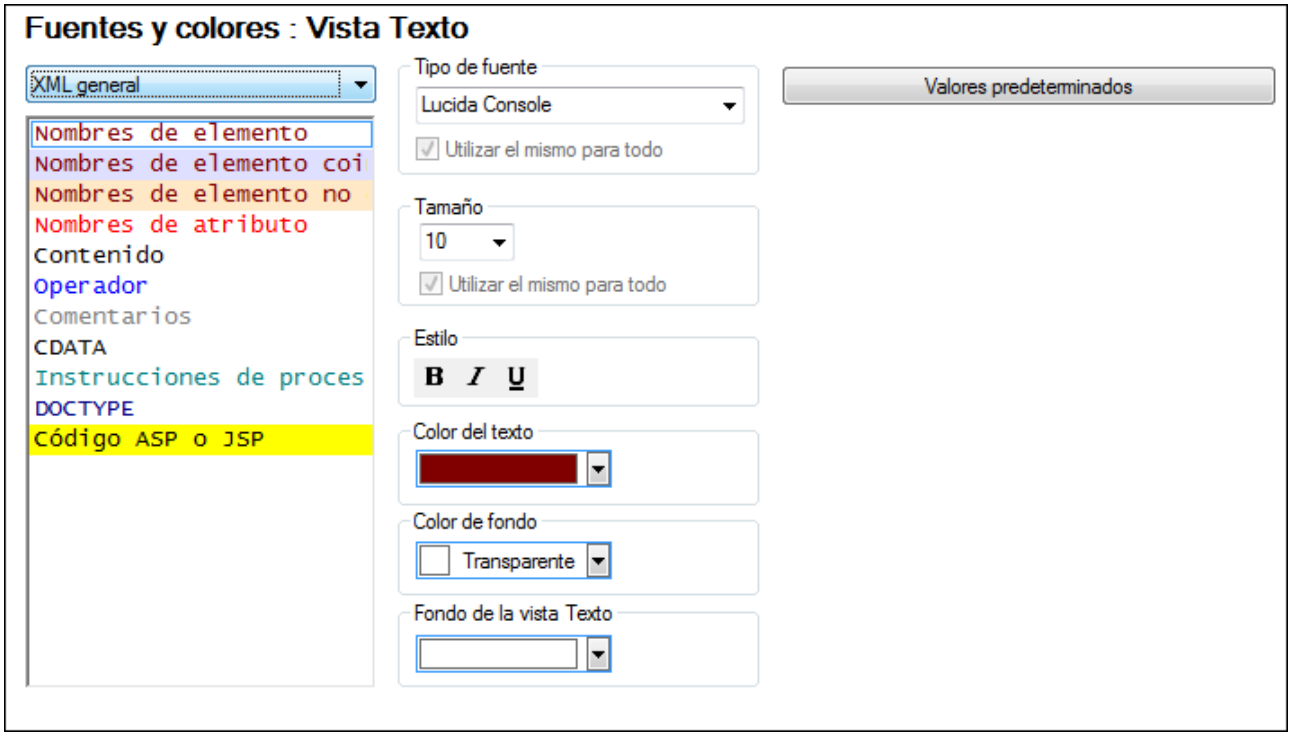
Cuando termine de modificar las opciones de configuración, haga clic en **Aceptar** para terminar.

28.17.15.8 Fuentes y colores

La sección ***Fuentes y colores*** ofrece opciones para personalizar el aspecto de los elementos de texto en cada una de las vistas de XMLSpy.

Nota: las opciones de la sección Fuentes y colores se aplican al tema que esté activo en ese momento. Para modificarlas en otros temas primero tiene que activar ese tema.

© 2016-2022 Altova GmbH



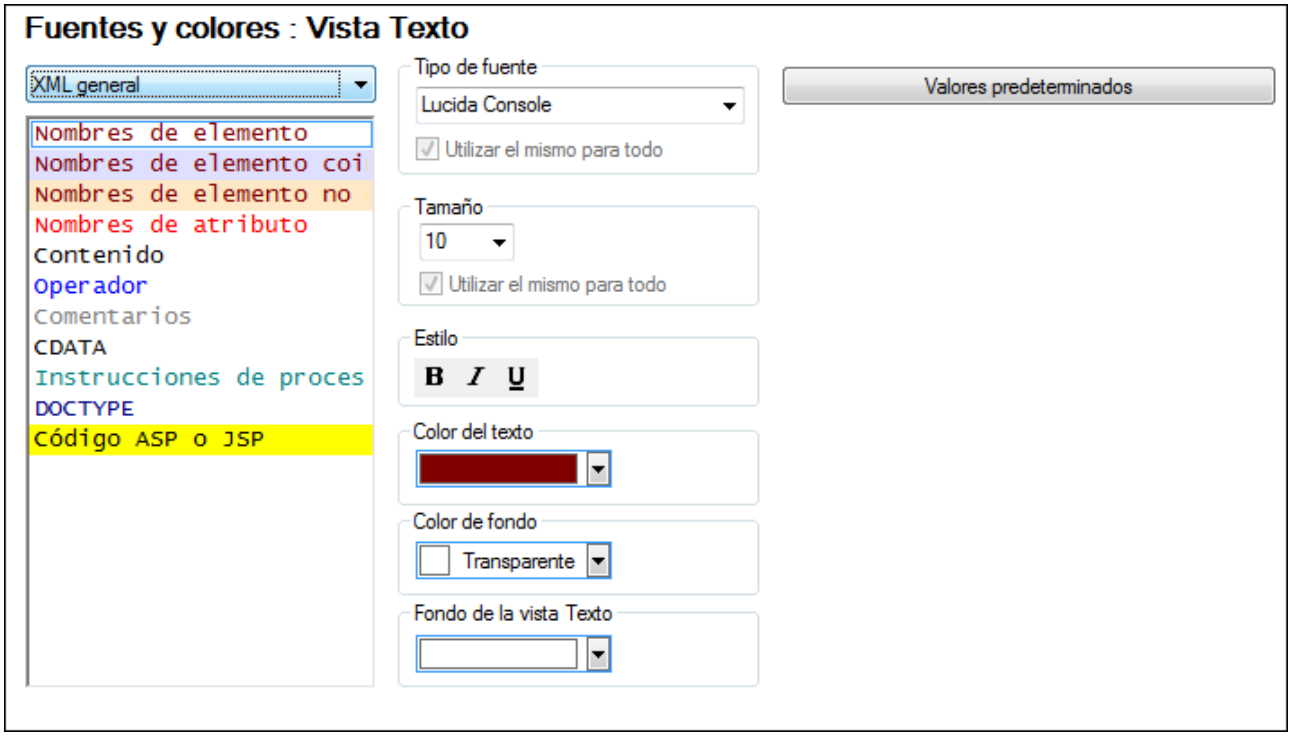
Aquí puede elegir opciones de configuración para estas vistas:

- Vista Texto 1603
- Vista Cuadrícula 1606
- Vista Diseño de esquemas 1608
- Vista WSDL 1611
- Vista Taxonomía XBRL 1612

Seleccione la vista que desea personalizar en el panel izquierdo del cuadro de diálogo. Los tipos de elemento de texto que se pueden configurar aparecen en el panel derecho (*imagen anterior*). Seleccione el tipo de elemento de texto que desea configurar y después asígnele los valores de formato necesarios.

28.17.15.8.1 Vista Texto

En la sección ***Vista Texto*** puede personalizar el aspecto de los componentes de texto en varios tipos de documentos (*imagen siguiente*). Por ejemplo, puede personalizar el formato con que se presentan los nombres de elemento y los nombres de atributo en los documentos XML o el formato con que se presentan las palabras claves y las variables en los documentos XQuery.



Nota: el formato se define desde el tema activo[1629]. Para configurar otro tema primero debe seleccionarlo como tema activo..

Tipos de documentos

Puede seleccionar un tipo de documento específico en el cuadro combinado que hay en la parte superior izquierda del cuadro de diálogo y después configurar el formato de los elementos de texto de ese tipo de documento.

Estos son los tipos de documento de la vista Texto que se pueden configurar:

- XML general
- XQuery
- CSS
- JSON
- C-family
- Python
- XULE
- Varios
- Salida

Instrucciones

Siga estos pasos para personalizar los diferentes tipos de componente de texto:

1. En el cuadro combinado de la esquina superior izquierda seleccione el tipo de documento cuyos componentes de texto desea configurar. En el cuadro inferior aparecen los diferentes componentes de

© 2016-2022 Altova GmbH

texto disponibles para el tipo de documento seleccionado. (En el ejemplo de la imagen anterior seleccionamos el tipo de documento `XML general`).
2. Haga clic en el tipo de componente de texto cuya fuente desea personalizar. (En el ejemplo de la imagen anterior seleccionamos `Nombres de elemento`.)
3. Elija las propiedades de fuente del tipo de componente de texto seleccionado (con ayuda de las opciones de los paneles situados a la derecha). La opción *Fondo de la vista Texto* sirve para seleccionar un color de fondo para toda la vista Texto.

Notas:

- Tenga en cuenta que se utiliza la misma fuente, el mismo tamaño y el mismo estilo para todos los tipos de componente de texto de cada tipo de documento (p. ej. para el tipo de documento XML general). Dentro de cada tipo de documento, solamente se puede modificar el color del texto y color de fondo de cada uno de los componentes de texto. Esto permite el funcionamiento de la característica *color de sintaxis*.
- En la categoría **XML general**, el tipo de componente de texto `Nombres de elemento` está compuesto por tres subtipos: (i) `Nombres de elemento` se aplica a los nombres de elemento que no están seleccionados, (ii) `Nombres de elemento coinciden` se aplica a los nombres de elemento que están seleccionados y cuya etiqueta inicial tiene el mismo nombre que la etiqueta final y (iii) `Nombres de elemento no coinciden` se aplica a los nombres de elemento que están seleccionados y cuya etiqueta inicial no tiene el mismo nombre que la etiqueta final. Es decir, con estos colores de fondo puede marcar los elementos que se están editando en función de si el nombre de las etiquetas coinciden o no. En este cuadro de diálogo solo puede modificar el color de fondo con el que se marcan los elementos. Para desactivar totalmente el marcado desactive la casilla *Resaltar elementos* en el cuadro de diálogo Configurar la vista Texto[1489].
-
- La configuración de la categoría XULE se aplican a: (a) el documento XULE de la ventana principal y (ii) las reglas XULE que se introducen en la ventana XULE[924].
- En la categoría **Varios**, (i) el tipo de texto `Selección` se refiere al contenido de texto que está seleccionado, mientras que `Selección: inactiva` se refiere a las demás instancias del mismo contenido de texto; (ii) `Marcador de búsqueda` se refiere a la instancia seleccionada de un resultado de la búsqueda, mientras que `Buscar marcador activo` se refiere a las demás instancias inactivas del resultado de la búsqueda; (iii) `Marcador de depuración/llamada` se refiere al paso seleccionado en ese momento en una sesión de depuración; (iv) `Espacio en blanco visible` se refiere a los marcadores de espacios en blanco en un documento. Los espacios en blanco de un documento se pueden hacer visibles activando los marcadores de espacios en blanco[1489]; (v) `Asignación inversa activa` se refiere al contenido que está seleccionado en el grupo de documentos de la sesión de asignación inversa[1401] (**`XML—XSLT/XQuery—Resultado`**), mientras que `Asignación inversa inactiva` se refiere al contenido controlado por la asignación inversa que está en los otros dos documentos.

Valores predeterminados

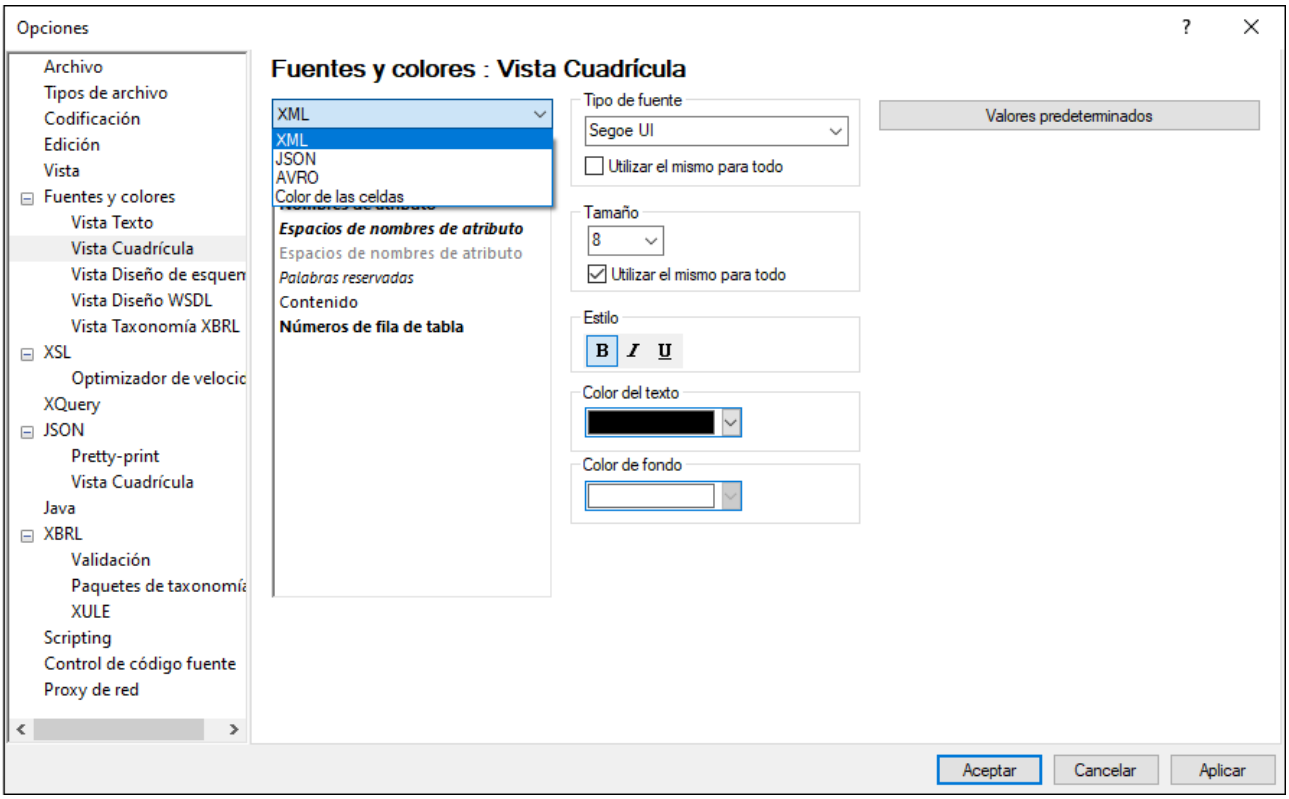
El botón **Valores predeterminados** restaura las fuentes a su estado original de instalación.

Guardar y salir

Para terminar haga clic en **Aceptar**.

28.17.15.8.2 Vista Cuadrícula

En la sección ***Vista Cuadrícula*** (*imagen siguiente*) podrá personalizar el aspecto del texto que aparece en la Vista Cuadrícula XML[169], la Vista Cuadrícula JSON[679] y la Vista Cuadrícula Avro[773] La opción *Color de las celdas* (*imagen siguiente*) permite definir el color de las celdas cuando los componentes de la cuadrícula se muestran como tablas. En el cuadro combinado puede seleccionar el tipo de documento para el que desea configurar la vista Cuadrícula. Después seleccione el tipo de componente de texto al que desea personalizar y asígnele las propiedades de formato necesarias (ver más abajo).



Nota: el formato se define desde el tema activo[1629]. Para configurar otro tema primero debe seleccionarlo como tema activo..

Vista Cuadrícula XML, vista Cuadrícula JSONy vista Cuadrícula Avro

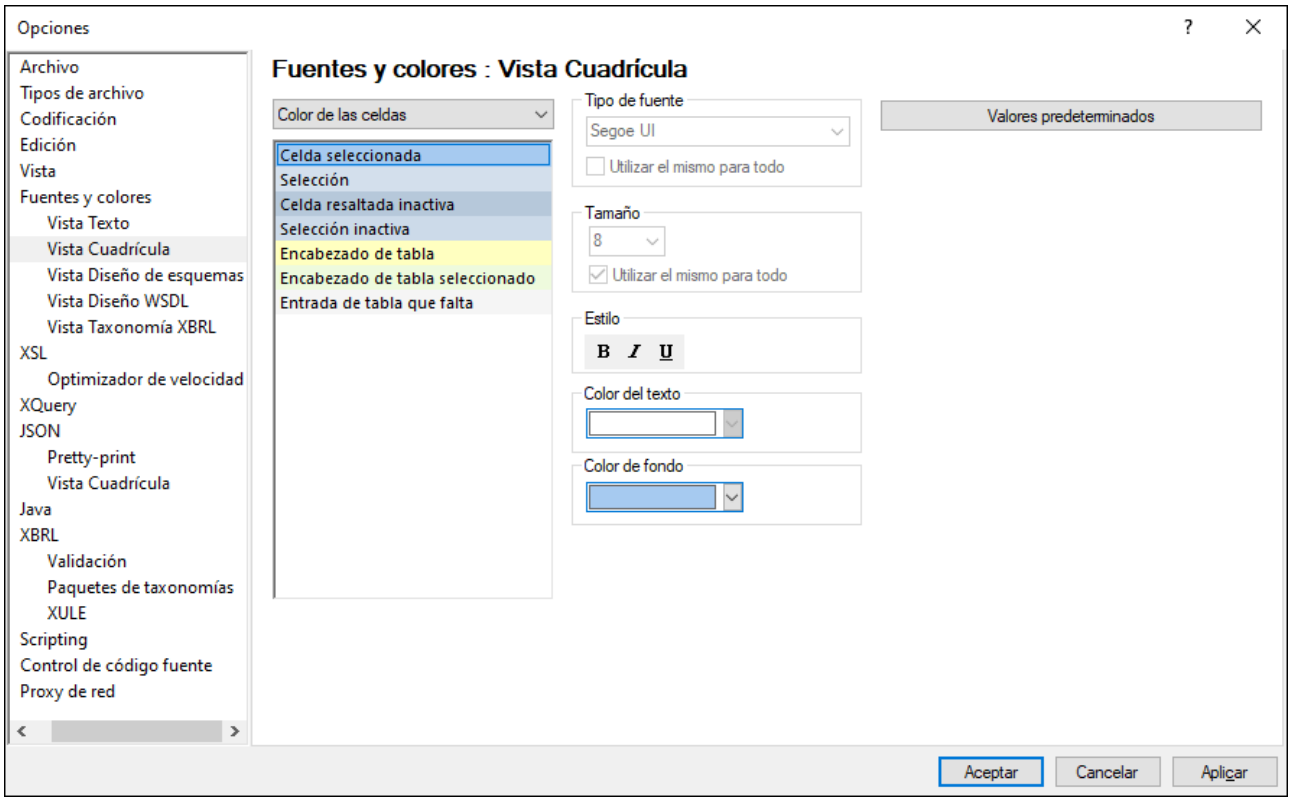
Estas son las propiedades que puede personalizar por cada tipo de componente de texto:

- *Tipo de fuente y tamaño:* la fuente seleccionada también se usará cuando se imprima la vista Cuadrícula. Si quiere usar el mismo tipo de fuente o tamaño para todos los tipos de componente de texto, marque la casilla *Utilizar el mismo para todo* según corresponda.
- *Estilo, color del texto y color de fondo:* establece el estilo y color del texto y el color de fondo de cada tipo de componente de texto. La configuración actual se refleja inmediatamente en la lista del panel izquierdo a modo de vista previa.

© 2016-2022 Altova GmbH

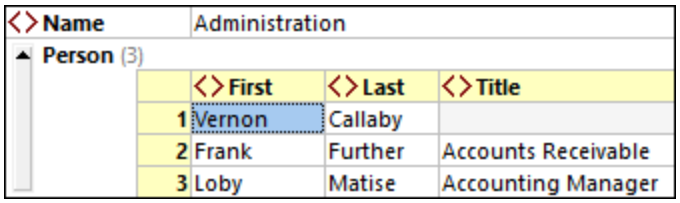
Color de la celda (de las tablas de todas las vistas Cuadrícula)

Las tablas de las distintas vistas Cuadrícula (XML, JSON, Avro) se pueden configurar no solamente en función de la semántica relacionada con el lenguaje (por ejemplo, un elemento XML puede tener un formato distinto que un comentario XML). También puede configurar la Vista Cuadrícula según la función de las celdas: por ejemplo, para diferenciar entre el estado seleccionado/no seleccionado de los distintos componentes de una tabla. El mejor método para esa diferenciación es asignar un color de fondo distinto a cada elemento (*imagen siguiente*), pero también puede usar otros propiedades para diferenciar cada tipo de celda.

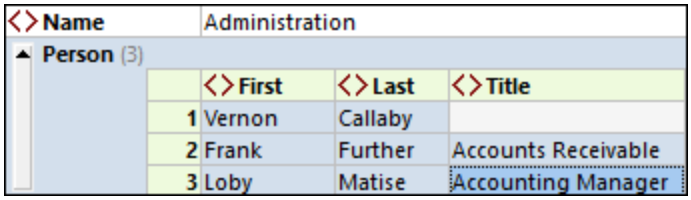


Existen opciones para estos componentes:

- *Encabezado de la tabla (no seleccionado) y Encabezado de tabla seleccionado*: estas dos opciones hacen referencia a los encabezados de columna y de fila. La imagen siguiente muestra encabezados no seleccionados; el color de fondo de estos encabezados es el que se definió en el cuadro de diálogo anterior. El color definido para los encabezados seleccionados se activa cuando se seleccionan todos los encabezados, no cuando se selecciona uno solo. Para seleccionar todos los encabezados haga clic en la celda en la que se cruzan el encabezado de la columna y el de la fila o seleccione el elemento creado como tabla o cualquiera de sus elementos de nivel superior.



- *Entrada de tabla que falta:* hace referencia a elementos o atributos del documento (*imagen siguiente*).
- *Selección y celda resaltada:* hace referencia a las celdas seleccionadas y a las que están activas. Por ejemplo, en la imagen siguiente se ha seleccionado toda la tabla `Person` y la celda de abajo a la derecha está resaltada.



- *Selección inactiva, celda resaltada inactiva*: si se selecciona un bloque de celdas y se resaltan una o más celdas en ese bloque, tanto la selección como la celda resaltada están activas mientras también lo esté la cuadrícula. Sin embargo, si se activa otro cuadro de diálogo u otra ventana sin haber tocado la selección ni la celda resaltada, entonces ambas pasarán a estar inactivas. Si la Vista Cuadrícula vuelve a estar activa, entonces la selección y la celda resaltada también vuelven a estarlo.

Nota: además de los colores que defina aquí el usuario, XMLSpy utiliza las preferencias sobre el uso de colores de la configuración de pantalla (en el panel de control) de su instalación de Windows.

Valores predeterminados

El botón **Valores predeterminados** restaura las fuentes a su estado original de instalación.

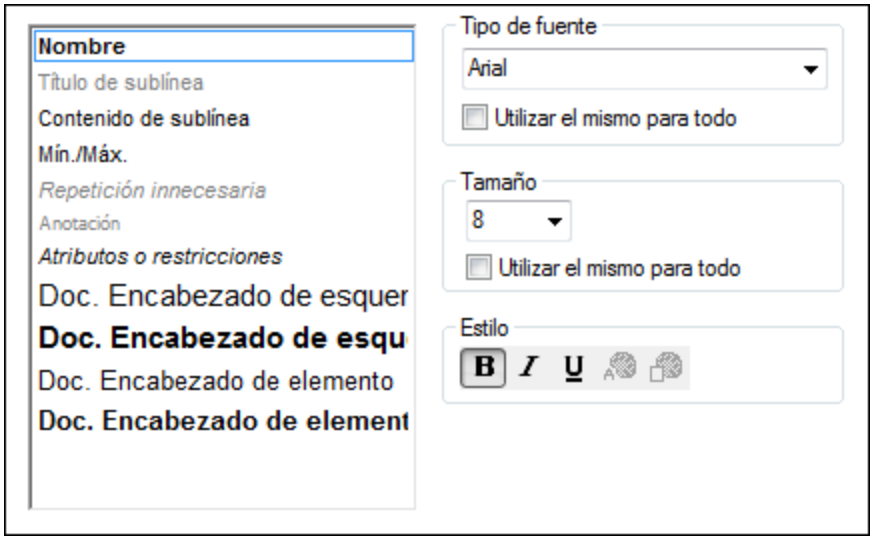
Guardar y salir

Para terminar haga clic en **Aceptar**.

28.17.15.8.3 Vista Diseño de esquemas

En la sección ***Vista Diseño de esquemas*** puede personalizar cómo se presentan los componentes de texto de esquemas XML 444 y esquemas JSON 671 en la vista Esquema 171.

© 2016-2022 Altova GmbH

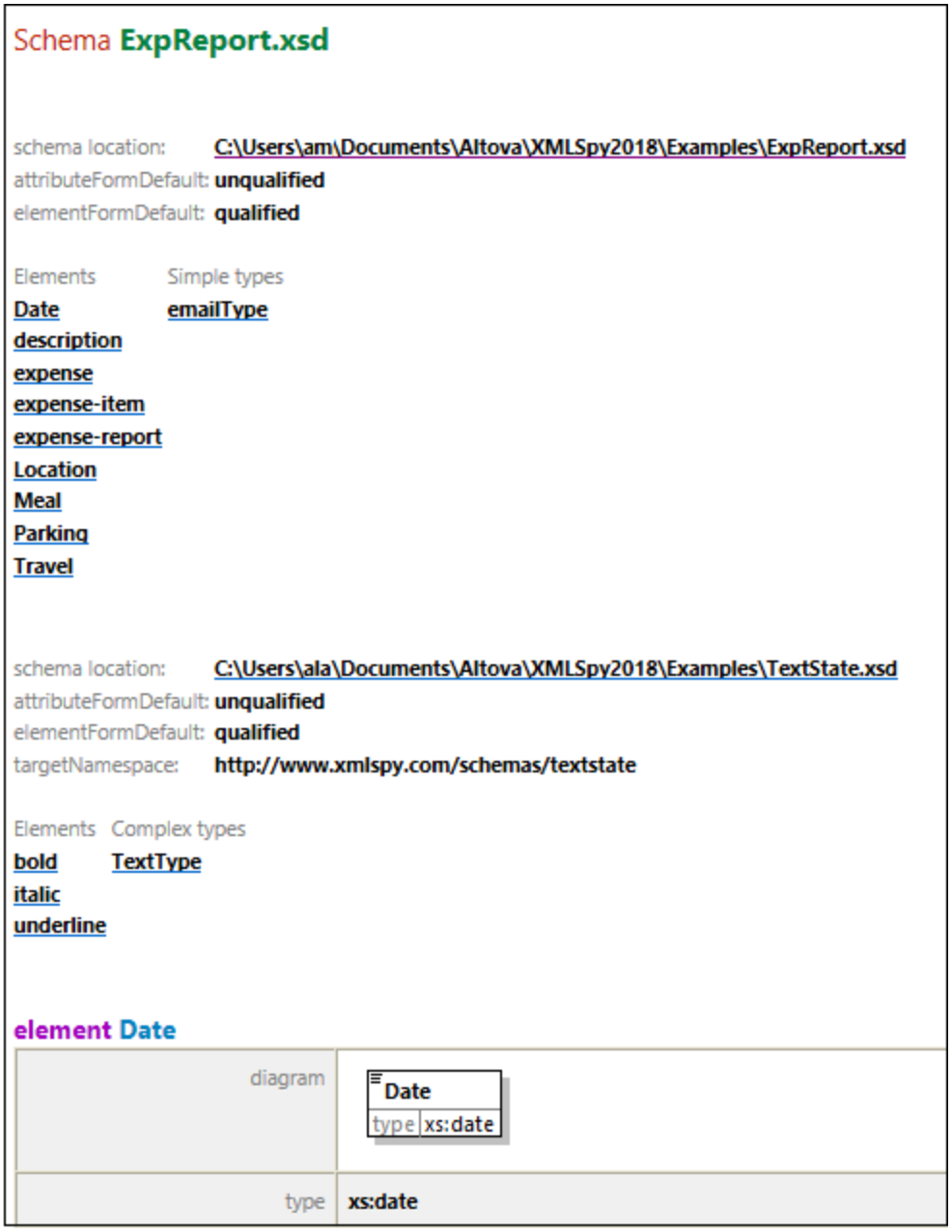


Instrucciones

Siga estos pasos para personalizar los diferentes tipos de componente de texto:

1. En el cuadro combinado de la esquina superior izquierda seleccione **Esquema XML** o **Esquema JSON**. Los componentes de texto del tipo de documento seleccionado aparecen en el recuadro inferior.
2. Haga clic en el tipo de componente de texto cuyo formato desea personalizar.
3. Elija las propiedades de fuente del tipo de componente de texto seleccionado (con ayuda de las opciones de los paneles situados a la derecha).

Nota:
Los componentes de texto *Doc. Encabezado de esquema(2)* y *Doc. Encabezado de elemento(2)* se refieren al encabezado de esquema y a los encabezados de elemento de la documentación del esquema que se puede generar con la aplicación[1373]. Compare los colores de estas propiedades en el cuadro de diálogo de la imagen anterior con los colores del encabezado de esquema y del encabezado de elemento de la documentación que se genera para el esquema (*imagen siguiente*).



Valores predeterminados

El botón **Valores predeterminados** restaura las fuentes a su estado original de instalación.

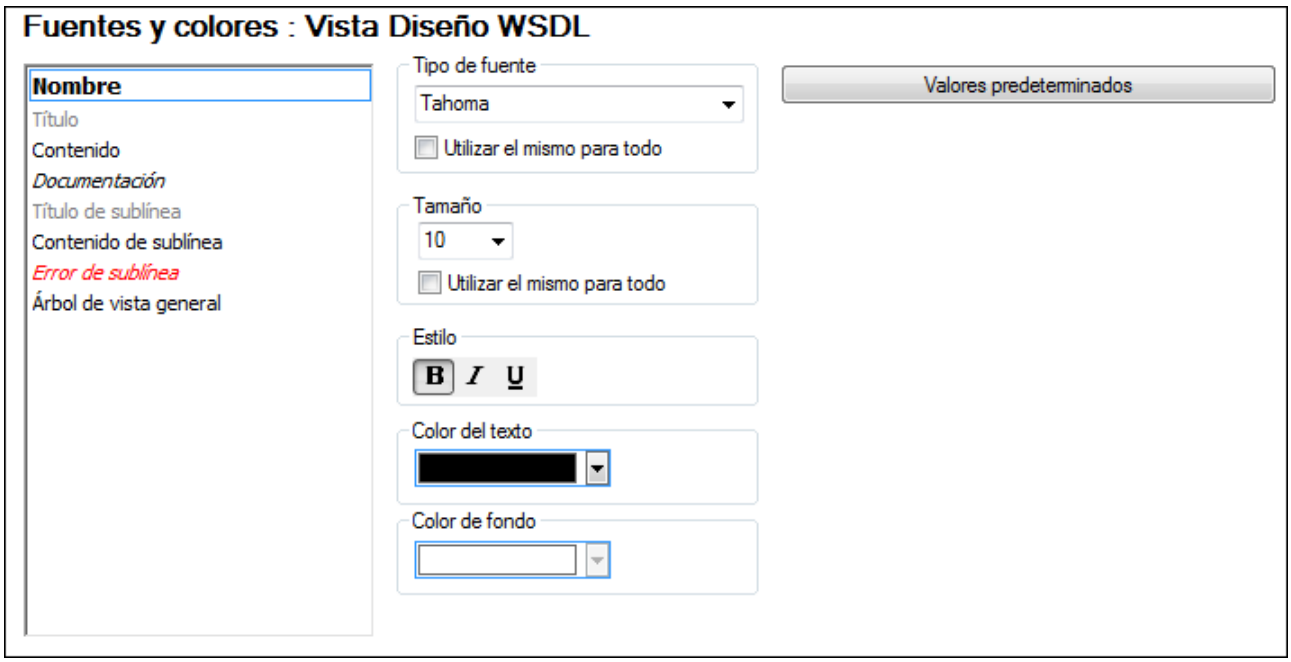
Guardar y salir

Para terminar haga clic en **Aceptar**.

© 2016-2022 Altova GmbH

28.17.15.8.4 Vista Diseño WSDL

La sección ***Vista Diseño WSDL*** ofrece opciones para personalizar el aspecto de los componentes de texto en la vista WSDL[251] (*imagen siguiente*). El formato se define desde el tema activo[1629]. Para configurar otro tema primero debe seleccionarlo como tema activo.



Estas son las propiedades que puede configurar para cada tipo de componente de texto:

- *Tipo de fuente y tamaño:* la fuente seleccionada también se usará cuando se imprima la vista WSDL. Si quiere usar el mismo tipo de fuente o tamaño para todos los tipos de componente de texto, marque la casilla *Utilizar el mismo para todo* según corresponda.
- *Estilo, color del texto y color de fondo:* establece el estilo y color del texto y el color de fondo de cada tipo de componente de texto. La configuración actual se refleja inmediatamente en la lista del panel izquierdo a modo de vista previa.

Nota:
El formato del encabezado WSDL y de los nombres de servicio de la documentación del documento WSDL[1499] se toma respectivamente de los componentes *Doc. Encabezado de esquema(2)* y *Doc. Encabezado de elemento(2)* configurados en la sección *Vista Diseño de esquemas*[1608].

Valores predeterminados

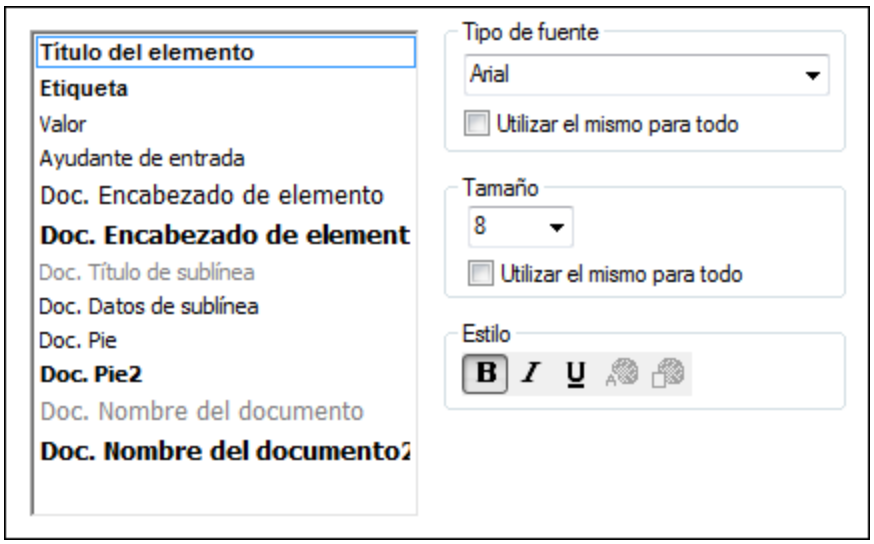
El botón **Valores predeterminados** restaura las fuentes a su estado original de instalación.

Guardar y salir

Para terminar haga clic en **Aceptar**.

28.17.15.8.5 Vista Taxonomía XBRL

La sección **Vista Taxonomía XBRL** ofrece opciones para personalizar el aspecto de los componentes de texto en la vista XBRL[264] (*imagen siguiente*).



Estas son las propiedades que puede configurar para cada tipo de componente de texto:

- *Tipo de fuente y tamaño:* la fuente seleccionada también se usará cuando se imprima la vista XBRL. Si quiere usar el mismo tipo de fuente o tamaño para todos los tipos de componente de texto, marque la casilla *Utilizar el mismo para todo* según corresponda.
- *Estilo, color del texto y color de fondo:* establece el estilo y color del texto y el color de fondo de cada tipo de componente de texto. La configuración actual se refleja inmediatamente en la lista del panel izquierdo a modo de vista previa.
- Las opciones de vista previa de tablas XBRL permiten configurar los componentes de las tablas XBRL, que se pueden ver como tales en la pestaña *Tablas* de la Vista XBRL.

Valores predeterminados

El botón **Valores predeterminados** restaura las fuentes a su estado original de instalación.

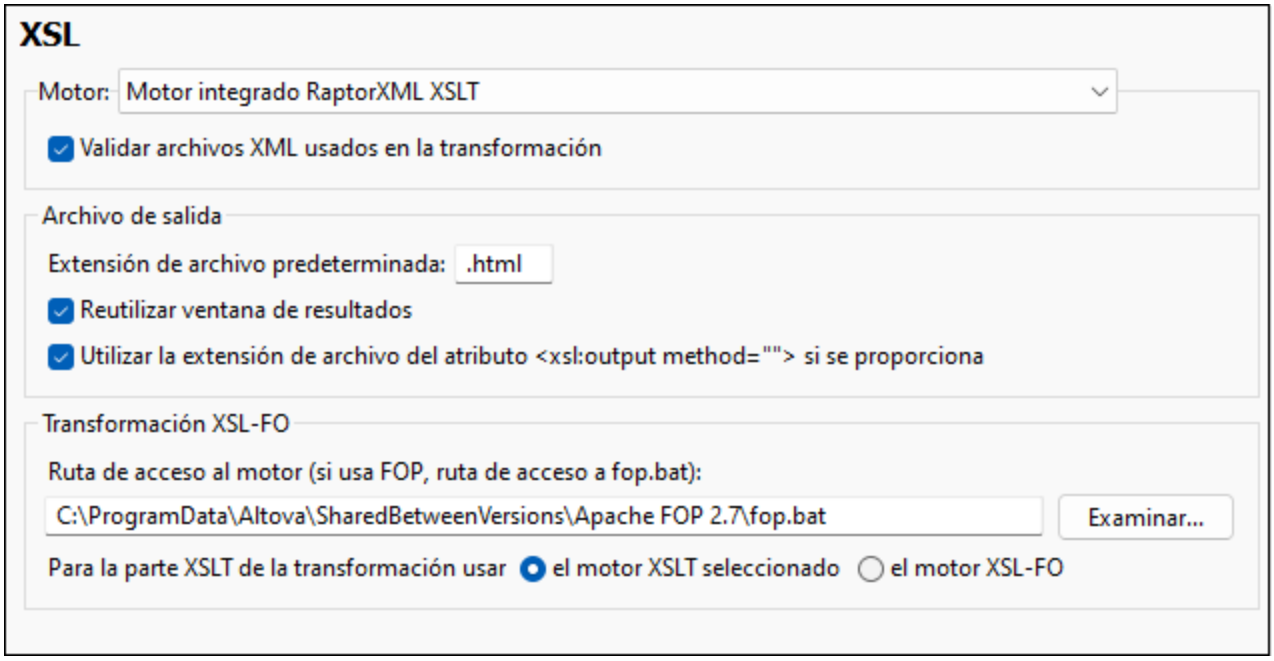
Guardar y salir

Para terminar haga clic en **Aceptar**.

28.17.15.9 XSL

En la sección ***XSL*** (*imagen siguiente*) puede definir las opciones de configuración para las transformaciones XSL[1613] y las transformaciones XSL-FO[1615] realizadas dentro de la aplicación.

© 2016-2022 Altova GmbH



Configuración del motor

Puede configurar un procesador XSLT para ejecutar transformaciones XSLT cuando se invoca el comando **Transformación XSLT**[1394].

Hay varios motores XSLT de entre los que elegir:

- El motor XSLT integrado de RaptorXML
- El analizador MSXML de Microsoft (MSXML)
- Un procesador XSLT externo

Nota: para depurar XSLT en XMLSpy se usa siempre el motor XSLT integrado de RaptorXML, incluso aunque aquí se haya seleccionado otro motor XSLT para las transformaciones.

Motor XSLT de Altova RaptorXML

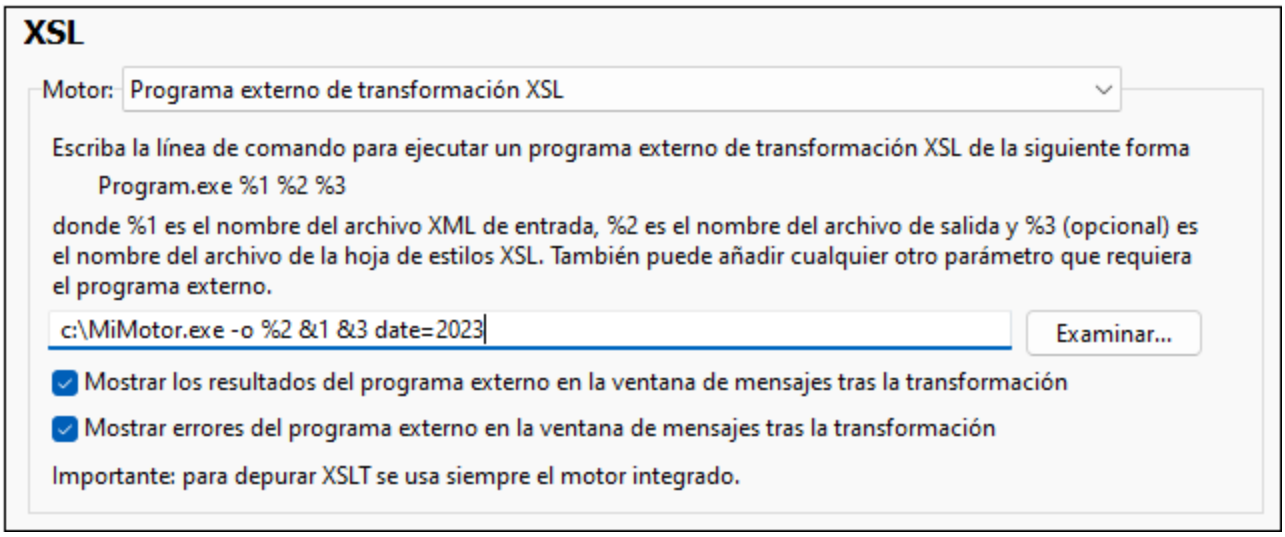
XMLSpy incluye los motores XSLT 1.0, XSLT 2.0 y XSLT 3.0 de Altova Raptor XML para transformaciones XSLT. Dependiendo del valor del atributo `version` del elemento `xsl:stylesheet` o `xsl:transform`, en la transformación se utiliza un motor u otro. Ocurre lo mismo al depurar XSLT con el depurador XSLT/XQuery de XMLSpy.

Para validar los archivos XML de las transformaciones, seleccione la opción Validar (*imagen anterior*).

Analizador (MSXML) de Microsoft

El analizador MSXML 3.0, 4.0 o 6.0 es una utilidad que viene preinstalada en su equipo. Si sabe qué versión utiliza su equipo, seleccione la opción correspondiente. Si no lo sabe, puede pedir a XMLSpy que elija la versión automáticamente (*Elegir la versión automáticamente* es la opción predeterminada). En ese caso, XMLSpy intenta seleccionar la versión más reciente.

Procesador XSLT externo
Puede elegir el procesador XSLT externo que usted prefiera introduciendo la ruta de acceso a su archivo ejecutable.



Debe indicar la cadena de línea de comandos que el procesador XSLT externo utiliza para ejecutar transformaciones. Estas son las variables que puede usar para crear la cadena de línea de comandos:

`%1` = documento XML que se debe procesar
`%2` = archivo de salida que se debe generar
`%3` = hoja de estilos XSLT que se debe usar (si el documento XML no contiene una referencia a la hoja de estilos)

Por ejemplo, imagine que tiene un procesador que utiliza este comando para ejecutar transformaciones XSLT:

```
myxsltengine.exe -o <output.xml> <input.xml> <stylesheet.xslt> <param-name>=<param-value>?
```

Entonces en XMLSpy puede usar las variables descritas más arriba para generar este comando. Por ejemplo:

```
c:\MyEngine.exe -o %2 %1 %3 date=2023
```

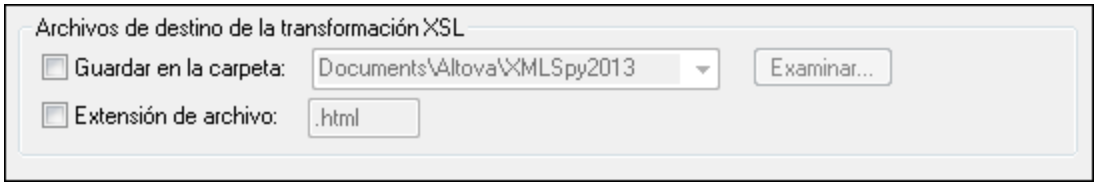
Por último, si desea ver los resultados y los errores del programa externo en la ventana Mensajes de XMLSpy marque las casillas correspondientes.

Nota: los parámetros definidos en el cuadro de diálogo Parámetros de entrada de XSLT[1396] de XMLSpy solamente se pasan a los motores XSLT integrados de Altova. Estos parámetros no se pasan a los procesadores externos.

© 2016-2022 Altova GmbH

Configurar el archivo de salida

- *Extensión de archivo predeterminada:* define la extensión de los archivos de salida; puede sobreescribir esta extensión cambiando la extensión de archivo del elemento XSLT **`xsl:output`** (*último punto*).
- *Reutilizar ventana de resultados:* las transformaciones que realice a partir de ahora mostrarán el documento de resultados en la misma ventana. Si el archivo XML pertenece a un proyecto y la casilla *Reutilizar ventana de resultados* está desactivada, la opción solamente surte efecto si también se desactiva la casilla *Guardar en la carpeta* de las propiedades del proyecto[1328] (*imagen siguiente*).



- *Usar extensión de archivo del elemento xsl:output:* indica si la extensión de archivo indicada en el elemento **`xsl:output`** de la hoja de estilos XSLT sobreescribe la extensión predeterminada indicada en la primera opción de esta lista.

Transformaciones XSL-FO

Los documentos FO se procesan usando un procesador FO y la ruta de acceso del ejecutable del procesador FO debe indicarse en la casilla del motor de transformación XSL-FO. La transformación se lleva a cabo con el comando de menú **XSL/XQuery | Transformación XSL-FO**[1395]. Si el archivo de entrada (el documento que estaba activo cuando se ejecutó el comando) es un documento XSL-FO, se invoca al procesador FO para la transformación. Si el documento de entrada es un documento XML, entonces es necesario realizar primero una transformación XSLT para convertir el documento XML en un documento XSL-FO. Esta transformación XSLT se puede llevar a cabo bien con el motor XSLT que indicó como motor predeterminado (*véase más arriba*[1613]) o bien con el motor XSLT que está integrado en el procesador FO que indicó como procesador FO predeterminado para la aplicación. Seleccione cuál de estos dos motores desea usar para esta transformación.

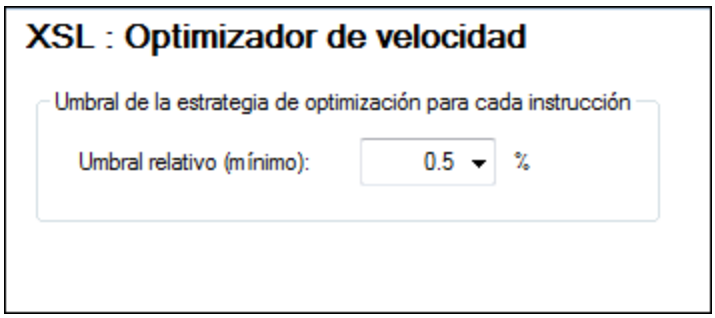
Nota: A no ser que desactive la casilla correspondiente en el programa de instalación de XMLSpy, el procesador FOP de Apache XML Project se instalará en la carpeta `C:\ProgramData\Altova\SharedBetweenVersions`. Y si el procesador está instalado, su ruta de acceso aparecerá automáticamente en el campo *Motor XSL-FO*. Si quiere puede cambiar esta ruta de acceso para usar otro procesador FO distinto. Sin embargo, recuerde que los demás productos de Altova que usan procesadores FO utilizarán la misma ruta de acceso (StyleVision y Authentic Desktop).

Guardar y salir

Cuando termine de modificar las opciones de configuración, haga clic en **Aceptar** para terminar.

28.17.15.9.1 Optimizador de velocidad

En la sección ***XSL: Optimizador de velocidad*** (*imagen siguiente*) puede configurar el optimizador de velocidad XSL [503].



Aquí puede especificar el umbral del optimizador para cada instrucción XSLT de una hoja de estilos XSLT. Puede elegir valores comprendidos entre un 0.1% de la duración total de la transformación y un 99%. Si una instrucción tarda en ejecutarse más tiempo del indicado como umbral, entonces se invoca el análisis de optimización. De lo contrario, en análisis no se lleva a cabo. Si el análisis de optimización no puede derivar una estrategia de optimización, puede que el umbral indicado sea demasiado alto. Baje el umbral y vuelva a intentarlo.

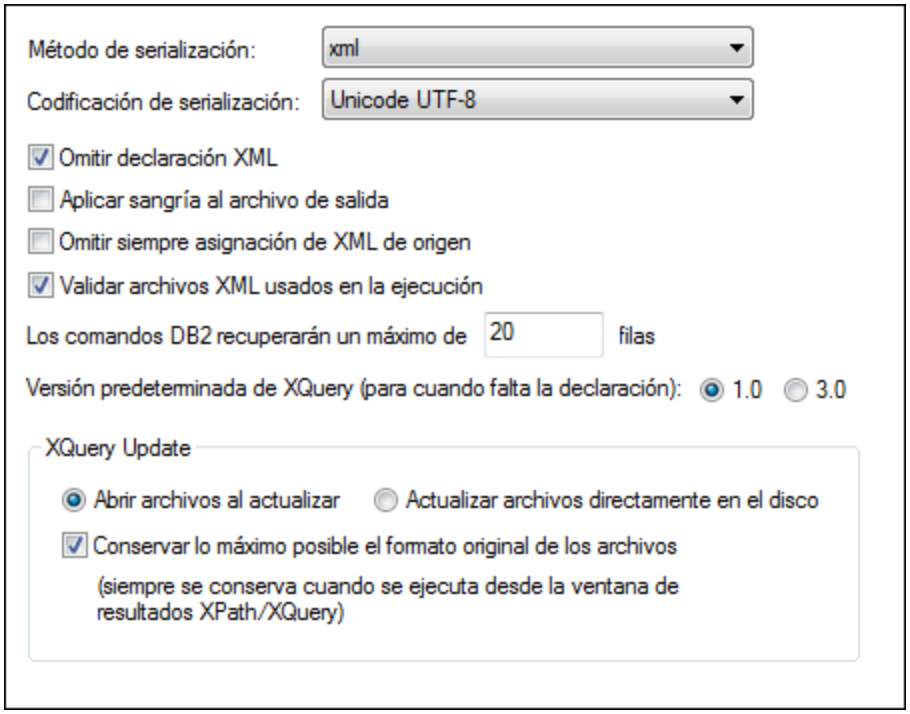
Guardar y salir

Cuando termine de modificar las opciones de configuración, haga clic en **Aceptar** para terminar.

28.17.15.10 XQuery

En la sección ***XQuery*** (*imagen siguiente*) puede configurar algunos aspectos de la edición y ejecución de documentos XQuery y XQuery Update.

© 2016-2022 Altova GmbH



Opciones generales de XQuery

- ***Serialización:*** la serialización se refiere al modo en que se escribe texto en el documento de salida. Puede elegir tanto el método de serialización (adaptable*, HTML, JSON, texto, XHTML o XML) como la codificación de serialización de la salida para distintos tipos de entrada. La serialización del documento de salida se puede seleccionar para entrada XML y JSON. (*Nota: el método adaptable permite procesar un documento de instancia sin errores y determina automáticamente el método de serialización en base al documento de entrada.*)
- ***Omitir declaración XML:*** marque esta casilla para omitir la declaración XML en el documento serializado (el documento de salida).
- ***Aplicar sangría al archivo de salida:*** marque esta casilla para aplicar sangría al documento de salida y ver la jerarquía del documento.
- ***Omitir siempre asignación de XML de origen:*** cuando se ejecuta un documento XQuery, XMLSpy puede pedirle que asigne el documento XML de origen en el que se debe ejecutar el documento XQuery. Marque esta casilla si quiere saltarse este paso y ejecutar el documento XQuery directamente. Si marca esta casilla, el documento XQuery debería ejecutarse correctamente sin necesidad de pasarle un documento XML. Esto puede ser porque no se necesita ningún documento XML o porque a los datos XML se accede por medio de las funciones que incluye el documento XQuery.
- ***Validar archivos XML usados en la ejecución:*** valida los archivos XML utilizados en la ejecución de documentos XQuery. La aplicación indica qué archivos XML no son válidos y el documento XQuery no se procesa.
- ***Recuperación de filas DB2:*** cuando trabaje con datos de BD, puede especificar el número máximo de filas que se deben recuperar. XMLSpy reconoce las extensiones de archivo[1587] `.xqr` como archivos XQuery para BD.

- ***Versión predeterminada de XQuery:*** aquí puede indicar qué versión del motor XQuery se utiliza para ejecutar documentos XQuery en los que falte la palabra clave `version`. Esto afecta tanto a documentos XQuery como XQuery Update y selecciona el motor XQuery predeterminado.

Opciones de XQuery Update

- ***Actualización de archivos:*** cuando se ejecuta un archivo XQuery Update, los archivos XML de destino se pueden actualizar directamente en el disco o se pueden abrir en XMLSpy y actualizarse en memoria. Seleccione la opción *Abrir archivos al actualizar* para repasar las actualizaciones y después guardar los cambios o rechazarlos (cerrando el archivo sin guardar).
- ***Conservar formato original:*** marque esta casilla si quiere conservar lo máximo posible el formato original de los archivos actualizados.

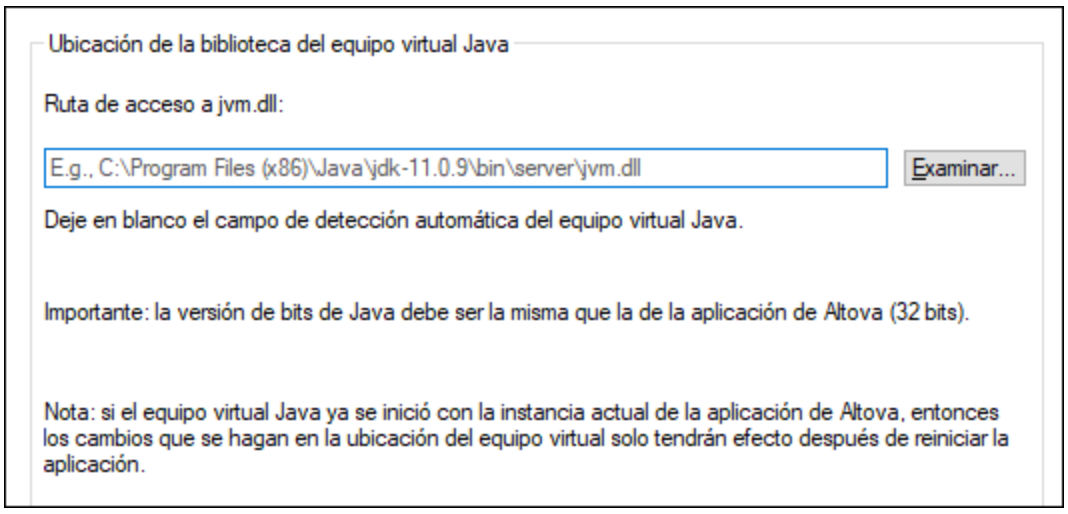
Guardar y salir

Cuando termine de modificar las opciones de configuración, haga clic en **Aceptar** para terminar.

28.17.15.11 Java

En la pestaña *Java* puede introducir la ruta de acceso a un equipo virtual java en su sistema de archivos. Tenga en cuenta que no siempre es necesario agregar una ruta de acceso personal a un equipo virtual. Por defecto, XMLSpy intenta detectar esta ruta automáticamente leyendo (en este orden) el registro de Windows y la variable de entorno JAVA_HOME. Si se detecta automáticamente cualquier otra ruta de equipo virtual java, tendrá prioridad la ruta personal que se indica en este cuadro de diálogo.

Puede que necesite añadir esta ruta personal de acceso a un equipo virtual java si está usando un equipo virtual java que no tiene instalador ni crea entradas de registro (por ejemplo, OpenJDK, de Oracle). También puede querer usar esta ruta para suprimir, por la razón que fuere, cualquier otra ruta que XMLSpy haya detectado automáticamente.



Observe lo siguiente:

 © 2016-2022 Altova GmbH

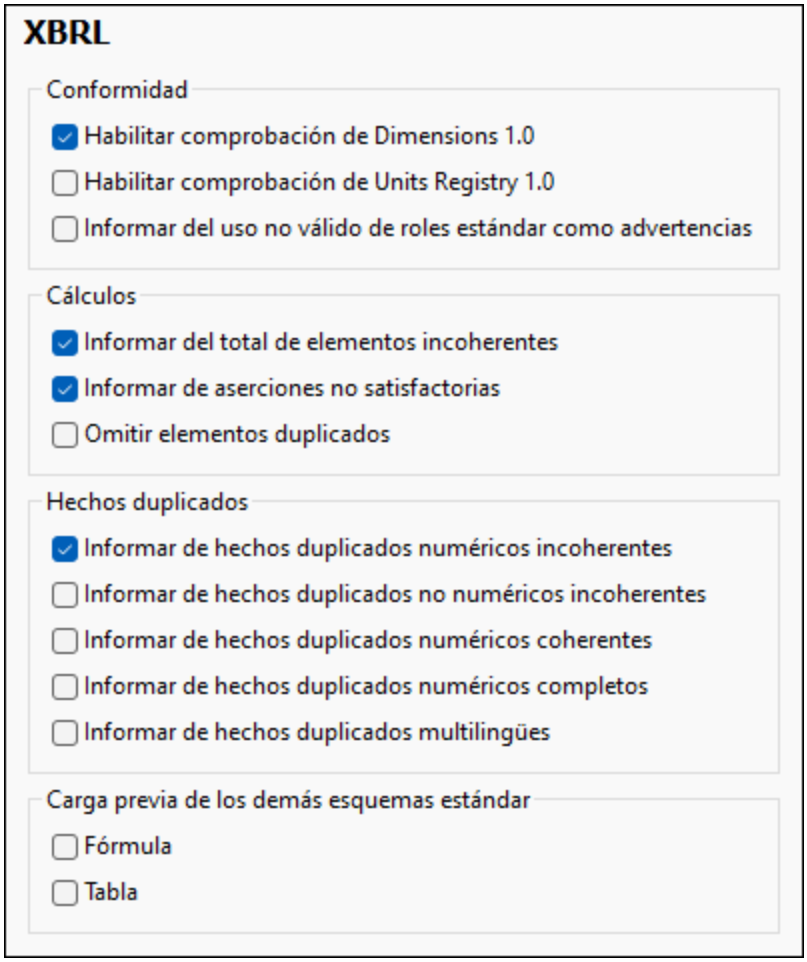
- la ruta de acceso al equipo virtual java es común a todas las aplicaciones de escritorio de Altova (no a las de servidor). En consecuencia, si cambia esta ruta en una de ellas, el cambio afectará automáticamente al resto de aplicaciones de Altova.
- la ruta debe apuntar al archivo jvm.dll desde los directorios **\bin\server** o **\bin\client**, relativos al directorio en el que está instalado el JDK.
- la plataforma de XMLSpy (versión de 31 o de 64 bits) debe ser la misma que la del JDK.
- después de cambiar la ruta de acceso al escritorio virtual java debe reiniciar XMLSpy para que surta efecto la nueva configuración.

Cambiar la ruta de acceso al escritorio virtual java afecta a:

- la conectividad JDBC
- las funciones de extensión Java para XSLT/XPath

28.17.15.12 XBRL

La sección **XBRL** ofrece opciones para la validación de documentos de instancia XBRL y XULE y para añadir y gestionar paquetes de taxonomías XBRL[1621].



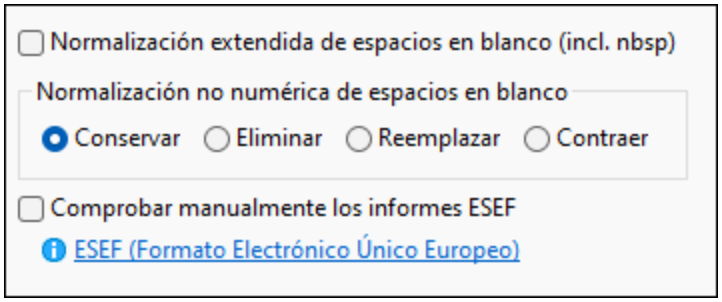
Validación XBRL

En la parte superior de las opciones XBRL (*imagen anterior*) puede indicar estas opciones de validación:

- Comprobar la conformidad con Dimensions 1.0 y Units Registry 1.0..
- Informar sobre (i) elementos incoherentes al sumar elementos, (ii) evaluaciones de aserción no cumplidas. Una opción para informar como advertencia del uso no válido de roles estándar.
- Ignorar duplicados (duplicados completos y duplicados coherentes) implicados en cálculos. Si hay un hecho duplicado se usa el más preciso y se ignoran los demás.
- Informar de duplicados. Para más información sobre cómo se clasifican los duplicados, consulte la especificaciones Handling Duplicate Facts in XBRL e Inline XBRL 1.0.
- Normalización de espacios en blanco en XBRL Inline, que se refiere a espacios en blanco como espacios de la barra espaciadora, tabulaciones, saltos de línea y retornos de carro.

28.17.15.12.1 XBRL inline

La pestaña *Inline XBRL* ofrece opciones de normalización para los espacios en blanco (espacios de la barra espaciadora, tabulaciones, saltos de línea y retornos de carro) en XBRL inline.



Estas son las opciones de validación que se pueden especificar:

- si marca la casilla *Normalización extendida* de espacios en blanco, los espacios en blanco fijos también se consideran espacios en blanco.
- la opción *Conservar* no cambian los espacios en blanco.
- la opción *Eliminar* elimina los espacios en blanco en ambos extremos de las cadenas.
- con la opción *Reemplazar* todas las instancias de tabulaciones, saltos de línea y retornos de carro se reemplazan por un espacio.
- la opción *Contraer* es una extensión de *Reemplazar*, por lo que primero se reemplazan esas instancias por espacios y después todos los espacios contiguos se contraen en uno solo.
- la opción *Comprobar manualmente los informes ESEF:* permite comprobar la validez de un documento que se valida con respecto al Manual de informes ESEF. Para más información sobre el formato ESEF consulte la página web del Formato Electrónico Único Europeo (ESEF por sus siglas en inglés).

© 2016-2022 Altova GmbH

28.17.15.12.2 Paquetes de taxonomías XBRL

Un paquete de taxonomías XBRL es un archivo comprimido que contiene una copia sin conexión de una taxonomía. Cada paquete de taxonomías contiene un archivo XML de catálogo que reasigna los URI a la ubicación de los archivos de la taxonomía sin conexión y, por tanto, pone la taxonomía a disposición de la aplicación sin necesidad de tener conexión a Internet. Las reglas que especifican cómo se debe estructurar y construir un paquete de taxonomías pueden encontrarse en la recomendación sobre paquetes de taxonomías de XBRL.org.

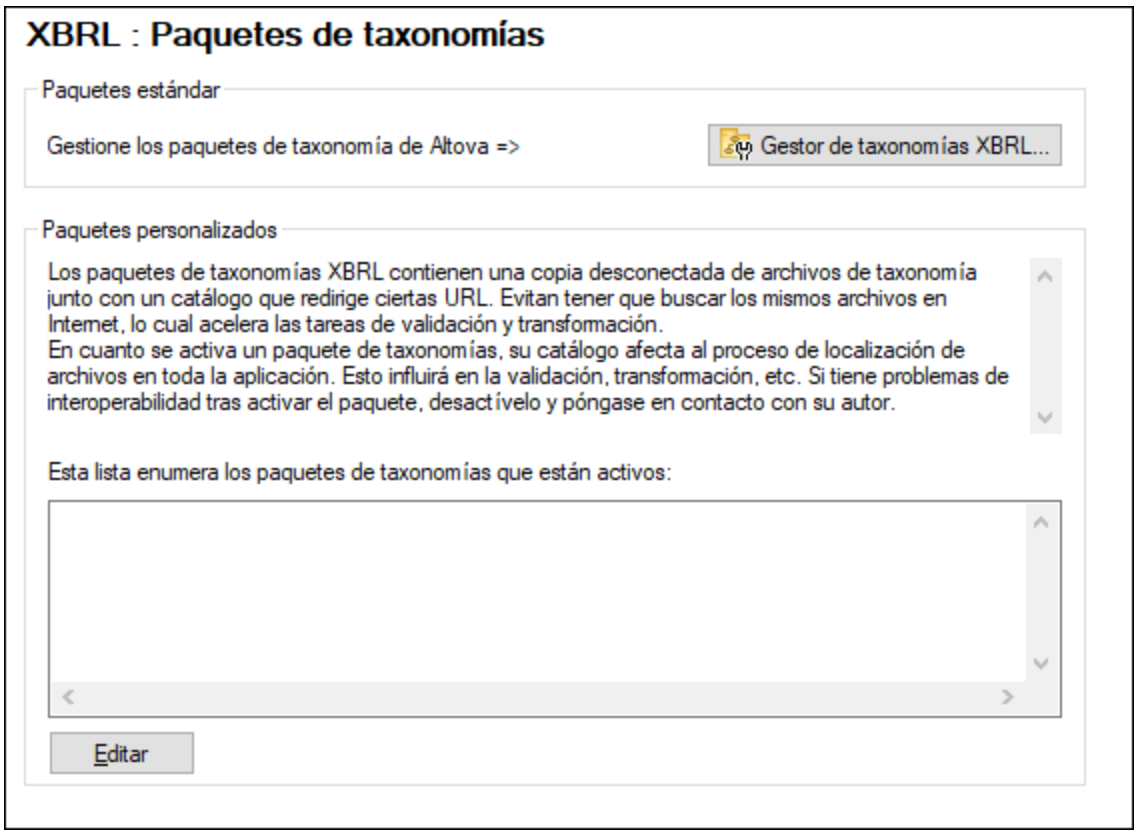
Tras descargar el paquete de taxonomías, podrá configurar XMLSpy para que identifique y utilice el archivo de catálogo de punto de entrada del paquete automáticamente. Para ello basta con añadir el paquete a la lista de paquetes de taxonomías activos. XMLSpy utilizará los archivos de catálogo de los paquetes activos para encontrar recursos para las diferentes operaciones, como por ejemplo las tareas de validación XBRL. Puede agregar dos tipos de paquetes de taxonomía:

- paquetes estándar, que puede administrar fácilmente para todos los productos de Altova con el Gestor de taxonomías XBRL de Altova 818 (consulte la descripción aquí 818).
- paquetes personales, que puede añadir desde este cuadro de diálogo.

Nota: los recursos a los que apunte un archivo de catálogo de un paquete activo se utilizarán para todas las operaciones de XMLSpy que requieran el uso de dichos recursos. Si un recurso difiere de alguna manera del recurso utilizado previamente por XMLSpy, es posible que se detecten errores durante la ejecución de la operación. Para más información consulte la nota que aparece al final de este apartado.

Agregar y administrar paquetes de taxonomías

Seleccione **Herramientas | Opciones | Paquetes de taxonomías XBRL** para abrir la sección *Paquetes de taxonomías XBRL* (*imagen siguiente*).

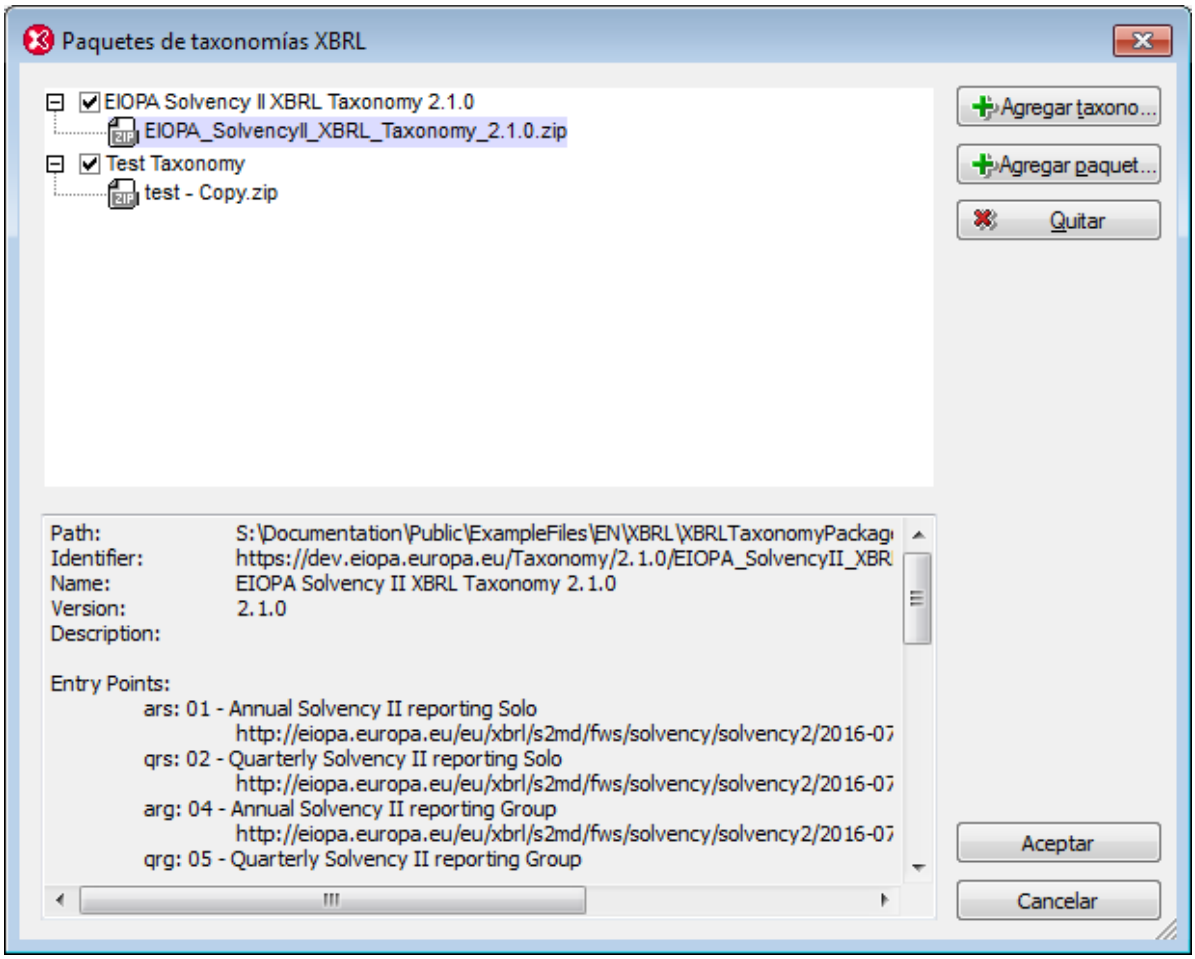


Para añadir un paquete de taxonomía estándar haga clic en **Gestor de taxonomías XBRL**. Para leer las instrucciones del Gestor de taxonomías XBRL de Altova[818] consulte aquí[818] su descripción.

Para añadir un paquete personal siga los pasos que explicamos a continuación.

1. Haga clic en el botón **Editar** para abrir el cuadro de diálogo "Paquetes de taxonomías XBRL" (*imagen siguiente*).

© 2016-2022 Altova GmbH



2. Haga clic en **Agregar taxonomía** y navegue hasta la ubicación del paquete de taxonomías. Selecciónelo y haga clic en **Abrir** (también puede agregar varios paquetes a la vez). El paquete se añadirá a la lista de paquetes de taxonomías del cuadro de diálogo. La lista se presenta de forma de árbol jerárquico de dos niveles. El primer nivel indica el nombre de la taxonomías y el segundo los paquetes que la componen. La casilla situada la izquierda de una taxonomía indica si está activa o no. Las taxonomías se activan por defecto al añadirlas.
3. Para terminar haga clic en **Aceptar**. Los paquetes que acaba de añadir aparecerán en la sección *Paquetes de taxonomías XBRL* del cuadro de diálogo "Opciones" (*primera imagen de este apartado*).

Debe tener en cuenta que:

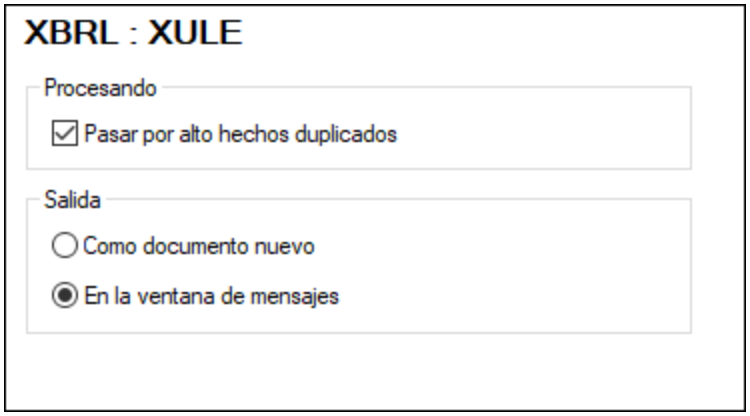
- Si desea añadir un paquete más a una taxonomía, (i) seleccione la taxonomía en el cuadro de diálogo "Paquetes de taxonomías XBRL" (*imagen anterior*) y (ii) añada los paquetes con el botón **Agregar paquetes**. Los paquetes aparecerán en el segundo nivel de la taxonomía.
- Cuando seleccione un paquete de taxonomías en la lista del panel superior del cuadro de diálogo "Paquetes de taxonomías XBRL", toda la información disponible sobre el paquete, incluida su ubicación sin conexión, aparece en el panel inferior del cuadro de diálogo.
- Para desactivar una taxonomía basta con desactivar su casilla de verificación. Si desactiva una taxonomía, sus archivos de catálogo dejarán de utilizarse. Se aconseja desactivar la taxonomía si, por ejemplo, quiere alternar entre dos versiones de una misma taxonomía.
- Para quitar un paquete basta con seleccionarlo y hacer clic en el botón **Quitar**.

- Todos estos productos ofrecen funciones de registro de paquetes de taxonomías: XMLSpy, MapForce y StyleVision. La lista de paquetes de taxonomías está disponible en todas estas aplicaciones y si edita la lista en una de ellas, la lista se modificará también en las otras aplicaciones. Si edita la lista de paquetes en una aplicación mientras esté abierta otra aplicación, esta última emitirá una alerta preguntando si desea volver a cargar la lista de paquetes para que se reflejen los cambios realizados en la primera aplicación.

AVISO: los catálogos de paquetes pueden redirigir a recursos incompatibles
Los recursos a los que apunte un archivo de catálogo de un paquete activo se utilizarán para todas las operaciones de XMLSpy que requieran el uso de dichos recursos. Un ejemplo de este tipo de recurso sería un esquema XML, que se utilizará tanto para la validación XML como para la validación XBRL. Si el recurso sin conexión que encuentra el archivo de catálogo del paquete es incompatible con el entorno actual, es posible que se produzcan errores. Cuando esto ocurra, desactive el paquete de taxonomías y póngase en contacto con el autor del paquete, comunicándole la información del mensaje de error.

28.17.15.12.3 XULE

La pestaña XULE (*imagen siguiente*) permite configurar el procesamiento de XULE.

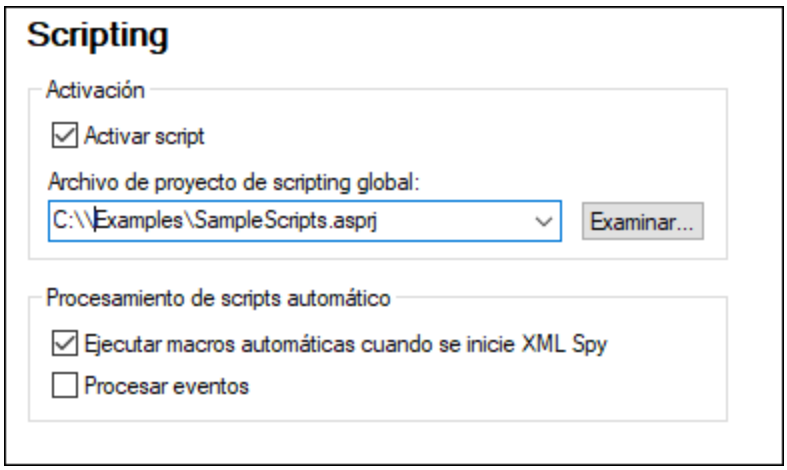


- *Ignorar hechos duplicados:* un hecho duplicado ocurre (habitualmente en Inline XBRL) si se nota más de una vez el mismo hecho en el código HTML. La opción Ignorar duplicados indica que este tipo de hechos deben aparecer una sola vez en el resultado.
- *Resultado:* si se procesa un documento XULE con el comando **XBRL | Ejecutar XULE** el resultado se puede generar en: (i) la ventana *Mensaje* o (ii) un documento nuevo que se muestra en XMLSpy y se almacena temporalmente en la memoria; este documento se puede guardar en un archivo con el comando **Archivo | Guardar como**[1269].

 © 2016-2022 Altova GmbH

28.17.15.13 Scripting

En la sección ***Script*** (*imagen siguiente*) puede habilitar el entorno de scripting 1538 al iniciarse la aplicación. Para ello marque la casilla *Activar script*. Después puede seleccionar el archivo de proyecto de script global (*imagen siguiente*).



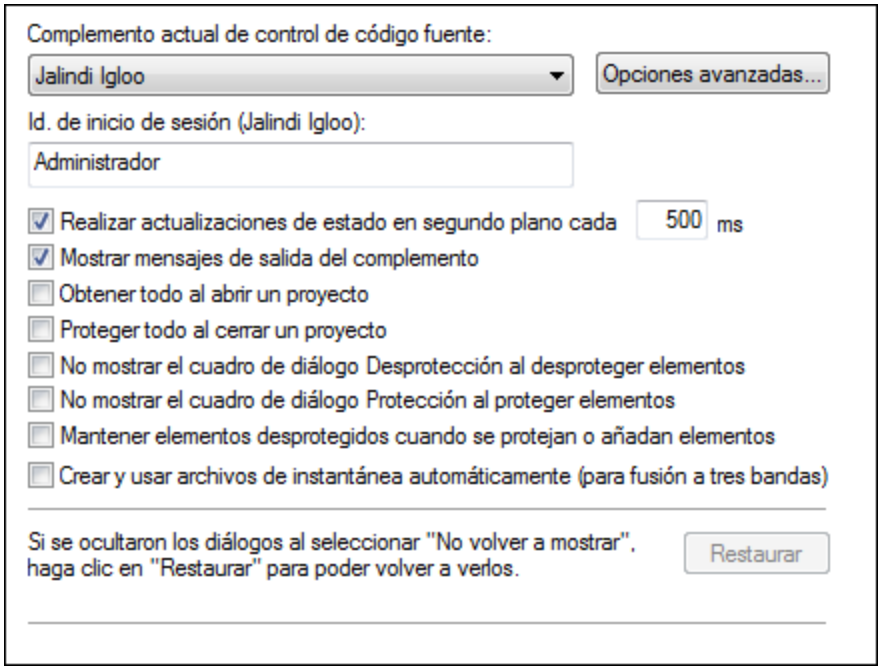
Para definir el proyecto de script global de XMLSpy, marque la casilla *Activar script* y busque el archivo de proyecto de script de Altova (`.asprj`) que desea utilizar. También puede elegir si (i) las macros automáticas del proyecto se ejecutan automáticamente cuando se inicia XMLSpy y (ii) si los controladores de eventos del proyecto de script de la aplicación se ejecutan automáticamente o no.

Guardar y salir

Cuando termine de modificar las opciones de configuración, haga clic en **Aceptar** para terminar. Las macros del proyecto de script global aparecerán en el submenú del comando **Macros**.

28.17.15.14 Control de código fuente

En la sección ***Control de código fuente*** (*imagen siguiente*) puede especificar el proveedor de control de código fuente, configurarlo y definir el id. de inicio de sesión para el proveedor.



Complemento de control de código fuente

Puede elegir un complemento de control de código fuente de los sistemas de control de código fuente que tiene instalados en el equipo. Para ello seleccione una opción de la lista desplegable del cuadro combinado. Después especifique el id. de inicio de sesión. El botón **Opciones avanzadas** abre un cuadro de diálogo del sistema de control de código fuente elegido, donde puede definir opciones de configuración propias de ese control de código fuente concreto.

Preferencias de usuario

En esta pestaña también puede definir las preferencias de usuario:

- Puede realizar actualizaciones de estado en segundo plano pasado un intervalo de tiempo definido por el usuario. También puede desactivar por completo estas actualizaciones. Las bases de datos de control de código fuente de gran tamaño pueden consumir muchos recursos de red y de CPU. Pero si deshabilita las actualizaciones de estado en segundo plano o utiliza un intervalo de tiempo mayor entre actualización y actualización el sistema puede ser más rápido.
- Cuando se abre y se cierra un proyecto, los archivos se pueden desproteger y proteger (respectivamente) de forma automática.
- Puede evitar que aparezcan los cuadros de diálogo Desprotección y Protección.
- El botón **Restaurar** se habilita si marcó/activó la opción *No volver a mostrar* en alguno de los cuadros de diálogo. Al pulsar el botón **Restaurar** se rehabilita el botón/la opción *No volver a mostrar*.

Guardar y salir

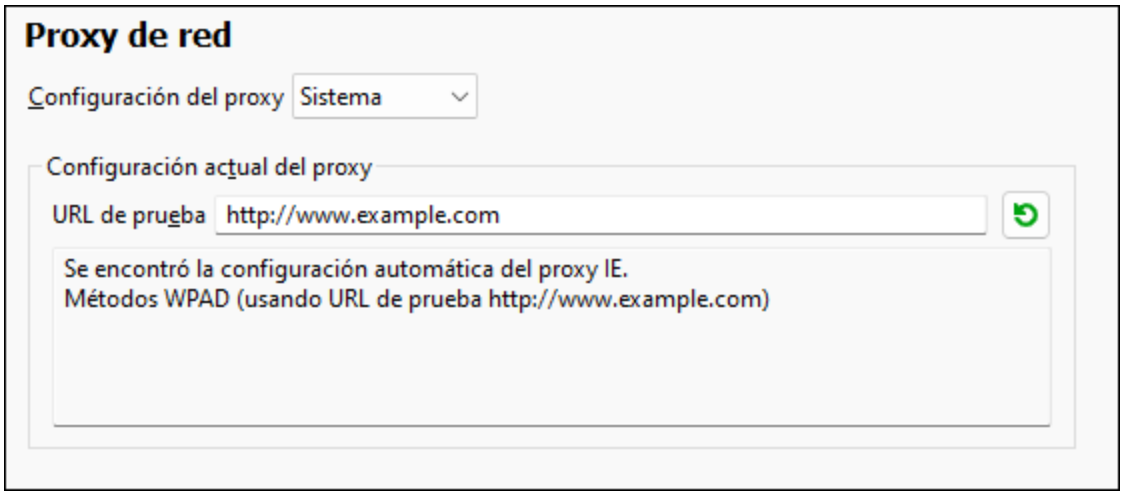
Cuando termine de modificar las opciones de configuración, haga clic en **Aceptar** para terminar.

© 2016-2022 Altova GmbH

28.17.15.15 Proxy de red

El cuadro de diálogo **Proxy de red** (*imagen siguiente*) permite personalizar la configuración del proxy de red. El sistema viene con una configuración predeterminada para el proxy, por lo que este funcionará sin necesidad de configurarlo, pero si quiere usar un proxy de red alternativo puede usar estas opciones para cambiar la configuración como quiera.

Nota: la configuración del proxy de red es común a todas las aplicaciones de Altova MissionKit. En consecuencia, si cambia esta configuración en cualquiera de esas aplicaciones, el cambio afectará automáticamente a todas las demás.



Usar la configuración del proxy del sistema
Usa los parámetros de Internet Explorer (IE), que se pueden configurar desde las opciones del proxy de red. También consulta los parámetros configurados con **`netsh.exe winhttp`**.

Configuración automática del proxy
Existen las siguientes opciones:

- *Configuración de detección automática:* consulta un script WPAD (**`http://wpad.LOCALDOMAIN/wpad.dat`**) vía DHCP o DNS y lo usa para configurar el proxy.
- *URL del script*: indica una HTTP URL a un script (`.pac`) de configuración automática del proxy cuyos parámetros se aplican para configurar el proxy.
- *Volver a cargar*: reinicia y vuelve a cargar la configuración automática actual del proxy. Esta acción requiere Windows 8 o superior y puede llegar a tardar 30 segundos en tener efecto.

Configuración manual del proxy
Puede indicar manualmente el nombre completo de host y el puerto para los proxys de los respectivos protocolos. Es posible que haya un esquema compatible incluido en el nombre de host (por ejemplo: **`http://hostname`**). Si el proxy es compatible no es necesario que el esquema sea el mismo que el protocolo correspondiente.



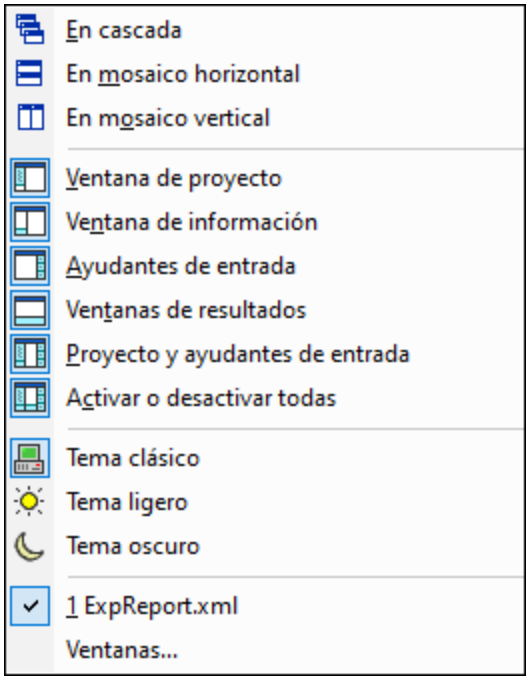
Existen las siguientes opciones:

- *Usar este servidor de proxy para todos los protocolos:* usa el nombre de host y el puerto del Proxy HTTP para todos los protocolos.
- *Ningún proxy para:* muestra una lista de elementos separados por punto y coma (`;`) que pueden ser nombres de host, nombres de dominios o direcciones IP para hosts para los que no hay que usar proxy. Las direcciones IP no se pueden truncar y las direcciones IPv6 deben colocarse entre corchetes (por ejemplo: `[2606:2800:220:1:248:1893:25c8:1946]`). Los nombres de dominio deben empezar por punto (por ejemplo: `.example.com`).
- *No use el servidor proxy para direcciones locales*: si se marca esta opción, se añade el elemento `<local>` a la lista *Ningún proxy para*. Si se selecciona esta opción no se usará proxy para: (i) `127.0.0.1`, (ii) [::1], (iii) todos los nombres de host que no contengan punto (`.`).

© 2016-2022 Altova GmbH

28.18 Menú Ventanas

El menú **Ventanas** contiene comandos estándar que se usan en la mayoría de las aplicaciones Windows y con los que podrá organizar las ventanas de documento de la sesión activa de XMLSpy. Las ventanas que están abiertas se pueden organizar en cascada o en mosaico. Desde este menú también puede activar o desactivar (mostrar/ocultar) las ventanas de los ayudantes de entrada y las diferentes ventanas de documento.



En cascada, En mosaico horizontal, En mosaico vertical

El comando **En cascada** reorganiza todas las ventanas de documento que están abiertas en forma de cascada (es decir, las ventanas se apilan una encima de otra).

Los comandos **En mosaico horizontal** y **En mosaico vertical** reorganiza todas las ventanas de documento que están abiertas y no minimizadas en forma de mosaico, es decir, se pueden ver todas las ventanas a la vez y se distribuyen de forma vertical u horizontal.

Ventanas Proyecto, Información, Ayudantes de entrada y Resultados

Estos comandos sirven para activar/desactivar (mostrar/ocultar) las ventanas Proyecto[122], Información[124], Ayudantes de entrada[124] y Resultados[125].

Estas ventanas son acoplables. Si hace clic en la barra de título de una de ellas y la arrastra, la ventana se desprende de su posición actual y se convierte en una ventana flotante. Haga clic con el botón derecho en la barra de título de la ventana flotante para acoplarla otra vez o para ocultarla.

Proyecto y ayudantes de entrada

Este comando activa/desactiva (muestra/oculta) a la vez la ventana Proyecto[122] y las ventanas de los ayudantes de entrada[124]. De esta forma no tiene que activar/desactivar estas ventanas individualmente.

Activar o desactivar todas

Este comando sirve para activar o desactivar todas las ventanas acoplables:

- la ventana Proyecto[122]
- la ventana Información[124]
- las ventanas de los ayudantes de entrada[124]
- las ventanas de Resultados[125]

Este comando es muy práctico si quiere ocultar rápidamente las ventanas que no sean de documento y conseguir el mayor espacio posible en la interfaz para ver mejor el documento en el que está trabajando.

Temas

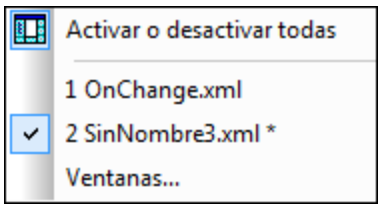
En XMLSpy puede escoger entre tres temas distintos. Cuando seleccione el que prefiere, este se aplica de inmediato.

- Clásico (predeterminado)
- Ligero
- Oscuro

Dentro del tema que esté usando, puede personalizar el formato de los componentes de texto de un tipo de documento. Para ello vaya a las pestañas *Fuentes y colores*[1602] del cuadro de diálogo Opciones (**Herramientas | Opciones**[1583]).

Lista de ventanas abiertas

Esta lista muestra todas las ventanas que están abiertas en cada momento y permite cambiar de una ventana a otra rápidamente.

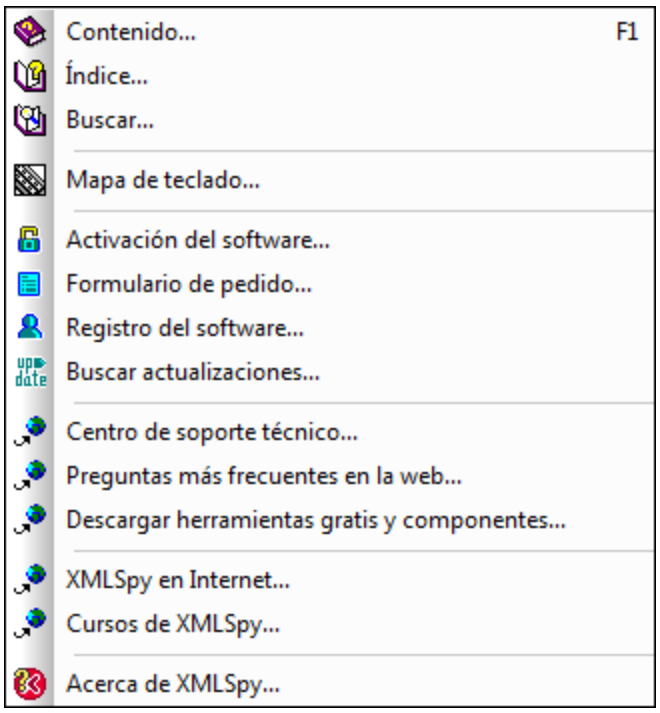


También puede usar las teclas de acceso rápido **Ctrl+F6** para recorrer todas las ventanas que están abiertas.

© 2016-2022 Altova GmbH

28.19 Menú Ayuda

Los comandos del menú **Ayuda** le ayudarán a obtener ayuda o información sobre XMLSpy y entre ellos se encuentran enlaces a páginas del sitio web de Altova.



El menú **Ayuda** también sirve para abrir el cuadro de diálogo Activación del software 1633, donde puede introducir su código clave de licencia cuando haya comprado el producto.

28.19.1 Contenido, Índice, Buscar...

- Contenido

  Abre la ayuda en pantalla de XMLSpy por la tabla de contenido, que aparece en el panel izquierdo de la ventana de ayuda. Esta tabla de contenido ofrece un resumen de la documentación. Haga clic en una entrada de la tabla para abrir la sección correspondiente de la documentación.

- Índice

  Abre la ayuda en pantalla de XMLSpy por el índice de palabras clave, que aparece en el panel izquierdo de la ventana de ayuda. Este índice enumera todas las palabras clave de la ayuda y permite navegar a un tema con solo hacer doble clic en la palabra clave correspondiente. Si una palabra clave está asociada a varios temas, la ventana de ayuda muestra todos estos temas en pantalla.

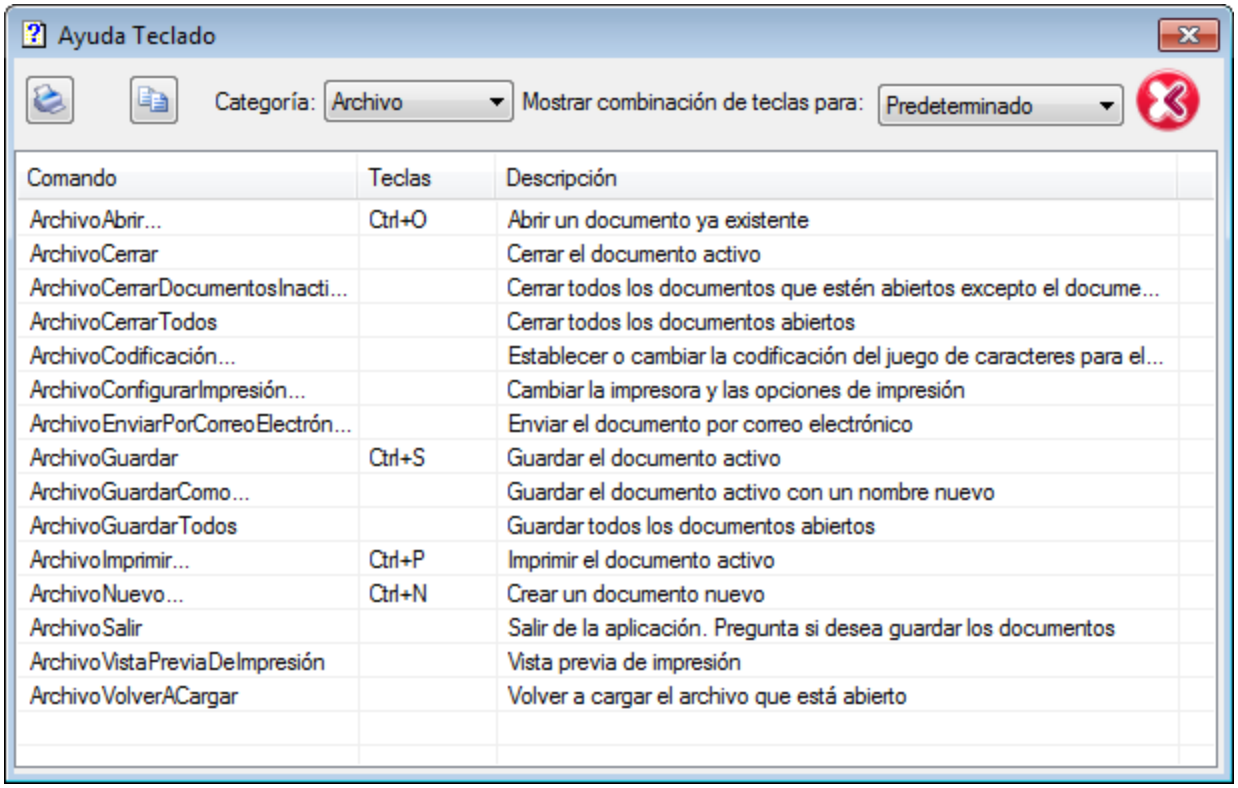
- Buscar

  Abre la ayuda en pantalla de XMLSpy por la función de búsqueda, que aparece en el panel izquierdo de la ventana de ayuda. Para buscar un término introduzca el término de búsqueda en el campo de consulta y

pulse **Entrar**. El sistema de ayuda realiza una búsqueda en toda la documentación y devuelve una lista de resultados. Haga doble clic en una entrada para abrir la sección correspondiente de la documentación.

28.19.2 Mapa de teclado

El comando **Ayuda | Mapa de teclado...** abre una cuadro de diálogo informativo que enumera todos los comandos de todos los menús de XMLSpy. Los comandos de menú van acompañados de una breve descripción y de sus teclas de acceso rápido.



Comando	Teclas	Descripción
ArchivoAbrir...	Ctrl+O	Abrir un documento ya existente
ArchivoCerrar		Cerrar el documento activo
ArchivoCerrarDocumentosInacti...		Cerrar todos los documentos que estén abiertos excepto el docume...
ArchivoCerrarTodos		Cerrar todos los documentos abiertos
ArchivoCodificación...		Establecer o cambiar la codificación del juego de caracteres para el...
ArchivoConfigurarImpresión...		Cambiar la impresora y las opciones de impresión
ArchivoEnviarPorCorreoElectrón...		Enviar el documento por correo electrónico
ArchivoGuardar	Ctrl+S	Guardar el documento activo
ArchivoGuardarComo...		Guardar el documento activo con un nombre nuevo
ArchivoGuardarTodos		Guardar todos los documentos abiertos
ArchivoImprimir...	Ctrl+P	Imprimir el documento activo
ArchivoNuevo...	Ctrl+N	Crear un documento nuevo
ArchivoSalir		Salir de la aplicación. Pregunta si desea guardar los documentos
ArchivoVistaPreviaDeImpresión		Vista previa de impresión
ArchivoVolverACargar		Volver a cargar el archivo que está abierto

Para ver los comandos de un menú determinado, seleccione el nombre del menú en el cuadro combinado *Categoría*. Si quiere imprimir el comando, haga clic en el icono en forma de impresora.

En relación a las teclas de acceso rápido, es necesario recordar que:

- Algunos comandos (y sus teclas de acceso rápido) solamente están disponibles en ciertas vistas. Por ejemplo, la mayoría de los comandos del menú XML solamente están disponibles en la vista Cuadrícula. Otros comandos (como **Archivo | Guardar** o **XML | Comprobar formato XML**, por ejemplo) están disponibles en varias vistas.
- Hay algunas teclas de acceso muy útiles. Por ejemplo, **Mayús+F10** abre el menú contextual en las vistas Texto y Esquema o **Ctrl+E**, que cuando el cursor está dentro de la etiqueta inicial o final de un elemento en la vista Texto, mueve el cursor a la otra etiqueta.
- En la pestaña *Teclado*[1569] del cuadro de diálogo Personalizar puede definir teclas de acceso rápido para los comandos de menú.

© 2016-2022 Altova GmbH

28.19.3 Activación, formulario de pedido, registro, actualizaciones

Activación del software

Tras descargar el producto de software de Altova puede registrarlo o activarlo con una clave de evaluación gratuita o con una clave de licencia permanente.

- ***Licencia gratuita:*** cuando inicie el software por primera vez aparecerá el cuadro de diálogo "Activación del software". Este cuadro de diálogo incluye un botón para solicitar una licencia de evaluación gratuita. Introduzca su nombre, el nombre de su compañía y su dirección de correo electrónico y haga clic en **Enviar solicitud**. Altova le enviará un archivo de licencia al correo electrónico proporcionado. Guarde el archivo en su equipo. Al hacer clic en **Enviar solicitud** aparece un campo en la parte inferior del cuadro de diálogo, que es donde debe introducir la ruta de acceso al archivo de licencia. Puede escribir la ruta de acceso o navegar hasta el archivo de licencia. Por último, haga clic en **Aceptar**. (En el cuadro de diálogo de activación del software también puede hacer clic en **Cargar licencia nueva** para acceder al cuadro de diálogo en el que se introduce la ruta de acceso.) El software permanecerá desbloqueado durante 30 días.
- ***Clave de licencia permanente:*** el cuadro de diálogo "Activación del software" también incluye un botón para comprar una clave de licencia permanente. Este botón conduce a la tienda en línea de Altova, donde podrá adquirir una clave de licencia permanente para el producto. Recibirá por correo electrónico un archivo que contiene los datos de la licencia. Existen tres tipos de licencias permanentes: de tipo instalado, de usuario concurrente y de usuario designado. Las *licencias de tipo instalado* son cada una para un único equipo. En el caso de las *licencias de usuario concurrentes*, si adquiere una de estas licencias para `N` usuarios podrá instalar la licencia en un máximo de `10N` equipos pero solo `N` usuarios podrán usar el software al mismo tiempo. Las *licencias de usuario designado* autorizan a un usuario específico a usar el software en un máximo de 5 equipos distintos. Para activar su software haga clic en **Cargar una licencia nueva** e introduzca la ruta de acceso al archivo de licencia en el cuadro de diálogo "Activación del software" que aparece. Por último, haga clic en **Aceptar**.

Nota: en el caso de licencias para varios usuarios, se le pedirá a cada usuario que introduzca su nombre.

Claves por correo electrónico y las distintas formas de activar las licencias de los productos de Altova
El correo electrónico que recibirá de Altova contiene, en un adjunto, el archivo de la licencia, que tiene la extensión `.altova_licenses`.

Para activar su producto de Altova, puede optar por una de las siguientes opciones:

- Guardar el archivo de licencia (`.altova_licenses`) en su equipo, hacer doble clic en el archivo de licencia, introducir los detalles necesarios en el cuadro de diálogo que aparece y finalmente hacer clic en **Aplicar claves**.
- Guardar el archivo de licencia (`.altova_licenses`) en su equipo. En su producto de Altova seleccione el comando de menú **Ayuda | Activación del software** y después **Cargar una licencia nueva**. Navegue hasta el archivo de licencia o introduzca la ruta de acceso al archivo de licencia en el cuadro de diálogo "Activación del software" y haga clic en **Aceptar**.

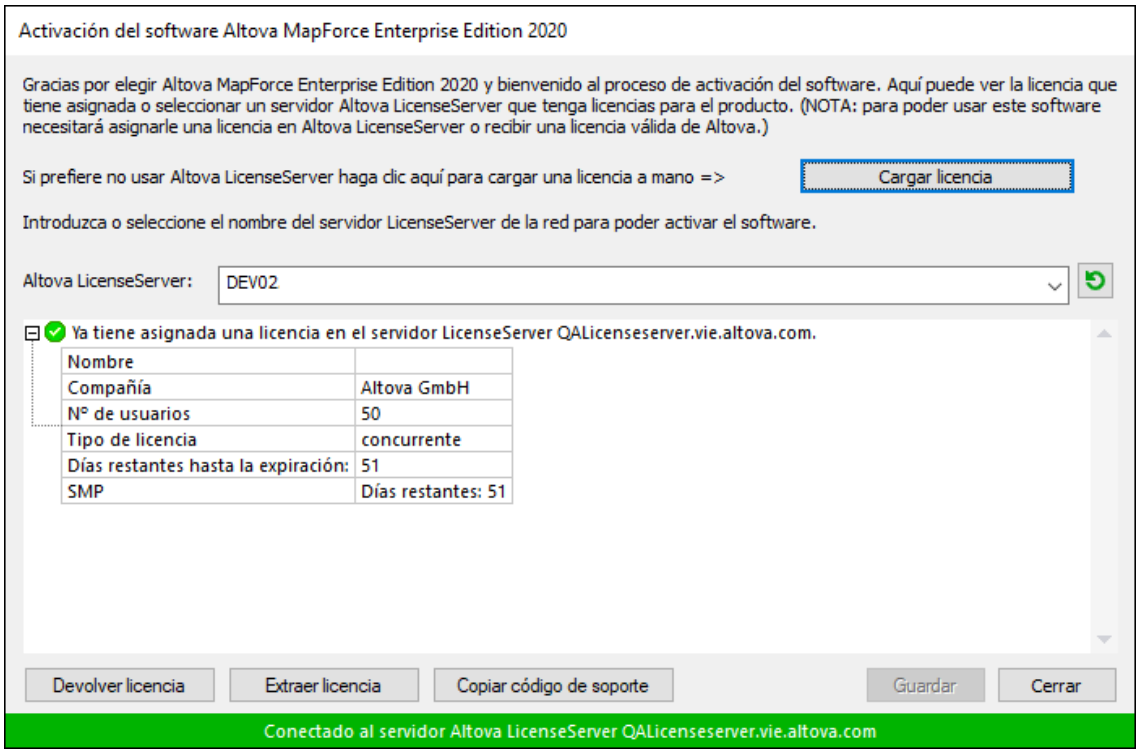
- Guardar el archivo de licencia (`.altova_licenses`) en su equipo y cargarlo desde esa ubicación a su Altova LicenseServer. Puede: (i) adquirir la licencia de su producto Altova con el cuadro de diálogo de activación de software del producto (*véase más abajo*) o (ii) asignar la licencia al producto de Altova LicenseServer. Para obtener más información sobre la gestión de licencias con el LicenseServer, lea el resto de esta sección.

El cuadro de diálogo "Activación del software" (*imagen siguiente*) se abre con el comando **Ayuda | Activación del software**.

Hay dos maneras de activar el software:

- registrando la licencia en el cuadro de diálogo "Activación del software". Para ello haga clic en **Cargar una licencia nueva** y navegue hasta el archivo de la licencia. Haga clic en **Aceptar** para confirmar la ruta de acceso al archivo de licencia y para confirmar los datos que haya introducido (su nombre, en el caso de licencias para más de un usuario); a continuación, haga clic en **Guardar** para finalizar el proceso.
- asignando una licencia a través de un servidor Altova LicenseServer de la red: para adquirir una licencia a través de un servidor Altova LicenseServer de la red haga clic en el botón **Usar Altova LicenseServer**, situado al final del cuadro de diálogo. Seleccione el equipo en el que está instalado el LicenseServer que quiere usar. Tenga en cuenta que la autodetección de los License Servers funciona con emisiones enviadas por LAN. Este tipo de emisiones se limitan a una subred, por lo que Altova License Server debe estar en la misma subred que el equipo del cliente para que funcione la autodetección. Si esta no funciona, introduzca el nombre del servidor. Para ello es necesario que el servidor LicenseServer tenga una licencia para su producto en el repositorio de licencias. Si así es, el cuadro de diálogo "Activación del software" emite un mensaje a tal efecto (*imagen siguiente*). Haga clic en el botón **Guardar** para adquirir la licencia.

© 2016-2022 Altova GmbH



Una vez se ha adquirido una licencia para un equipo específico (es decir, "instalada") del servidor LicenseServer, no se puede devolver al mismo hasta 7 días después. Transcurridos estos 7 días podrá devolver la licencia de ese equipo (con el botón **Devolver licencia**) para que pueda ser adquirida por otro cliente. No obstante, el administrador de LicenseServer puede anular asignaciones de licencias desde la interfaz web del servidor LicenseServer en cualquier momento. Observe que únicamente se pueden devolver las licencias instaladas en equipos específicos, no las licencias concurrentes.

Extracción de licencias

Puede extraer una licencia del repertorio durante un período máximo de 30 días de modo que la licencia se almacene en el equipo donde se ejecuta el producto. Esto le permitirá trabajar sin conexión a Internet, lo cual puede ser útil si desea trabajar en un entorno que no dispone de acceso a su servidor Altova LicenseServer (p. ej. cuando el producto servidor de Altova está instalado en un equipo portátil y el usuario se encuentra de viaje). Mientras la licencia esté extraída, LicenseServer indicará que la licencia está en uso y no podrá ser utilizada por ningún otro equipo. La licencia vuelve automáticamente a su estado insertado cuando finaliza el período de extracción de la licencia. La licencia extraída también se puede insertar en el servidor en cualquier momento con el botón **Insertar** del cuadro de diálogo "Activación del software".

Siga estas instrucciones para extraer una licencia:

1. En el cuadro de diálogo "Activación del software" haga clic en el botón **Extraer licencia** (*imagen anterior*).
2. Aparece el cuadro de diálogo "Extracción de licencias". Seleccione el periodo de extracción y haga clic en **Extraer**.
3. La licencia se extrae y ocurren dos cosas: (i) el cuadro de diálogo "Activación del software" muestra información sobre la extracción de la licencia, incluida la fecha y la

hora en la que expira el plazo de extracción y (ii) en lugar del botón **Extraer licencia**, aparece el botón **Insertar licencia**. Para insertar la licencia basta con hacer clic en este botón. Como la licencia vuelve automáticamente a su estado de inserción cuando finaliza el plazo de extracción, compruebe que el plazo seleccionado coincide con el período de tiempo que tiene pensado trabajar sin conexión a Internet.

Para devolver una licencia esta debe ser de la misma versión principal que el producto de Altova con el que se extrajo. Por tanto, es recomendable devolver la licencia antes de actualizar el producto de Altova correspondiente a la siguiente versión principal.

Nota: para poder extraer licencias esta característica debe estar habilitada en el servidor LicenseServer. Si esta característica no está habilitada, recibirá un mensaje de error a tal efecto cuando trate de extraer una licencia. Cuando esto ocurra, póngase en contacto con el administrador de su servidor LicenseServer.

Copiar código de soporte
Haga clic en **Copiar código de soporte** para copiar los detalles de la licencia en el portapapeles. Esta es la información que deberá introducir al ponerse en contacto con el equipo de soporte técnico a través del formulario de soporte técnico.

Altova LicenseServer es una práctica herramienta para administrar en tiempo real todas las licencias de Altova de la red y ofrece información detallada sobre cada licencia, asignaciones a clientes y uso de las licencias. La ventaja de usar este producto está en sus características administrativas. Altova LicenseServer puede descargarse gratis del sitio web de Altova. Para más información consulte la documentación de Altova LicenseServer.

Formulario de pedido

Hay dos maneras de comprar licencias para los productos de Altova: con el botón **Comprar una licencia permanente** del cuadro de diálogo "Activación del software" (*ver apartado anterior*) o con el comando **Ayuda | Formulario de pedido**, que le lleva directamente a la tienda en línea de Altova, donde puede adquirir claves de licencias para los productos de Altova.

Registro del software

Este comando abre la página de registro de productos de Altova en una pestaña del explorador. Si registra el software, recibirá información sobre actualizaciones y versiones nuevas del producto.

Buscar actualizaciones

Comprueba si existe una versión más reciente del producto en el servidor de Altova y emite un mensaje a tal efecto.

28.19.4 Otros comandos

Centro de soporte técnico

Es un enlace al centro de soporte técnico del sitio web de Altova. El centro de soporte técnico incluye preguntas frecuentes, foros de debate y un formulario para ponerse en contacto con el equipo de soporte técnico de Altova.

© 2016-2022 Altova GmbH

- Preguntas más frecuentes

  Es un enlace a la página de preguntas frecuentes del sitio web de Altova. Esta página se actualiza constantemente con las preguntas que recibimos de nuestros clientes.

- Descargar herramientas gratis y componentes

  Es un enlace al centro de descargas de componentes del sitio web de Altova. Aquí puede descargar software adicional para usarlo con los productos de Altova, como procesadores XSLT y XSL-FO y paquetes de integración. Estos componentes suelen ser totalmente gratis.

- XMLSpy en Internet

  Es un enlace al sitio web de Altova, donde encontrará más información sobre XMLSpy, otros productos de Altova y tecnologías relacionadas.

- Cursos de XMLSpy

  Es un enlace a la página de cursos del sitio web de Altova. Aquí puede seguir todos los cursos sobre productos y tecnologías relacionados con la línea de software de Altova.

- Acerca de XMLSpy

  Abre la pantalla de presentación de la aplicación, que incluye el número de versión del producto e información sobre copyright. Si usa la versión de 64 bits de la aplicación, esto se ve en el nombre de la aplicación, que lleva el sufijo (x64). La versión de 32 bits no lleva ningún sufijo.

28.20 Línea de comandos

Algunas acciones de XMLSpy se pueden llevar acabo desde la línea de comandos. Se trata de estos comandos:

Abrir un archivo

Comando: `xmlspy.exe archivo.xml`
Acción: abre el archivo, `archivo.xml`, en XMLSpy

Nota: si un archivo XML tiene asignado un archivo SPS, el archivo XML se abre en la vista Authentic. De lo contrario, el archivo XML se abre en la vista Texto. Puede asignar un archivo SPS al archivo XML con la marca `/sps` (*véase más abajo*).

Abrir varios archivos

Comando: `xmlspy.exe archivo1.xml archivo2.xml`
Acción: abre los archivos, `archivo1.xml` y `archivo2.xml`, en XMLSpy

Asignar un archivo SPS a un archivo XML para editarlo en la vista Authentic

Comando: `xmlspy.exe mixml.xml /sps misps.sps`
Acción: abre el archivo XML, `mixml.xml`, en la vista Authentic con `misps.sps` como archivo SPS asociado. La marca `/sps` especifica que el archivo SPS que sigue a la marca debe usarse con el archivo XML que precede a la marca `/sps` (para poder editar el archivo XML en la vista Authentic).

Abrir una plantilla XML nueva a través de un archivo SPS

Comando: `xmlspy.exe misps.sps`
Acción: abre un archivo XML nuevo en la vista Authentic. La presentación del archivo se basa en el SPS y el archivo XML nuevo tendrá una estructura inicial basada en el esquema del SPS. Debe asignar un nombre al archivo XML nuevo cuando lo guarde.

Abrir un archivo SPS como documento XML en la vista Texto

Comando: `xmlspy.exe /raw misps.sps`
Acción: abre el archivo `misps.sps` como documento XML en la vista Texto. La marca `/raw` especifica que el archivo SPS que sigue a la marca debe ser editado como archivo XML.

© 2016-2022 Altova GmbH

29 Referencia del programador

XMLSpy es un servidor de automatización: expone objetos programables a otras aplicaciones, que se denominan clientes de automatización. Como resultado los clientes de automatización pueden tener acceso directo a los objetos y a las funciones que pone a su disposición el servidor de automatización. Por ejemplo, los clientes de automatización de XMLSpy pueden usar la función de validación XML de XMLSpy. Por consiguiente, los desarrolladores pueden mejorar sus aplicaciones con las funciones predefinidas de XMLSpy.

Los objetos programables de XMLSpy se ponen a disposición de los clientes de automatización por medio de la API de XMLSpy, que es una API de COM. En adelante usaremos el término "API de la aplicación" para referirnos a la API de XMLSpy. En la presente documentación describimos el modelo de objetos de la API de la aplicación y todos los objetos disponibles (consulte la sección API de la aplicación 1687).

Entornos de ejecución

A la API de la aplicación se puede acceder desde estos entornos:

- El editor de scripts 1641
- Los complementos para entornos IDE 1669
- Programas externos 1687
- La integración con ActiveX 2052

A continuación describimos estos entornos brevemente.

Editor de scripts: personalizar y modificar las funciones de XMLSpy
Puede personalizar su instalación de XMLSpy modificando sus funciones o agregando funciones nuevas. También puede crear formularios y modificar la interfaz del usuario añadiendo comandos de menú e iconos nuevos. Para ello basta con escribir scripts que interactúen con objetos de la API de la aplicación. A fin de ayudarle a realizar estas tareas de forma eficaz, XMLSpy viene con un editor de scripts integrado. Para más información sobre las funciones del editor de scripts e instrucciones de uso, consulte la sección Editor de scripts 1641 de esta referencia. Los lenguajes de programación compatibles con el editor de scripts son **JScript** y **VBScript**.

Complementos para entornos IDE: crear complementos para XMLSpy
En XMLSpy puede crear sus propios complementos e integrarlos en XMLSpy. Para ello puede usar la interfaz especial de XMLSpy para complementos. Para más información sobre cómo crear complementos, consulte la sección Complementos para entornos IDE 1669.

Un objeto de la aplicación se pasa a la mayoría de los métodos que debe implementar el complemento para entornos IDE y la aplicación llama al objeto. Los lenguajes que se suelen utilizar para implementar un complemento para entornos IDE son **C#** y **C++**. Para más información, consulte la sección Complementos para entornos IDE 1669.

Programas externos

También puede manipular XMLSpy con scripts externos. Por ejemplo, puede escribir un script para abrir XMLSpy a una hora concreta, abrir un archivo XML en XMLSpy, validarlo e imprimirlo. Los scripts externos utilizan la API de la aplicación para llevar a cabo estas tareas. Para más información, consulte la sección API de la aplicación 1687.

Para usar la API de la aplicación desde fuera de XMLSpy es necesario iniciar una instancia de XMLSpy primero. La manera de iniciar la instancia de XMLSpy depende del lenguaje de programación utilizado. Para más información consulte el apartado Lenguajes de programación[1688].

Básicamente XMLSpy se inicia por medio de su registro COM. Después se devuelve el objeto `Application` asociado con la instancia de XMLSpy. Dependiendo de la configuración COM, puede devolverse un objeto asociado con una instancia de XMLSpy que ya se esté ejecutando. Puede utilizarse cualquier lenguaje de programación que permita crear e invocar objetos COM. A continuación enumeramos los más comunes:

- Los archivos de script de JScript[1711] y VBScript[1698] tienen una sintaxis sencilla y están diseñados para acceder a objetos COM. Pueden ejecutarse directamente desde la línea de comandos de DOS o al hacer doble clic en ellos desde el explorador de Windows. Son ideales para tareas de automatización sencillas.
- C#[1701] es un lenguaje de programación con un gran número de funciones. El acceso a objetos COM se puede encapsular automáticamente con C#.
- C++ permite un control directo sobre el acceso a COM pero requiere cantidades de código relativamente grandes en comparación con otros lenguajes.
- Java[1711]: los productos de Altova vienen con clases nativas de Java, que encapsulan la API de la aplicación y ofrecen un entorno similar a Java.
- Otros lenguajes de programación que pueden ser útiles son Visual Basic for Applications, Perl y Python.

Integración con ActiveX

Otra manera de acceder a la API de la aplicación es por medio del control ActiveX de XMLSpy. Esta característica sólo está disponible si tiene instalado el paquete de integración de XMLSpy[2052]. Todos los controles ActiveX tienen una propiedad que devuelve un objeto COM equivalente para su función subyacente. El control del administrador ofrece un objeto `Application`, el control de documentos ofrece un objeto `Document` y el objeto marcador de posición, si contiene un árbol de proyecto, devuelve el objeto `Project`. Los métodos compatibles con estos objetos se describen en detalle en el apartado Interfaces de la sección API de la aplicación[1723]. Debe tener cuidado y no utilizar métodos que no tienen sentido en el contexto de la integración para el control ActiveX. Para más información consulte la sección Integración con ActiveX[2052].

Acerca de la referencia del programador

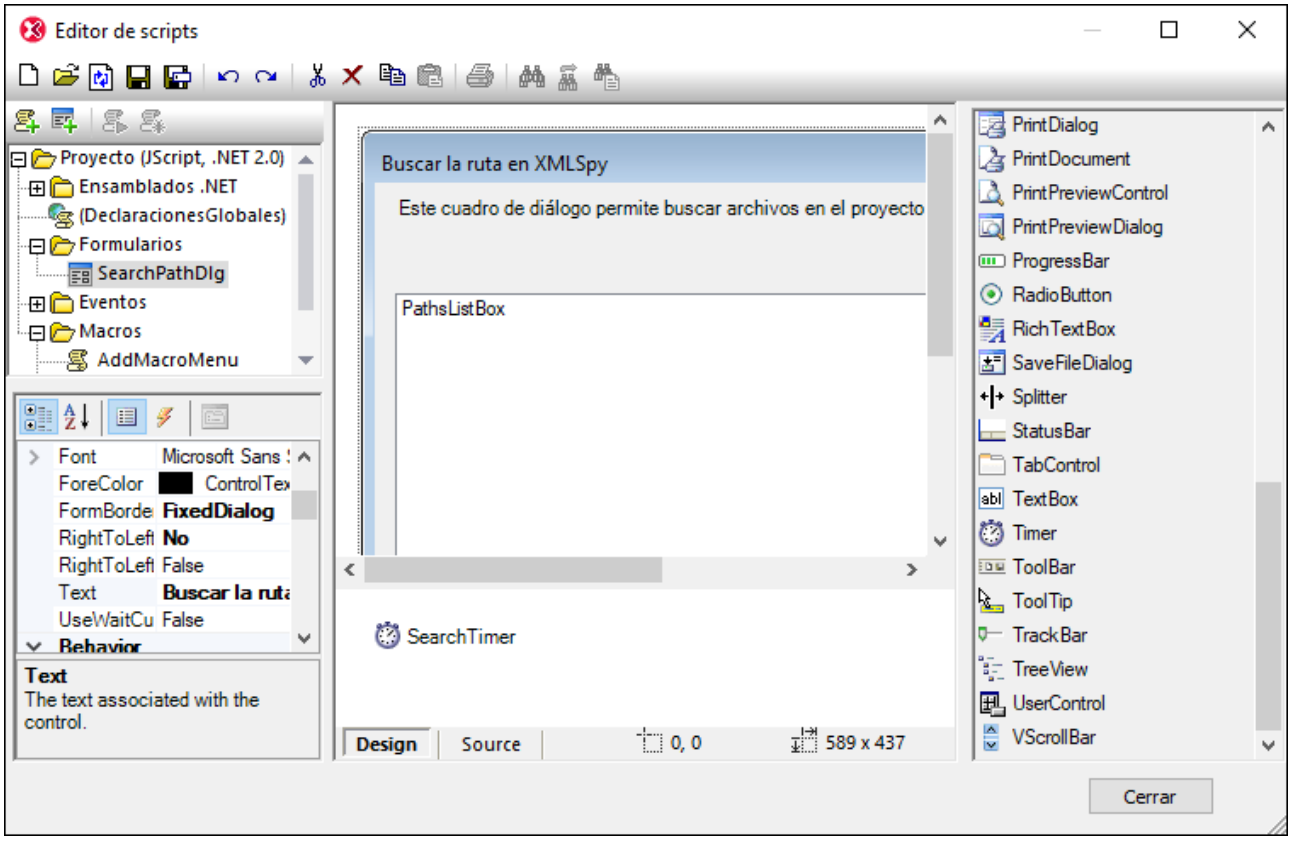
La *Referencia del programador* de la documentación de XMLSpy se divide en estas secciones:

- Editor de scripts[1641]: una referencia del usuario para el entorno de scripting incluido en XMLSpy
- Complementos para entornos IDE[1669]: explica cómo crear complementos para XMLSpy
- API de la aplicación[1687]: una referencia de la API de la aplicación
- Integración para ActiveX[2052]: una guía y referencia para aprender a integrar la IGU y las funciones de XMLSpy usando el control ActiveX

© 2016-2022 Altova GmbH

29.1 Editor de scripts

El Editor de scripts un entorno de desarrollo integrado en XMLSpy desde donde puede personalizar las funciones de XMLSpy con ayuda de los scripts JScript o VBScript. Por ejemplo, puede añadir un elemento nuevo de menú que ejecute una tarea personalizada en el proyecto o puede hacer que XMLSpy desencadene algún comportamiento cada vez que se abra o cierre un documento.Para ello debe crear proyectos de scripting, que son archivos con la extensión `.asprj` (Altova Scripting Project).



Editor de scripts

Los proyectos de scripting suelen incluir una o varias macros, que son programas que ejecutan diversas tareas personalizadas cuando se invocan. Puede ejecutar macros bien de forma explícita desde un elemento de menú (o un botón de la barra de herramientas, si está configurado para ello) o configurar una para que se ejecute automáticamente siempre que se inicie XMLSpy. El entorno de scripting también se puede integrar con la API COM de XMLSpy. Por ejemplo, los scripts que cree con VBScript o JScript pueden gestionar eventos de aplicación o documentos como iniciar o cerrar XMLSpy, abrir o cerrar un proyecto, etc. Los proyectos de scripting pueden incluir formularios de Windows Forms, que puede diseñar de forma visual de forma parecida a como cuando trabaja con Visual Studio. También existen varios comandos integrados de los que se puede ayudar para instanciar y usar clases .NET de código VBScript o JScript.

Una vez haya completado el proyecto de scripting puede habilitarlo de forma global en todo XMLSpy o solamente para proyectos específicos.

El Editor de scripts necesita que instale .NET Framework 2.0 o más avanzado antes de instalar XMLSpy.

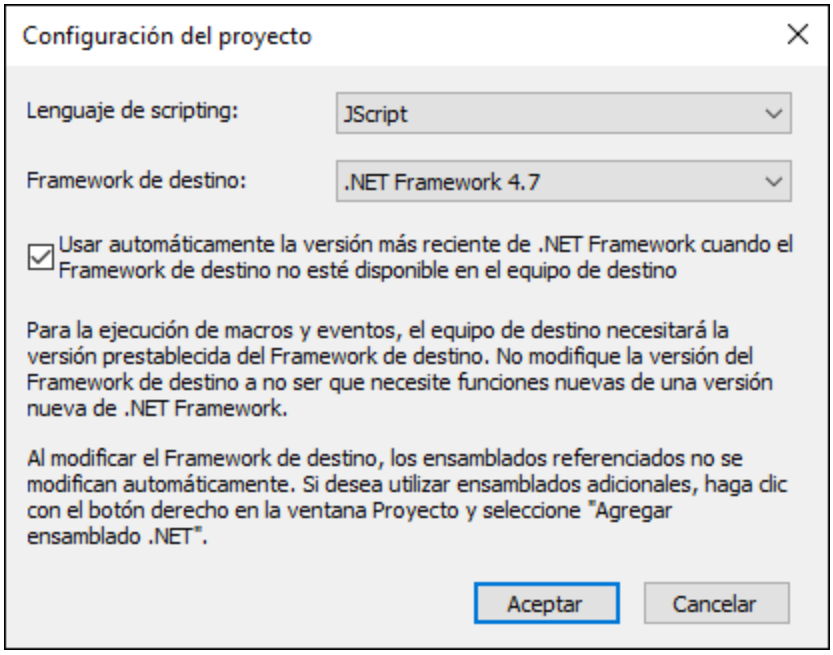
29.1.1 Crear un proyecto de scripting

Todos los scripts, así como la información de scripting que se crean en el Editor de scripts se guardan en los proyectos de scripting de Altova (archivos .asprj). Un proyecto de scripting puede contener macros, controladores de eventos de aplicaciones y formularios (que también pueden tener sus propios controladores de eventos). También puede añadir variables globales y funciones a un script "Declaraciones globales", que hace que esas variables y funciones sean accesibles en todo el proyecto.

Para empezar un proyecto nuevo ejecute el comando **Herramientas | Editor de scripts**.

Los lenguajes que se pueden usar en los proyectos de scripting son JScript y VBScript (no debe confundirse con Visual Basic, que no es compatible). Estos motores de scripting están disponibles por defecto en Windows y no precisan de requisitos especiales para ejecutarse. Para seleccionar el lenguaje de scripting que quiere usar:

1. Haga clic con el botón derecho en el elemento Proyecto, en el panel superior izquierdo y seleccione **Configuración del proyecto** en el menú contextual.
2. Seleccione un lenguaje (JScript o VBScript) y haga clic en **Aceptar**.



Desde el cuadro de diálogo "Configuración el proyecto", en la imagen anterior, también puede cambiar la versión de .NET Framework de destino. Esto suele ser necesario si su proyecto de scripting requiere alguna función que solamente exista en las versiones más recientes de .NET Framework. Tenga en cuenta que cualquier cliente que use su proyecto de scripting tendrá que tener instalada la misma versión de .NET Framework que usted (o una más reciente, siempre que sea compatible).

Por defecto, un proyecto de scripting hace referencia a varios ensamblados .NET, como `System`, `System.Data`, `System.Windows.Forms` entre otros. Si lo necesita puede importar más ensamblados .NET, incluidos

 © 2016-2022 Altova GmbH

ensamblados de caché de ensamblados global de .NET (GAC por sus siglas en inglés) o archivos .dll personalizados. Puede importar ensamblados:

1. De forma estática, añadiéndolos manualmente al proyecto. Haga clic con el botón derecho en el panel superior izquierdo y seleccione **Agregar ensamblado .NET** en el menú contextual.
2. De forma dinámica, en tiempo de ejecución, llamando al comando `CLR.LoadAssembly`[1659] desde el código.

Puede crear varios proyectos de scripting y guardar uno en disco para después volver a cargarlo más tarde en el Editor de scripts. Para ello use los botones estándar de Windows de la barra de herramientas: **Nuevo**, **Abrir**, **Guardar**, **Guardar como**. Una vez haya probado el proyecto y esté listo para implementarse puede cargarlo en XMLSpy y ejecutar cualquiera de sus macros o controladores de eventos. Para más información consulte el apartado Habilitar scripts y macros[1666].

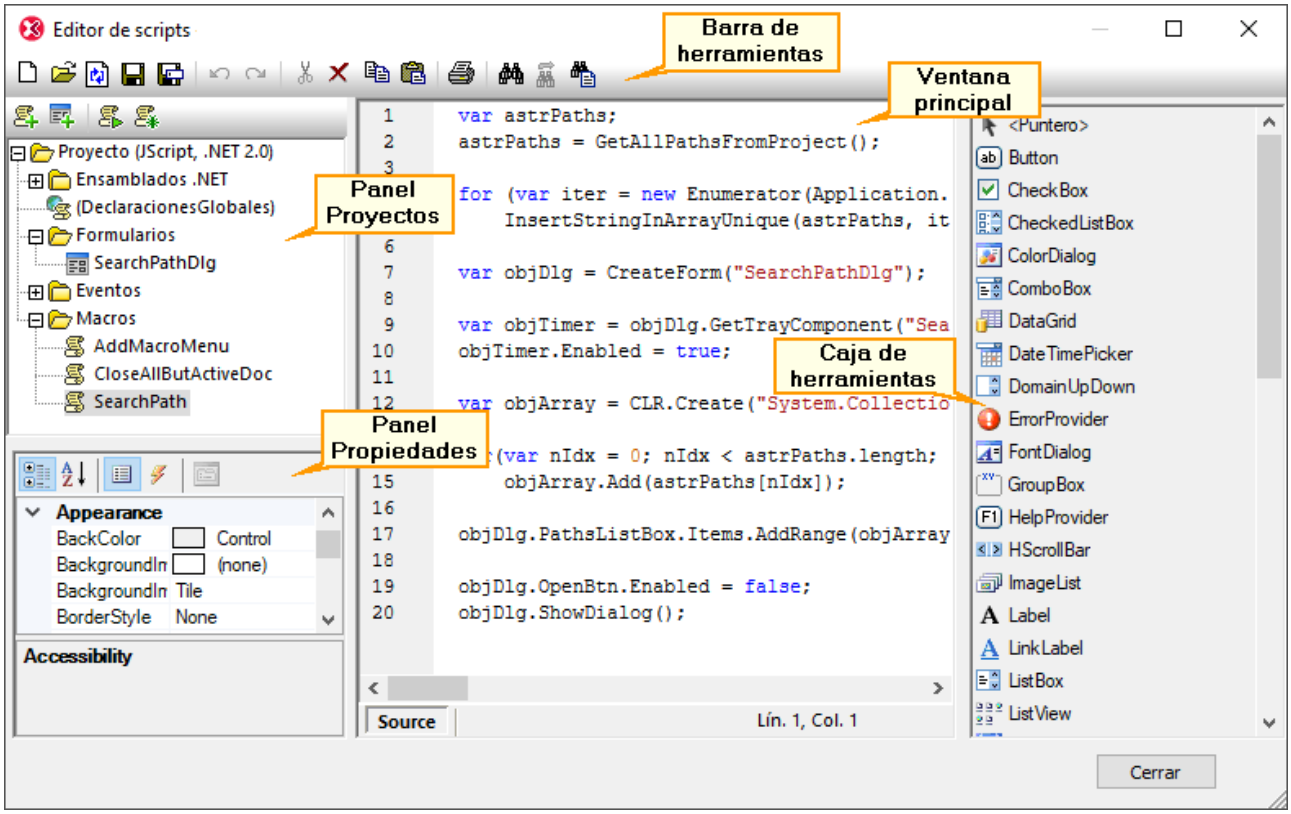
También puede encontrar un proyecto de scripting de ejemplo en: **C:\Usuarios\<user>\Documentos\Altova\XMLSpy2023\Examples\SampleScripts.asprj**.

Las secciones siguientes se centran en las partes de un proyecto de scripting: declaraciones globales, macros, formularios y eventos.

29.1.1.1 Información general

El Editor de scripts contiene estas partes:

- Barra de herramientas
- Panel Proyectos
- Panel Propiedades
- Ventana principal
- Cuadro de herramientas



Barra de herramientas

La barra de herramientas incluye comandos estándar de Windows para la gestión de archivos (**Nuevo**, **Abrir**, **Guardar**, **Guardar como**) y comandos de edición (**Copiar**, **Cortar**, **Eliminar**, **Pegar**). Al editar código fuente también se activan los comandos **Buscar** y **Reemplazar**, así como el comando **Imprimir**.

Panel Proyectos

En el panel Proyectos puede ver y gestionar la estructura del proyecto. Un proyecto de scripting consiste en varios componentes que pueden funcionar juntos y que no tienen que crearse en un orden determinado:

- *Un script "Declaraciones globales"*. Como el nombre indica, este script contiene información que se usa en todo el proyecto. En este script puede declarar cualquier variable o función que necesite tener disponible para todos los formularios, scripts de gestión de eventos y macros.
- *Formularios*. Los formularios suelen ser necesarios para recopilar información de los usuarios o para suministrar cuadros de diálogo informativos. Un formulario se invoca llamándolo sea con una función (en el script *Declaraciones globales*) o directamente con una macro.
- *Eventos*. La carpeta "Eventos" muestra eventos de la aplicación XMLSpy suministrados por la API COM. Para escribir un script que se ejecute cuando ocurra un evento haga doble clic en cualquier evento y después teclee el código de manejo en el editor. No debe confundir los eventos de la aplicación con los eventos de los formularios; estos últimos se manejan a nivel del formulario, como detallamos más adelante.
- *Macros*. Una macro es un script que puede invocarse a petición desde un menú contextual o ejecutarse automáticamente cuando empiece XMLSpy. Las macros no tienen parámetros o valores de

© 2016-2022 Altova GmbH

retorno. Una macro puede acceder a todas las variables y funciones declaradas en el script *Declaraciones globales* y también puede mostrar formularios.

Haga clic con el botón derecho en cualquiera de los componentes para ver los comandos del menú contextual y los atajos de teclado correspondientes. Haga doble clic en cualquier archivo (como un formulario o un script) para abrirlo en la ventana principal.

Con los botones de la barra de herramientas puede acceder a estos comandos rápidos:

	**Macro nueva**	Agrega una macro nueva al proyecto en el directorio **Macros**.
	**Formulario nuevo**	Agrega un formulario nuevo al proyecto en el directorio **Formularios**.
	**Ejecutar macro**	Ejecuta la macro seleccionada.
	**Depurar macro**	Ejecuta la macro seleccionada en modo depuración.

Panel Propiedades

El panel Propiedades es muy parecido al de Visual Studio y contiene:

- Propiedades del formulario, si se selecciona uno
- Propiedades del objeto, si se selecciona uno
- Eventos del formulario, si se selecciona uno
- Eventos del objeto, si se selecciona uno

Para alternar entre las propiedades y los eventos del componente seleccionado haga clic en los botones Propiedades  o Eventos .

Los iconos **Por categorías**  y **Alfabético**  muestran las propiedades o los eventos organizados por categorías o por orden alfabético ascendente.

Cuando se selecciona una propiedad o un evento aparece una breve descripción de ese elemento en la parte inferior del panel Propiedades.

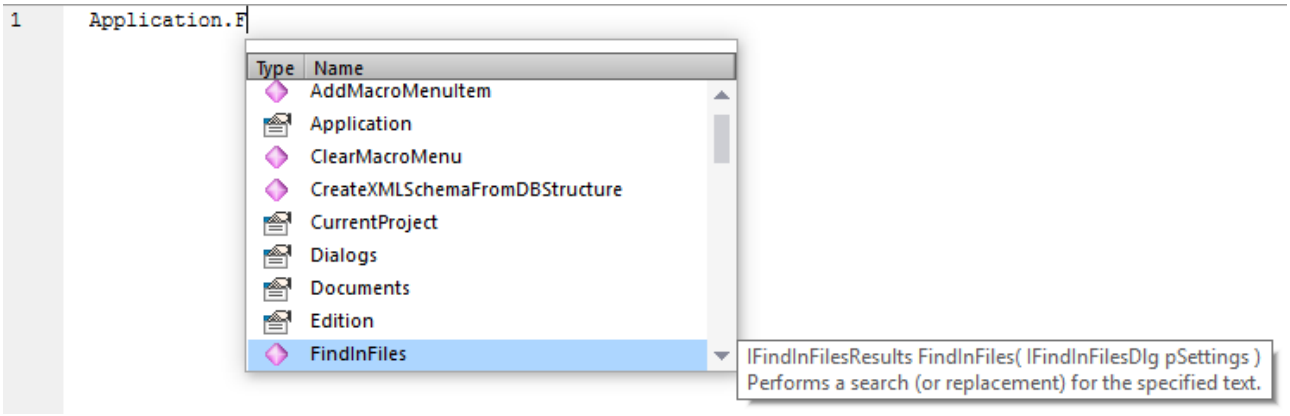
Ventana principal

La ventana principal es el área de trabajo donde puede introducir código fuente o modificar el diseño del formulario. Al editar formularios puede trabajar en dos pestañas: *Design* y *Source*. La pestaña *Design* muestra el diseño del formulario, mientras que la pestaña *Source* contiene el código fuente, como métodos de control de eventos.

El editor de código fuente cuenta con ayudas para la edición de código, como color de sintaxis, plegamiento de código, resaltado de los corchetes de inicio y final, zoom, sugerencias de finalización automática y marcadores.

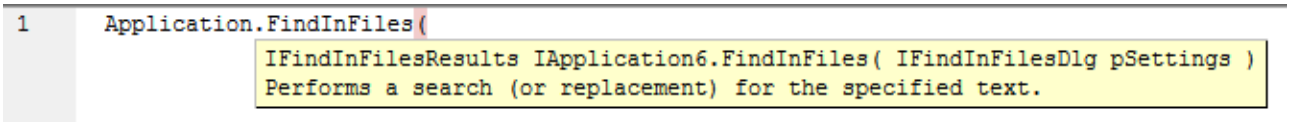
Sugerencias de finalización automática

JScript y VBScript son dos lenguajes sin tipo, por lo que la finalización automática se limita a los nombres de la API COM y a los comandos 1656 integrados de XMLSpy. El método completo o la firma de la propiedad aparecen junto al ayudante de entrada de finalización automática.



Si los nombres empiezan por `objDocument`, `objProject`, `objXMLData`, or `objAuthenticRange`, entonces se mostrarán los miembros de la interfaz correspondiente.

Para ver la firma (y la documentación, si la hay) de un método o una propiedad conocidos, coloque el cursor del ratón sobre el elemento en cuestión, por ejemplo:



El ayudante de entrada de finalización automática suele aparecer automáticamente al editar, pero también puede activarlo pulsando **Ctrl+Barra espaciadora**.

Marcadores

- Para guardar o eliminar un marcador, haga clic dentro de una línea y después pulse **Ctrl+F2**
- Para ir al marcador siguiente pulse **F2**
- Para ir al marcador anterior pulse **Mayús+F2**
- Para eliminar todos los marcadores pulse **Ctrl+Mayús+F2**

Alejarse y acercarse con el zoom

- Para alejarse y acercarse con el zoom mantenga pulsada la tecla Ctrl y pulse las teclas "+" o "-", o gire la rueda del ratón.

Configurar la vista Texto

Para activar estas opciones haga clic con el botón derecho en el editor y seleccione **Configuración de la vista Texto** en el menú contextual.

Fuentes

Para cambiar la fuente haga clic con el botón derecho en el editor y seleccione **Fuentes de la vista Texto...** en el menú contextual.

Caja de herramientas

La Caja de herramientas contiene todos los objetos que hay disponibles para diseñar formularios, como son los botones, las cajas de texto, los cuadros combinados, etc.

© 2016-2022 Altova GmbH

Para añadir un elemento de la Caja de herramientas a un formulario:

1. Cree o abra un formulario y seleccione la pestaña **Diseño**.
2. Haga clic en el objeto de la caja de herramientas (por ejemplo, **Botón**) y después haga clic en el lugar del formulario donde lo quiere insertar. También puede arrastrar el objeto directamente hasta el formulario.

Algunos objetos, como el `Temporizador`, no se añaden al formulario, sino que se crean en una bandeja, en la parte inferior de la ventana principal. Puede seleccionar el objeto en la bandeja y configurar las propiedades y los controladores de eventos para el objeto del panel Propiedades. Para ver un ejemplo consulte el apartado Control de eventos de formularios 1649.

También puede añadir controles ActiveX registrados al formulario. Para ello haga clic con el botón derecho en el área de la Caja de herramientas y seleccione **Agregar control ActiveX** en el menú contextual.

29.1.1.2 Declaraciones globales

El script "Declaraciones globales" existe por defecto en todos los proyectos de scripting, por lo que no lo tiene que crear explícitamente. Cualquier variable o función que quiera añadir a este script se considera como global a todo el proyecto. Esto significa que puede hacer referencia a esas variables y funciones desde cualquier macro o evento del proyecto. A continuación se muestra un extracto de ejemplo del script Declaraciones globales que importa el espacio de nombres `System.Windows.Forms` en el proyecto. Para ello el código invoca el comando `CLR.Import` que está integrado en el Editor de scripts.

```
// importa el espacio de nombres System.Windows.Forms para todas las macros, los
formularios y eventos:
CLR.Import( "System.Windows.Forms" );
```

Nota: cada vez que se ejecuta una macro o que se llama a un controlador de eventos las declaraciones globales se vuelven a inicializar.

29.1.1.3 Macros

Las macros son scripts que contienen declaraciones JScript (o VBScript, en función del lenguaje del proyecto), como declaraciones y funciones variables.

Puede añadir macros a su proyecto si quiere; para ello haga clic con el botón derecho en el panel Proyectos, seleccione **Agregar macro** en el menú contextual y después introduzca el código de la macro en el formulario principal. Ese código puede ser tan simple como una alerta, por ejemplo:

```
alert("Hola, soy una macro");
```

Una macro más avanzada podría contener variables y funciones locales. Las macros también pueden contener código que invoque formularios desde el proyecto. El extracto siguiente representa una macro de ejemplo que muestra un formulario. Se asume que el formulario ya se ha creado en la carpeta "Formularios" y que se le ha dado el nombre "FormularioEjemplo" (consulte también el apartado Formularios 1648).

```
// muestra un formulario
ShowForm( "SampleForm" );
```

En el extracto anterior `ShowForm` es un comando integrado en el Editor de scripts. Para ver más comandos que puede usar para trabajar con formularios y objetos .NET consulte el apartado Comandos integrados 1656.

Puede agregar varias macros al mismo proyecto y puede convertir cualquier macro en "automática". Al hacerlo, la macro se ejecuta automáticamente cuando se inicia XMLSpy. Para definir una macro como automática haga clic con el botón derecho en ella y seleccione **Establecer como macro automática** en el menú contextual.

Solo se puede ejecutar una macro a la vez. Una vez se ha ejecutar la macro (o el evento) el script se cierra y las variables globales pierden sus valores.

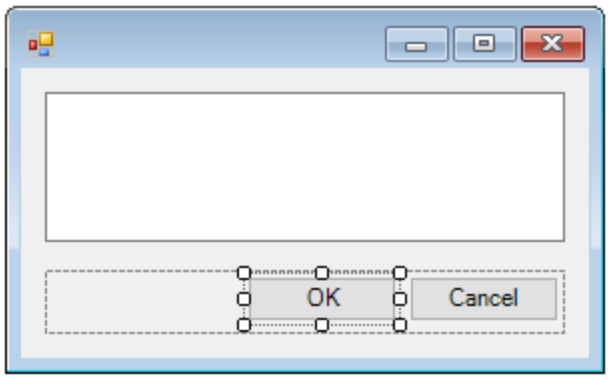
Para ejecutar una macro directamente en el Editor de scripts haga clic en Ejecutar macro . Para depurar una macro con el depurador de Visual Studio haga clic en Depurar macro . Para más información sobre cómo habilitar y ejecutar macros en XMLSpy, consulte el apartado Habilitar scripts y macros 1666.

29.1.1.4 Formularios

Los formularios son especialmente prácticos si necesita recolectar datos de entrada de usuarios o mostrar datos a usuarios. Para llevar a cabo esas acciones un formulario puede contener varios controles como botones, casillas de verificación, cuadros combinados, etc.

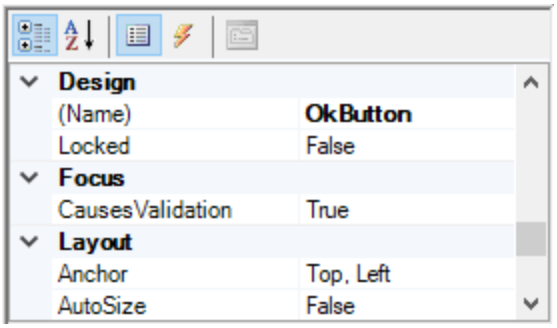
Para añadir un formulario haga clic con el botón derecho en el panel Proyecto y seleccione **Agregar formulario** en el menú contextual. Para añadir un control a un formulario arrástrelo desde la Caja de herramientas que está a la derecha del Editor de scripts y suéltela en el formulario.

Puede cambiar la posición y el tamaño de los controles directamente en el formulario usando las manijas que aparecen al hacer clic en un control, por ejemplo:



Todos los controles de los formularios tienen propiedades que puede ajustar en el panel Propiedades. Para ello, seleccione primero el control en el formulario y después edite las propiedades en cuestión en el panel Propiedades.

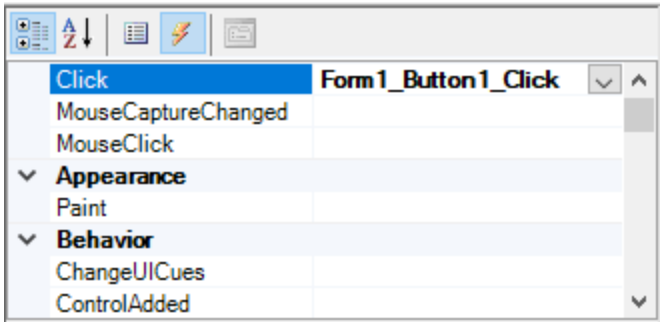
 © 2016-2022 Altova GmbH



Control de eventos de formularios

Cada control del formulario expone también varios eventos a los que puede vincular el proyecto de scripting. Por ejemplo, puede que quiera invocar algún método de la API COM de XMLSpy cuando se haga clic en un botón. Para crear una función que vincule a un evento de formulario siga estos pasos:

1. En el panel Propiedades haga clic en **Eventos** .
2. En la columna Acción haga doble clic en el evento donde necesita el método (por ejemplo, en la imagen siguiente, el evento de controlador es "Click").



También puede añadir métodos de controlador haciendo doble clic en un control del formulario. Por ejemplo, al hacer doble clic en un botón del diseño del formulario se genera un método de controlador para el evento "Click" de ese botón.

Una vez se ha generado el cuerpo del método del controlador puede teclear el código que controla este evento, por ejemplo:

```
//Ocurre cuando se hace clic en el componente.
function MyForm_ButtonClick( objSender, e_EventArgs )
{
   alert("A button was clicked");
}
```

Para mostrar un formulario sin terminar fuera del Editor de scripts haga clic con el botón derecho en el formulario y seleccione **Probar el formulario** en el menú contextual. Observe que el comando **Probar el formulario** solamente muestra el formulario; los eventos del formulario (como hacer clic en un botón) siguen

estando deshabilitados. Para que el formulario reaccione a los eventos debe llamarlo desde una macro, por ejemplo:

```
// Instanciar y mostrar un formulario
ShowForm( "SampleForm" );
```

Acceder a los controles del formulario

Puede acceder a cualquier componente de un formulario desde el código usando sintaxis de acceso a los campos. Por ejemplo, imagine que tiene un formulario con este diseño:

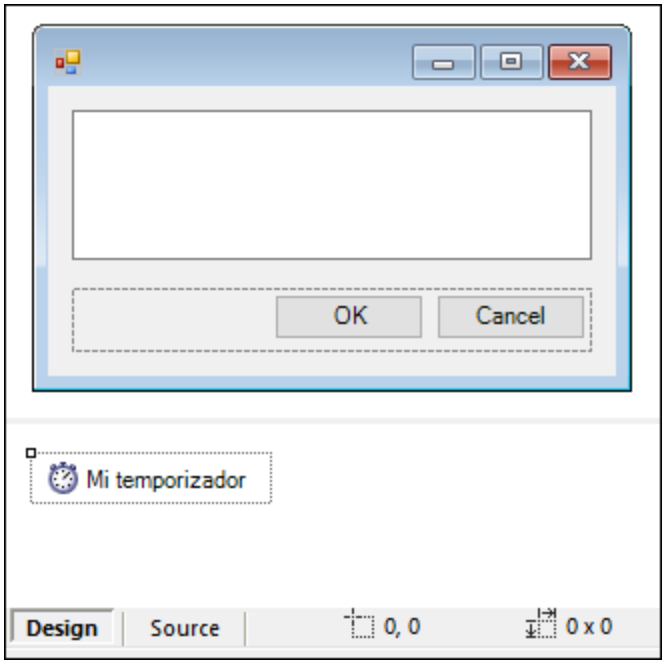
```
// MiFormulario
//   PanelBotón
//      BotónAceptar
//      BotónCancelar
//   EditorTexto
//      ReproductorMediaAx1
// ComponentesBandeja
//   MiTemporizador
```

El código siguiente muestra cómo instanciar el formulario, acceder a algunos de sus controles con la sintaxis de acceso a los campos y después mostrarlo:

```
// Instanciar el formulario
var objForm = CreateForm("MyForm");
// Deshabilitar el botón Aceptar
objForm.ButtonPanel.OkButton.Enabled = false;
// Cambiar el texto del Editor de texto
objForm.TextEditor.Text = "Hello";
// Mostrar el formulario
objForm.ShowDialog();
```

Al añadir al formulario ciertos controles, como los temporizadores, estos no aparecen en el formulario, sino que se muestran como componentes de la bandeja en la parte inferior del diseño del formulario, por ejemplo:

© 2016-2022 Altova GmbH



Para acceder a los controles de la bandeja puede usar el método `GetTrayComponent` en el objeto formulario y suministrar el nombre del control como argumento. En este ejemplo, para obtener una referencia a `MiTemporizador` y habilitarlo debe usar este código:

```
var objTimer = objForm.GetTrayComponent("MiTemporizador");
objTimer.Enabled = true;
```

En el caso de los controles ActiveX puede acceder al objeto COM subyacente con la propiedad OCX:

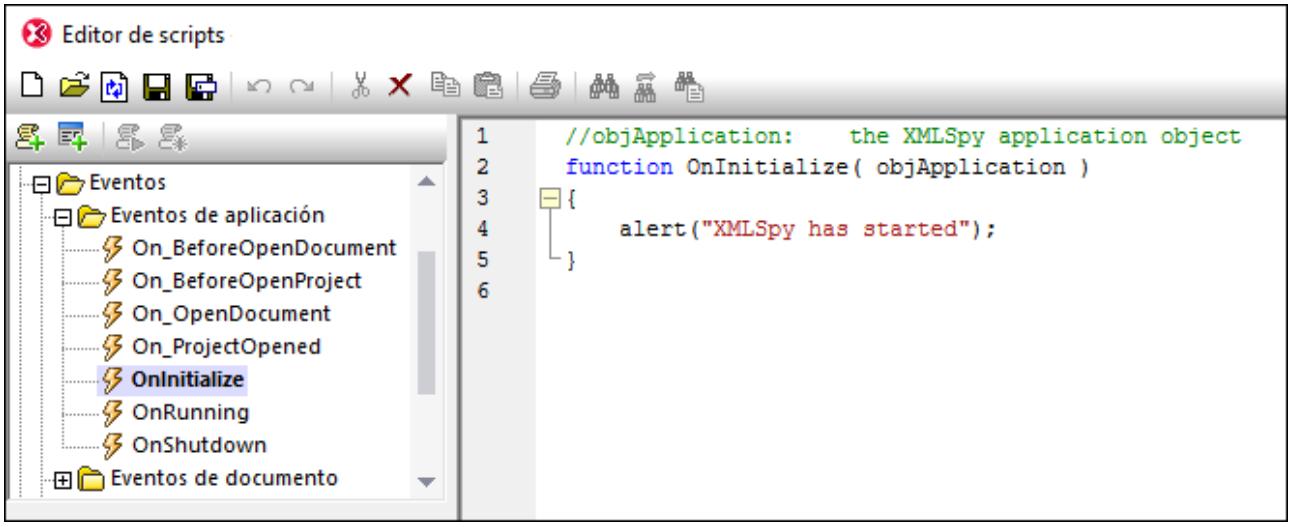
```
var ocx = lastform.AxMediaPlayer1.OCX; // obtener objeto COM subyacente
ocx.enableContextMenu = true;
ocx.URL = "mms://apasf.apa.at/fm4_live_worldwide";
```

29.1.1.5 Eventos

Un proyecto de scripting puede incluir scripts que manejen eventos de XMLSpy como abrir, cerrar o guardar un documento, iniciar o cerrar XMLSpy, añadir un elemento a un diagrama, etc. Estos eventos los suministra la API COM de XMLSpy y los puede encontrar en la carpeta "Eventos" del proyecto de scripting. Tenga en cuenta que estos eventos son eventos específicos de XMLSpy, en contraposición a los eventos de formulario. Los eventos están organizados en carpetas:

- Eventos de aplicación
- Eventos de documento
- Eventos de la vista Authentic
- Eventos de la vista Cuadrícula
- Eventos de la vista Texto

Para crear un script de controlador de eventos haga clic con el botón derecho en un evento y seleccione **Abrir** en el menú contextual (o haga doble clic en el evento). El script del controlador de eventos aparece en la ventana principal, donde puede editarlo. Por ejemplo, el controlador de eventos de la imagen siguiente muestra una alerta cada vez que se inicia XMLSpy:



Tenga en cuenta que:

- El comando `alert` se puede aplicar a JScript. El equivalente en VBScript es `MsgBox`. Véase también alert 1657.
- No debe cambiar el nombre de la función del controlador de eventos; de lo contrario no se llamará al script del controlador de eventos.
- Para que se procesen eventos debe marcar la casilla *Procesar eventos* al habilitar el proyecto de scripting en XMLSpy. Para más información consulte el apartado Habilitar scripts y macros 1666.

También puede definir variables locales y funciones de ayuda dentro del script del controlador, por ejemplo:

```
var local;

function OnInitialize( objApplication )
{
   local = "OnInitialize";
   Helper();
}

function Helper()
{
   alert("I'm a helper function for " + local);
}
```

29.1.1.6 Consejos para programar con JScript

A continuación ofrecemos algunos consejos para programar con JScript que le ayudarán a la hora de desarrollar un proyecto de scripting en el Editor de scripts de XMLSpy.

© 2016-2022 Altova GmbH

Parámetros out

Los parámetros out de los métodos de NET Framework requieren variables especiales en JScript. Por ejemplo:

```
var dictionary =
CLR.Create("System.Collections.Generic.Dictionary<System.String,System.String>");
dictionary.Add("1", "A");
dictionary.Add("2", "B");

// use el método de JScript para acceder a los parámetros out
var strOut = new Array(1);
if ( dictionary.TryGetValue("1", strOut) ) // TryGetValue establece el parámetro out
  alert( strOut[0] ); // use out parameter
```

Números enteros como argumentos

Los métodos .NET que requieren números enteros como argumentos no se deben llamar directamente con los objetos numéricos de JScript, que son valores de puntos flotantes. Por ejemplo, en lugar de:

```
var objCustomColor = CLR.Static("System.Drawing.Color").FromArgb(128,128,128);
```

use:

```
var objCustomColor =
CLR.Static("System.Drawing.Color").FromArgb(Math.floor(128),Math.floor(128),Math.floor(12
8));
```

Iterar colecciones .NET

Para iterar colecciones .NET se pueden usar el Enumerador de JScript y el iterador .NET. Por ejemplo:

```
// iterar con el iterador de JScript
var itr = new Enumerator( coll );
for ( ; !itr.atEnd(); itr.moveNext() )
  alert( itr.item() );

// iterar con el iterador .NET
var itrNET = coll.GetEnumerator();
while( itrNET.MoveNext() )
  alert( itrNET.Current );
```

Plantillas .NET

Las plantillas .NET se pueden instanciar así:

```
var coll = CLR.Create( "System.Collections.Generic.List<System.String>" );
```

o así

```
CLR.Import( "System" );
CLR.Import( "System.Collections.Generic" );
var dictionary = CLR.Create( "Dictionary<String,Dictionary<String,String>>" );
```

Valores .NET de enumeración

Para acceder a los valores .NET de enumeración puede usar:

```
var enumValStretch = CLR.Static( "System.Windows.Forms.ImageLayout" ).Stretch;
```

Literales de enumeración

Para acceder a los literales de enumeración de la API de XMLSpy (no necesita conocer su valor numérico) puede usar:

```
objExportXMIFileDlg.XMIType = eXMI21ForUML23;
```

29.1.1.7 Ejemplo: proyecto de scripting

XMLSpy viene con un proyecto de scripting de ejemplo que puede encontrar aquí: **C:\Users\<user>\Documents\Altova\XMLSpy2023\Examples\SampleScripts.asprj**. Este proyecto consiste en unas pocas macros y un formulario de Windows.

Para cargar el proyecto de scripting en el Editor de scripts:

1. En el menú **Herramientas** haga clic en **Editor de scripts**.
2. Haga clic en **Abrir** y navegue hasta el archivo **SampleScripts.asprj** desde la ruta indicada más arriba.

El proyecto contiene vaciar macros en el directorio "Macros".

Macro	Descripción
AddMacroMenu	Esta macro añade un elemento de menú nuevo a XMLSpy invocando el método `Application.AddMacroMenuItem` de la API COM. El primer argumento del método `Application.AddMacroMenuItem` es el nombre de la macro que se quiere añadir (en este ejemplo, "CloseAllButActiveDoc") y el segundo argumento es el texto que aparece con ese elemento del menú. Siempre que se ejecute esta macro se añade un comando de menú nuevo llamado "CloseAllButActiveDoc" al menú **Herramientas**. Para eliminar los elementos de menú de la macro creados anteriormente puede reiniciar XMLSpy o crear una macro que llame al método de la API `Application.ClearMacroMenu`.

© 2016-2022 Altova GmbH

CloseAllButActiveDocument	Cuando se ejecuta, esta macro itera por todos los documentos abiertos en XMLSpy y los cierra todos, excepto el que esté activo.
SearchPath	Esta macro muestra un formulario que permite a los usuarios buscar archivos dentro del proyecto actual. El formulario está en el directorio "Formularios",, donde puede ver el diseño y los controladores de eventos asociados. El método `GetAllPathsFromProject()` devuelve como matriz todas las rutas de acceso a archivos que pertenecen al proyecto abierto en ese momento. La definición de este método está en el script **GlobalDeclarations** del proyecto. El método `InsertStringInArrayUnique` controla que solamente se añadan rutas de acceso únicas a la matriz. A continuación, el formulario se inicializa con CreateForm 1662. Por último, la matriz se convierte en un tipo .NET con el método CreateForm 1662. y el formulario se rellena con la colección `ArrayList` resultante. El botón Abrir del formulario tiene un controlador que llama al método `Application.Documents.OpenFile` de la API para abrir el archivo seleccionado.

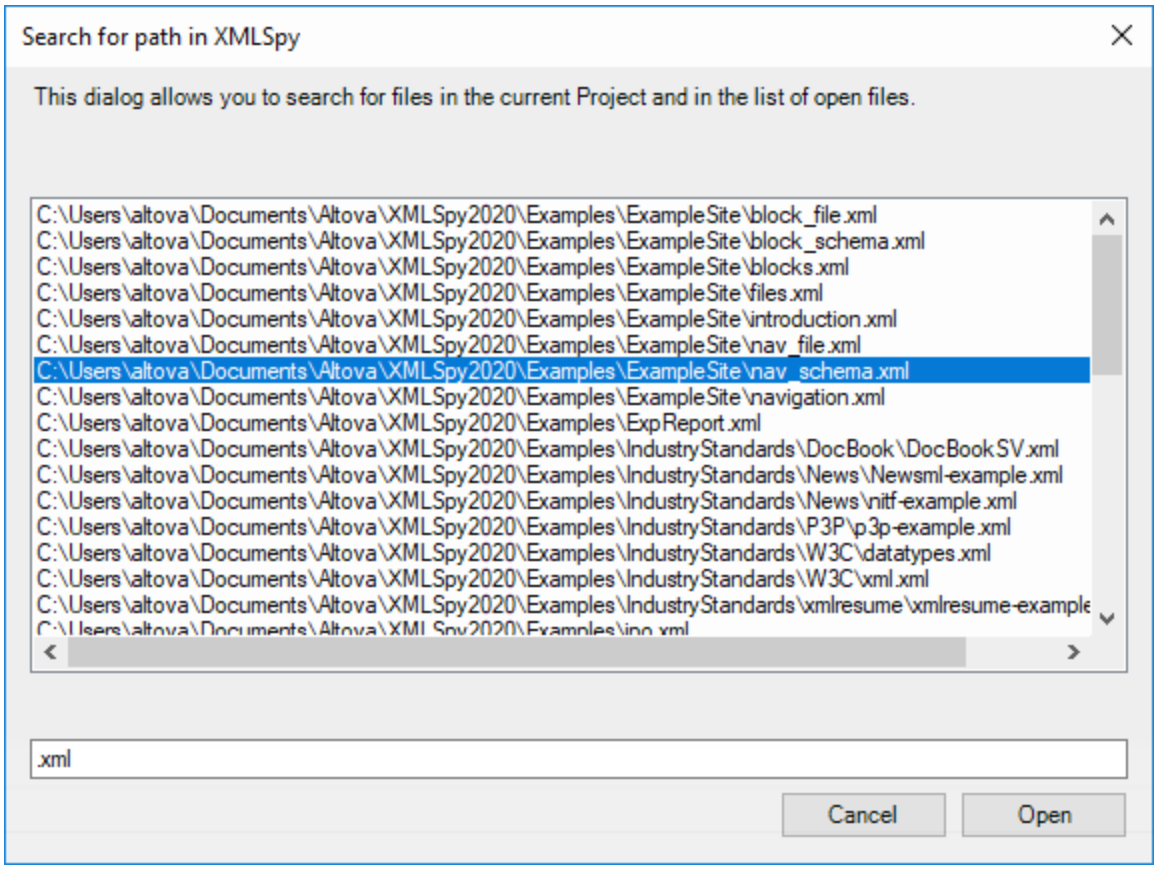
Para habilitar el proyecto de scripting como proyecto global de sc de XMLSpy:

1. En el menú **Herramientas** haga clic en **Opciones**.
2. Haga clic en la pestaña *Scripting*.
3. En "Archivo de proyecto de scripting global" haga clic en **Examinar** y seleccione el archivo **SampleScripts.asprj** en la ruta indicada más arriba.
4. Este proyecto de scripting no tiene macros o controladores de eventos automáticos, por lo que no necesita marcar las casillas **Ejecutar macros automáticas...** o **Procesar eventos**.
5. Haga clic en **Aplicar**.

En este punto en el menú **Herramientas | Macros** aparecen varios elementos nuevos.

Para ejecutar la macro "SearchPath":

1. Abra un proyecto de XMLSpy que contenga varios archivos (en este ejemplo, **C:\Users\<user>\Documents\Altova\XMLSpy2023\Examples\Examples.spp**).
2. En el menú Herramientas haga clic en Macros y después haga clic en **Ruta de búsqueda** (**SearchPath**).
3. Teclee el término de búsqueda (en este ejemplo, ".xml").



En la imagen se ve cómo aparecen todos los nombres de archivos que contienen el término de búsqueda. Para mostrar uno de esos elementos en el editor principal haga clic en cualquiera de ellos y después haga clic en **Abrir**.

29.1.2 Comandos integrados

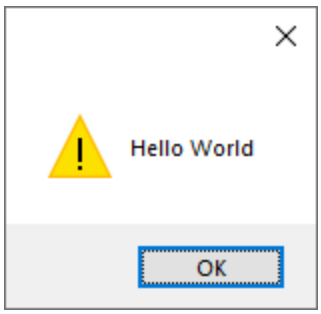
En este apartado se explican todos los comandos que puede usar el el Editor de scripts de XMLSpy.

- alert 1657
- confirm 1657
- CLR.Create 1658
- CLR.Import 1659
- CLR.LoadAssembly 1659
- CLR.ShowImports 1660
- CLR.ShowLoadedAssemblies 1661
- CLR.Static 1662
- CreateForm 1662
- doevents 1663
- lastform 1664
- prompt 1664
- ShowForm 1665
- watchdog 1666

© 2016-2022 Altova GmbH

29.1.2.1 alert

Muestra un cuadro de mensaje en el que aparece un texto dado y el botón "Aceptar". Para continuar el usuario debe hacer clic en "Aceptar".



Firma

Para JScript la firma es:

```
alert(strMessage : String) -> void
```

Para VBScript la firma es:

```
MsgBox(strMessage : String) -> void
```

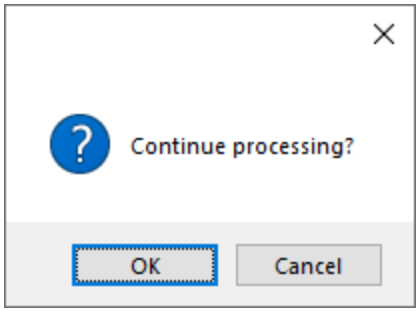
Ejemplo

Est código JScript muestra una caja de mensaje con el texto "Hello, World".

```
alert("Hello World");
```

29.1.2.2 confirm

Abre un cuadro de diálogo en el que aparecen un mensaje dado, un botón para confirmar y uno para cancelar. El usuario tiene que hacer clic en "Aceptar" o "Cancelar" para continuar. Devuelve un valor booleano que representa la respuesta del usuario. Si este hace clic en "Aceptar" la función devuelve **true**; si hace clic en "Cancelar" la función devuelve **false**.



Firma

```
confirm(strMessage : String) -> result : Boolean
```

Ejemplo (JScript)

```
if ( confirm( "¿Seguir procesando?" ) == false )
  alert("Ha cancelado esta acción");
```

Ejemplo (VBScript)

```
If ( confirm( "¿Seguir procesando?" ) = false ) Then
  MsgBox("Ha cancelado esta acción")
End If
```

29.1.2.3 CLR.Create

Crea una instancia de objeto .NET nueva con el nombre de clase dado como argumento. Si se pasa más de un argumento, los argumentos siguientes se interpretan como argumentos para el constructor del objeto .NET. El valor de retorno es una referencia al objeto .NET que se creó.

Firma

```
CLR.Create(strTypeNameCLR : String, constructor arguments ... ) -> object
```

Ejemplo

Este extracto de código de JScript ilustra cómo crear instancias de varias clases .NET.

```
// Crear una lista ArrayList
var objArray = CLR.Create("System.Collections.ArrayList");
// Crear una lista ListViewItem
var newItem = CLR.Create( "System.Windows.Forms.ListViewItem", "NewItemText" );
// Crear una lista List<string>
```


© 2016-2022 Altova GmbH

```
var coll = CLR.Create( "System.Collections.Generic.List<System.String>" );
// Importa los espacios de nombres necesarios y crea un objeto Diccionario
CLR.Import( "System" );
CLR.Import( "System.Collections.Generic" );
var dictionary = CLR.Create( "Dictionary< String, Dictionary< String, String > >" );
```

29.1.2.4 CLR.Import

Importa un espacio de nombres. Este es el equivalente en scripting de `using` en C# o `imports` en VB.Net. Al llamar a `CLR.Import` no se puede excluir la parte del espacio de nombres en llamadas posteriores, como `CLR.Create()` and `CLR.Static()`.

Nota: al importar un espacio de nombres no se añade ni se carga el ensamblado correspondiente en el proyecto de scripting. Puede añadir ensamblados al proyecto de scripting de forma dinámica (en el momento de la ejecución) en el código fuente llamando a CLR.LoadAssembly 1659.

Firma

```
CLR.Import(strNamespaceCLR : String) -> void
```

Ejemplo

En lugar de tener que usar espacios de nombres completos, como aquí:

```
if ( ShowForm( "FormName" ) == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
{
    var sName = lastform.textboxFirstName.Text + " " + lastform.textboxLastName.Text;
    CLR.Static( "System.Windows.Forms.MessageBox" ).Show( "Hola " + sName );
}
```

Puede importar espacios de nombres primero y después usar la forma abreviada:

```
CLR.Import( "System.Windows.Forms" );

if ( ShowForm( "FormName" ) == CLR.Static( "DialogResult" ).OK )
{
    var sName = lastform.textboxFirstName.Text + " " + lastform.textboxLastName.Text;
    CLR.Static( "MessageBox" ).Show( "Hola " + sName );
}
```

29.1.2.5 CLR.LoadAssembly

Carga el ensamblado .NET con el nombre de ensamblado largo dado o con la ruta de acceso al archivo. Devuelve el valor booleano **true** si el ensamblado se pudo cargar; en caso contrario devuelve **false**.

Firma

```
CLR.LoadAssembly(strAssemblyNameCLR : String, showLoadErrors : Boolean) -> result :
Boolean
```

Ejemplo

Este código JScript intenta establecer el texto del portapapeles cargando el ensamblado necesario de forma dinámica.

```
// establecer el texto del portapapeles (si se puede)
// System.Windows.Clipboard es parte del ensamblado PresentationCore, así que carga
primero este ensamblado:
if ( CLR.LoadAssembly( "PresentationCore, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35", true ) )
{
    var clipboard = CLR.Static( "System.Windows.Clipboard" );
    if ( clipboard != null )
        clipboard.SetText( "HelloClipboard" );
}
```

29.1.2.6 CLR.ShowImports

Abre un cuadro de mensaje que muestra el espacio de nombres importado en ese momento. El usuario tiene que hacer clic en "Aceptar" para continuar.

Firma

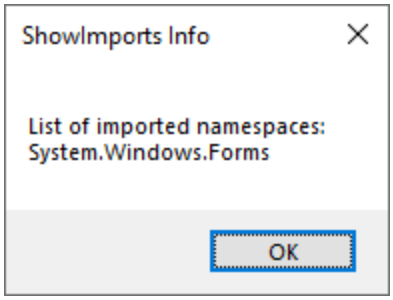
```
CLR.ShowImports() -> void
```

Ejemplo

Este código JScript primero importa un espacio de nombres y después muestra la lista de espacios de nombres importados:

```
CLR.Import( "System.Windows.Forms");
CLR.ShowImports();
```


© 2016-2022 Altova GmbH



29.1.2.7 CLR.ShowLoadedAssemblies

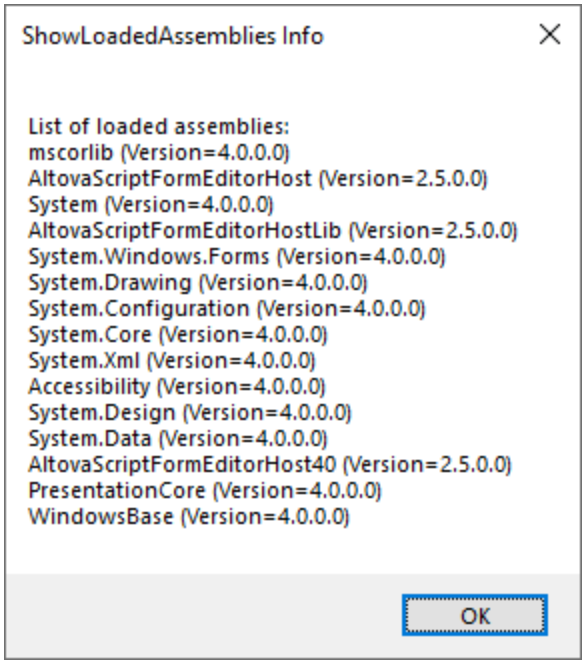
Abre un cuadro de mensaje que muestra los ensamblados cargados en ese momento. El usuario tiene que hacer clic en "Aceptar" para continuar.

Firma

```
CLR.ShowLoadedAssemblies() -> void
```

Ejemplo

```
CLR.ShowLoadedAssemblies();
```



29.1.2.8 CLR.Static

Devuelve una referencia a un objeto .NET estático. Puede usar esta función para acceder a tipos .NET que no tengan instancias y contengan solamente miembros estáticos.

Firma

```
CLR.Static(strTypeNameCLR : String) -> object
```

Ejemplo (JScript)

```
// Obtener el valor de una .NET Enum en una variable
var enumValStretch = CLR.Static( "System.Windows.Forms.ImageLayout" ).Stretch

// Establecer el valor del portapapeles de Windows
var clipboard = CLR.Static( "System.Windows.Clipboard" );
clipboard.SetText( "HelloClipboard" );

// Comprobar qué botón pulsó el usuario en un cuadro de diálogo
if ( ShowForm( "FormName" ) == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
    alert( "ok" );
else
    alert( "cancel" );
```

29.1.2.9 CreateForm

Instancia el objeto formulario `Form` identificado por el nombre dado como argumento. El formulario debe existir en la carpeta "Formularios" del proyecto de scripting. Devuelve el objeto formulario (`System.Windows.Forms.Form`) que corresponde al nombre dato o `null` si no existe ningún formulario con ese nombre.

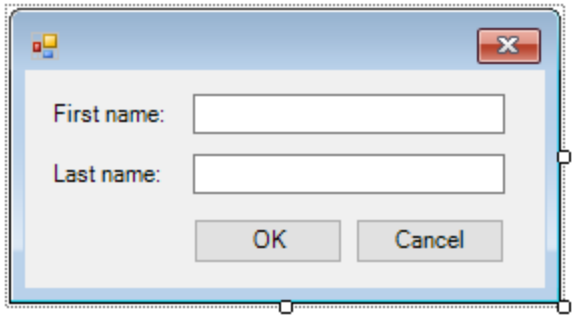
Firma

```
CreateForm (strFormName : String) -> System.Windows.Forms.Form | null
```

Ejemplo

Imagine que en el proyecto de scripting existe un formulario llamado "FormName".

© 2016-2022 Altova GmbH



Este código JScript instancia el formulario con algunos valores predeterminados y lo muestra al usuario.

```
var myForm = CreateForm( "FormName" );
if ( myForm != null )
{
   myForm.textboxFirstName.Text = "Daniela";
   myForm.textboxLastName.Text = "Heidegger";
   var dialogResult = myForm.ShowDialog();
}
```

En consecuencia, `dialogResult` se puede seguir evaluando así:

```
if ( dialogResult == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
    alert( "ok" );
else
    alert( "cancel" );
```

Nota: el código de la imagen anterior solamente funciona si la propiedad `DialogResult` de los botones "Aceptar" y "Cancelar" está configurada correctamente en el panel *Propiedades* (por ejemplo, debe ser `OK` para el botón "Aceptar").

29.1.2.10 doevents

Procesa todos los mensajes de Windows que estén en ese momento en la cola de mensajes.

Firma

```
doevents() -> void
```

Ejemplo (JScript)

```
for ( i=0; i < nLongLastingProcess; ++i )
{
    // ejecutar procesos de larga duración
```

```
    doevents(); // procesar mensajes de Windows; permitir que la IGU se actualice
}
```

29.1.2.11 lastform

Este es un campo global que devuelve una referencia al último objeto formulario que se creó con `CreateForm()` o `ShowForm()`.

Firma

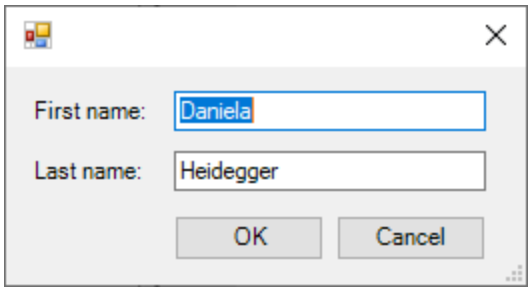
```
lastform -> formObj : System.Windows.Forms.Form
```

Ejemplo

Este código JScript muestra el formulario "FormName" como cuadro de diálogo.

```
CreateForm( "FormName" );
if ( lastform != null )
{
    lastform.textboxFirstName.Text = "Daniela";
    lastform.textboxLastName.Text = "Heidegger";
    var dialogResult = lastform.ShowDialog();
}
```

Los valores de los dos controles de campos de texto se inicializan con ayuda de `lastform`.



29.1.2.12 prompt

Abre un cuadro de diálogo en el que aparece un mensaje y un control de campo de texto con una respuesta predeterminada. Esto permite al usuario introducir un valor simple de cadena de texto. El valor de retorno es una cadena que contiene el valor del campo de texto o `null` si el usuario seleccionó "Cancelar".

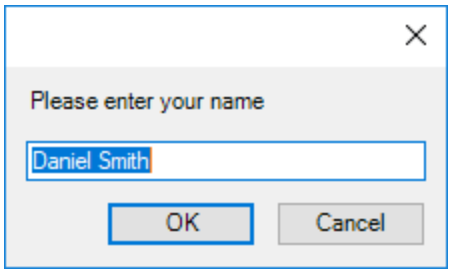
Firma

```
prompt(strMessage : String, strDefault : String) -> val : String
```


© 2016-2022 Altova GmbH

Ejemplo

```
var name = prompt( "Please enter your name", "Daniel Smith" );
if ( name != null )
    alert( "¡Hola, " + name + "!" );
```



29.1.2.13 ShowForm

Instancia un objeto formulario nuevo a partir del nombre de formulario dado y lo muestra de inmediato como cuadro de diálogo. El valor de retorno es un número entero que representa el resultado generado: `DialogResult` (`System.Windows.Forms.DialogResult`). Para ver la lista de posibles valores consulte la documentación de `DialogResult` Enum (https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.dialogresult?view=netframework-4.8).

Firma

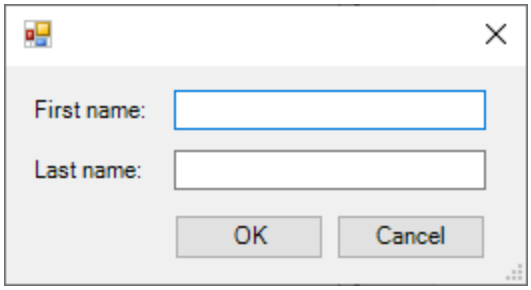
```
ShowForm(strFormName : String) -> result : Integer
```

Ejemplo

Este código JScript

```
var dialogResult = ShowForm( "FormName" );
```

muestra el formulario "FormName" como cuadro de diálogo:



Así, `DialogResult` se puede seguir evaluando, por ejemplo:

```
if ( dialogResult == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
    alert( "ok" );
else
    alert( "cancel" );
```

Nota: el código de la imagen anterior solamente funciona si la propiedad `DialogResult` de los botones "Aceptar" y "Cancelar" está configurada correctamente en el panel *Propiedades* (por ejemplo, debe ser `OK` para el botón "Aceptar").

29.1.2.14 watchdog

Los scripts largos que dan un uso intensivo a la CPU pueden preguntar al usuario si el script debe finalizar. El método `watchdog()` se usa para deshabilitar o habilitar la opción de perro guardián. Por defecto, esta opción está habilitada.

También se puede usar `watchdog(true)` para reiniciar la función de perro guardián. Esto puede ser útil antes de ejecutar tareas largas que hagan un uso intensivo de la CPU para asegurarse de que cuentan con la capacidad de procesamiento máxima permitida.

Firma

```
watchdog(bEnable : boolean) -> void
```

Ejemplo

```
watchdog( false );  // deshabilitar watchdog: la siguiente declaración hace un uso
intensivo de la CPU
doCPUIntensiveScript();
watchdog( true );   // volver a habilitar watchdog
```

29.1.3 Habilitar scripts y macros

Una vez ha completado y probado un proyecto de scripting puede usarlo de varias maneras:

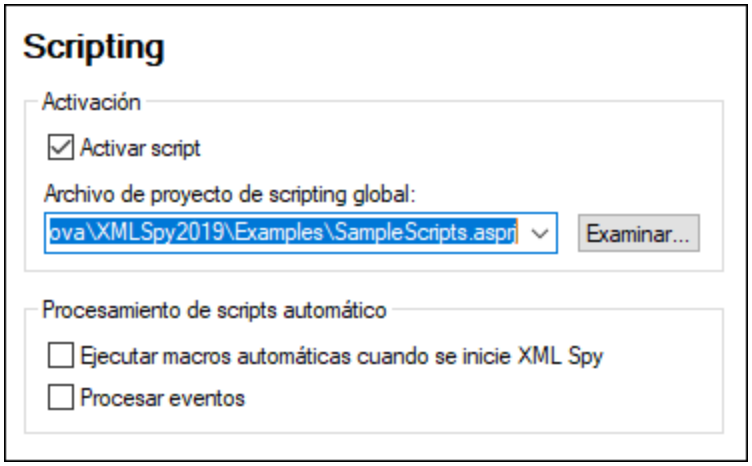
1. Como proyecto global de scripting para XMLSpy. Esto significa que XMLSpy puede usar todos los scripts y las macros del proyecto de scripting.
2. A nivel del proyecto. Esto significa que junto al proyecto de XMLSpy se guarda también una referencia al archivo .asprj. Cuando se abre el proyecto de XMLSpy también se puede llamar a los scripts y las macros asociados.

Para definir un proyecto de scripting como global:

1. En el menú **Herramientas** haga clic en **Opciones**.
2. Haga clic en la pestaña **Script/Scripting**.

© 2016-2022 Altova GmbH

3. Marque la casilla *Activar script* y navegue hasta el archivo `.asprj` para usarlo como proyecto global de scripting.



También puede habilitar estas otras opciones de procesamiento de scripts:

Ejecutar macros automáticas cuando se inicie XMLSpy	Si marca esta casilla, cualquier macro que esté definida como "automática" en el proyecto se desencadena automáticamente cuando se inicia XMLSpy.
Procesar eventos	Marque esta casilla si sus scripts están vinculados a algún evento de aplicación. Desmárquela para evitar que los scripts reaccionen con los eventos.

Para habilitar el proyecto de scripting a nivel del proyecto:

1. Abra el proyecto.
2. En el menú **Proyecto** haga clic en **Configuración del proyecto**.
3. Marque la casilla **Activar scripts de proyecto** y navegue hasta el archivo .asprj.

La casilla *Ejecutar macros automáticas* funciona como hemos explicado en el punto anterior.

29.1.3.1 Ejecutar macros

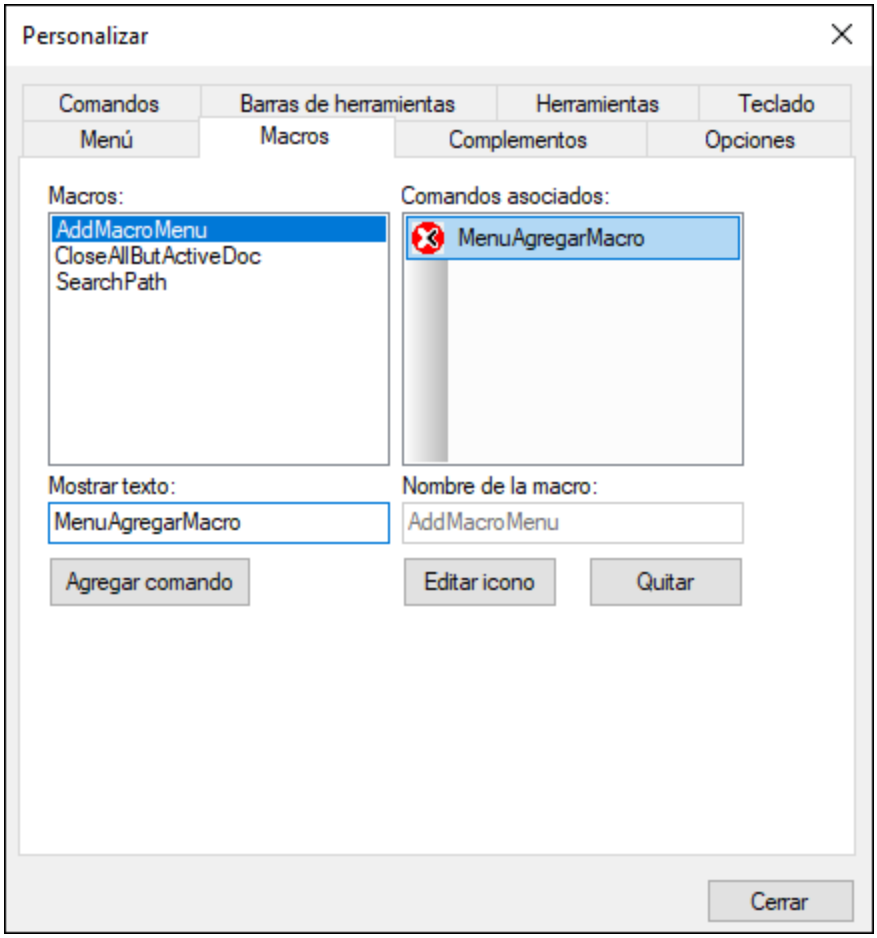
Cuando un proyecto de scripting está activo en XMLSpy, cualquier macro que esté disponible en ese proyecto aparece en el menú **Herramientas | Macros**, por lo que puede ejecutar una macro en cualquier momento con el comando de menú correspondiente, como por ejemplo **Herramientas | Macros | <UnaMacro>**.

Las macros que se hayan configurado como automáticas se ejecutan automáticamente al iniciarse XMLSpy, siempre que se haya habilitado este comportamiento en las opciones, tal y como se describe en el apartado Habilitar scripts y macros 1666.

Para poder acceder más fácilmente a las macros puede crear botones para la barra de herramientas que activen esas macros:

1. En el menú **Herramientas** haga clic en **Personalizar**.

2. Haga clic en la pestaña *Macros*. En la lista aparecen todas las macros disponibles a nivel de la aplicación (en el proyecto global de scripting).
3. Haga clic en **Agregar comando**.



4. También puede hacer clic en **Editar icono** y seleccionar un icono nuevo para esa macro. Para asignar un atajo de teclado para esa macro, vaya a la pestaña *Teclado*.
5. Arrastre la macro desde el panel *Comandos asociados* hasta la barra de herramientas en la que quiere que aparezca.

Para eliminar una macro de una barra de herramientas:

1. En el menú **Herramientas** haga clic en **Personalizar**.
2. Haga clic en la pestaña *Macros*.
3. Arrastre la macro desde la barra de herramientas en la que aparece de vuelva hasta el panel *Comandos asociados*.

© 2016-2022 Altova GmbH

29.2 Complementos de XMLSpy para entornos IDE

XMLSpy permite crear complementos para entornos IDE e integrarlos en XMLSpy.

Puede usar estos complementos para:

- Configurar su versión de XMLSpy y agregar comandos a través de menús, iconos, botones, etc.
- Reaccionar a eventos desde XMLSpy.
- Ejecutar su propio código dentro de XMLSpy con acceso a toda la API de XMLSpy

XMLSpy espera que el complemento implemente la interfaz IXMLSpyPlugIn[1679]. Con el paquete de instalación vienen ejemplos en C# y C++, los lenguajes compatibles con XMLSpy, que se instalan en la carpeta `XMLSpy2023\Examples\IDEPlugin` de la carpeta de instalación:

Windows 7, 8, 10, 10, 11	`C:/Users/<usuario>/Documentos`

Consulte el apartado *Archivos de muestra ATL*[1673] para ver un ejemplo con C++.

29.2.1 Cómo registrar complementos para entornos IDE

XMLSpy mantiene una clave en el registro en la que almacena todos los complementos registrados para entornos IDE:

```
HKEY_CURRENT_USER\Software\Altova\XML Spy\PlugIns
```

Todos los valores de esta clave se tratan como referencias a los complementos registrados y deben tener este formato:

Nombre del valor:	`ProgID` del complemento
Tipo de valor:	debe ser `REG_SZ`
Datos del valor:	`CLSID` del componente

Cada vez que se inicia la aplicación, se examinan los valores de la clave `PlugIns` y se cargan los complementos que estén registrados.

Registrar el complemento a mano

Para registrar un complemento a mano utilice el cuadro de diálogo "Personalizar" del menú **Herramientas** de XMLSpy. Con el botón **Agregar complemento...** puede especificar qué DLL implementa el complemento. XMLSpy registra el DLL como servidor de COM y añade la correspondiente entrada en su clave `PlugIns`.

Si tiene problemas durante el registro manual, compruebe si el CLSID del complemento coincide con el de la clave `PlugIns`. Si es así, puede que el nombre del DLL del complemento no fuera lo suficientemente único. Utilice otro nombre o registre el complemento directamente (ver más abajo).

Registrar el complemento directamente

Los complementos se pueden registrar directamente: para ello debe registrar el DLL primero y después agregar el valor que le corresponda en la clave `PlugIns` de XMLSpy durante la configuración del complemento, por ejemplo. El nuevo complemento se activará la próxima vez que se inicie XMLSpy.

Crear complementos

En la carpeta `Examples\IDEPlugin` de la carpeta (Mis) Documentos [36] de la aplicación encontrará código fuente para crear ejemplos de complementos en XMLSpy. Siga estas instrucciones para crear un complemento a partir de este código fuente:

1. Abra la solución que desea crear como complemento en Visual Studio.
2. Cree el complemento con el comando del menú **Compilación**.
3. El archivo DLL del complemento se creará en la carpeta `Bin` o `Debug`. Este archivo DLL es el que se debe añadir como complemento (ver instrucciones más arriba).

Nota: actualmente los lenguajes compatibles son C# y C++.

29.2.2 Controles ActiveX

Los controles ActiveX son compatibles y los complementos para entornos IDE que también sean control ActiveX aparecerán en una barra de control de diálogo. Además, en la carpeta `IDEPlugin` de la carpeta `Examples` de la aplicación hay un complemento de muestra que también es un control ActiveX.

29.2.3 Configuración XML

El complemento para entornos IDE permite modificar la interfaz de usuario de XMLSpy. Esto se hace describiendo cada una de las modificaciones por separado mediante secuencias de datos XML. La configuración XML se pasa a XMLSpy con el método GetUIModifications [1684] de la interfaz IXMLSpyPlugIn.

El archivo XML que contiene las modificaciones de la interfaz de usuario para el complemento debe tener esta estructura:

```
<ConfigurationData>
              <ImageFile>ruta de acceso del archivo de imagen</ImageFile>
              <Modifications>
                     <Modification>
                            ...
                     </Modification>
                     ...
              </Modifications>
       </ConfigurationData>
```

Puede definir iconos o botones de barras de herramientas para los nuevos comandos de menú que el complemento añade a la interfaz gráfica de XMLSpy. La ruta de acceso del archivo que contiene las imágenes se establece con el elemento `ImageFile`. Cada imagen debe tener 16 x 16 píxels y usar 256 colores como

© 2016-2022 Altova GmbH

máximo. Las referencias a la imagen deben organizarse de izquierda a derecha en una sola línea (<ImageFile>...). El valor de índice de la imagen situada más a la derecha es cero.

El elemento `Modifications` puede tener un número indefinido de elemento secundarios `Modification`. Cada elemento `Modification` define un cambio concreto en la interfaz gráfica estándar de XMLSpy. A partir de la versión 4.3 también se pueden quitar elementos de la interfaz de XMLSpy.

Estructura de los elementos Modification

Todos los elementos `Modification` están compuestos por dos elementos secundarios:

```
<Modification>
       <Action>tipo de acción</Action>
       <UIElement Type="tipo de elemento de interfaz gráfica">
       </UIElement>
</Modification>
```

Estos son los valores válidos para el elemento `Action`:

`Add`: para agregar el elemento a la interfaz de XMLSpy
`Hide`: para ocultar el elemento en la interfaz de XMLSpy
`Remove`: para quitar el elemento del cuadro de lista *Comandos* del cuadro de diálogo "Personalizar".

También puede combinar varios valores en el elemento `Action` (p. ej. `Hide Remove`).

El elemento `UIElement` describe los elementos de la interfaz gráfica de XMLSpy, nuevos o ya existentes. Se trata de estos elementos: barras de herramientas nuevas, botones nuevos, menús o comandos de menús. El atributo **type** define qué elemento es descrito por el elemento XML.

Elementos secundarios frecuentes de UIElement

Los elementos ID y Name son válidos para todos los tipos diferentes de fragmentos XML `UIElement`. Sin embargo, se puede ignorar uno de los valores para un tipo concreto de `UIElement` (p. ej. Name se ignora para un separador).

```
<ID></ID>
<Name></Name>
```

Si `UIElement` describe un elemento actual de la interfaz gráfica, el valor del elemento ID viene predefinido en XMLSpy. Lo normal es que estos valores del elemento ID no sean conocidos. Si el fragmento XML describe una parte nueva de la interfaz gráfica, entonces el ID es aleatorio y su valor debería ser menor a 1000.

El elemento `Name` establece el valor textual. Los elementos actuales de la interfaz gráfica se pueden identificar con su nombre (p. ej. menús y elementos de menús). En el caso de elementos nuevos de la interfaz, el elemento `Name` establece el título (p. ej. el título de una barra de herramientas o el texto de un comando de menú).

Barras de herramientas y menús

Para definir una barra de herramienta hace falta especificar el ID y el nombre de la barra de herramientas. Para las barras de herramientas ya existentes se puede indicar el nombre solamente o el ID, si se conoce. Para crear una barra de herramientas nueva es necesario indicar ambos valores. El atributo **type** debe ser igual a "ToolBar".

```
<UIElement Type="ToolBar">
	<ID>1</ID>
	<Name>TestPlugIn</Name>
</UIElement>
```

Para especificar un menú de XMLSpy son necesarios dos parámetros:

- El ID de la barra de menú que contiene el menú. Si en la ventana principal no hay ningún documento XML abierto, el ID de la barra de menú es 128. Si hay algún documento XML abierto, el ID de la barra de menú es 129.
- El nombre del menú. Los menús no tienen un valor ID asociado. En el fragmento siguiente, por ejemplo, se define el menú Edición de la barra de menú que está activa cuando hay un documento XML abierto como mínimo:

```
<UIElement Type="Menu">
	<ID>129</ID>
	<Name>Edición</Name>
</UIElement>
```

Si quiere crear un menú nuevo hace falta usar un elemento más: el elemento `Place`, que define la posición del nuevo menú en la barra de menús:

```
<UIElement Type="Menu">
	<ID>129</ID>
	<Name>Menú PlugIn</Name>
	<Place>12</Place>
</UIElement>
```

El valor -1 para el elemento `Place` coloca el botón o menú nuevo al final de la barra de herramientas o del menú respectivamente.

Comandos

Si añade un comando nuevo, a través de una barra de herramientas o de un comando de menú, el fragmento `UIElement` puede contener cualquiera de estos subelementos:

```
<MacroName></MacroName>
<Info></Info>
<ImageID></ImageID>
```

Si se especifica el elemento `MacroName`, XMLSpy busca una macro que tenga el mismo nombre en el entorno de scripting y la ejecuta cada vez que se procese este comando.

El elemento `Info` contiene una breve descripción que aparece en la barra de estado cuando se pasa el puntero del ratón por encima del comando asociado (tanto por el botón como por el comando de menú).

`ImageID` define el índice del icono del archivo de imagen. Recuerde que todos los iconos se almacenan en un solo archivo de imagen.

Para definir un botón de barra de herramientas debe crear un elemento `UIElement` con esta estructura:

```
<UIElement Type="ToolBarItem">
	<!--no volver a utilizar ID locales incluso si los comandos hacen lo mismo-->
```

 © 2016-2022 Altova GmbH

```
            <ID>5</ID>
            <Name>Abrir archivo del repositorio...</Name>
            <!--Establecer posición en -1 si se trata del primer botón que se debe
insertar-->
            <Place>-1</Place>
            <ImageID>0</ImageID>
            <ToolBarID>1</ToolBarID>
            <!--en lugar del ID de la barra de herramientas también se puede usar el
nombre de la barra de herramientas-->
            <ToolBarName>TestPlugIn</ToolBarName>
      </UIElement>
```

Otros elementos que se pueden utilizar para declarar un botón de barra de herramientas son `Place`, `ToolBarID` y `ToolBarName`. Los elementos `ToolBarID` y `ToolBarName` sirven para identificar la barra de herramientas que contiene el botón nuevo o el botón actual. El valor textual de `ToolBarName` distingue entre mayúsculas y minúsculas. El atributo **type** (del elemento `UIElement`) debe ser igual a "ToolBarItem".

Para definir un comando de menú, además de los elementos estándar, también están los elementos `MenuID`, `Place` y `Parent`. El elemento `MenuID` puede ser 128 o 129. Para más información sobre estos valores consulte el párrafo anterior *Barras de herramientas y menús*.

El elemento `Parent` sirve para identificar el **menú** donde se debe insertar el comando de menú nuevo. Como los comandos de los submenús no tienen un ID único de Windows, necesitamos identificar de otra manera el primario del comando de menú. El valor del elemento `Parent` es una ruta de acceso al comando de menú.

El valor textual del elemento `Parent` debe equivaler al nombre del **menú primario del submenú** y el nombre del submenú debe ir separado por dos puntos. Si el menú no es un primario (porque no es un submenú) añada dos puntos al principio del nombre. El atributo **type** debe tener el valor "MenuItem". Por ejemplo, este elemento `UIElement` define un comando de menú:

```
      <UIElement Type="MenuItem">
            <!--este elemento es un ID de comando local-->
            <ID>3</ID>
            <Name>Abrir archivo de repositorio...</Name>
            <Place>-1</Place>
            <MenuID>129</MenuID>
            <Parent>:Menú PlugIn</Parent>
            <ImageID>0</ImageID>
      </UIElement>
```

XMLSpy permite añadir separadores de barras de herramientas y de menús si el valor del elemento `ID` es 0.

29.2.4 Archivos de muestra ATL

Esta sección explica cómo crear un sencillo DLL para el complemento de XMLSpy para entornos IDE usando la biblioteca ATL. Para generar el DLL es necesario saber trabajar con la biblioteca ATL, conocer los asistentes que generan objetos ATL nuevos y tener conocimientos de MS Visual Studio.

Para acceder a la API la implementación importa la biblioteca de tipos de XMLSpy. El código lee varias propiedades y llama a métodos usando los punteros inteligentes que ofrece la instrucción #import.

Además el código de muestra usa la clase MFC CString y macros de conversión ATL como W2T.

En general, estos son los pasos necesarios para crear un DLL con la biblioteca ATL:

1. Abra VisualStudio y seleccione Nuevo en el menú Archivo.
2. Seleccione la pestaña Proyectos.
3. Seleccione Asistente para proyectos ATL y escriba el nombre del proyecto.
4. Si quiere usar clases MFC o quiere crear un proyecto para el código de muestra, seleccione Compatibilidad para MFC.

Una vez creados los archivos de proyecto puede añadir un objeto ATL para implementar la interfaz IXMLSpyPlugIn:

1. En el menú Insertar seleccione Objeto ATL nuevo.
2. En el asistente seleccione Objeto simple y haga clic en Siguiente.
3. Escriba un nombre para el objeto.
4. En la pestaña Atributos seleccione Personal para el tipo de interfaz y deshabilite la opción Agregación.

El resultado es un código base para la implementación de la interfaz del complemento para entornos IDE. Consulte el resto de apartados de esta sección para modificar el código y añadir funciones básicas.

29.2.4.1 Descripción de interfaz (IDL)

El IDL del objeto ATL recién creado contiene una declaración para una interfaz COM.

- Esta declaración de interfaz debe reemplazarse con la declaración de IXMLSpyPlugIn tal y como aparece más abajo.
- El IDL también debe incluir la definición de la enumeración SPYUpdateAction.
- Reemplace el nombre de interfaz predeterminado generado (creado con el asistente) con el nombre *IXMLSpyPlugIn* en la declaración coclass. El IDL se debería parecer al código del ejemplo siguiente:

Tras crear el objeto ATL, es necesario implementar la interfaz del complemento de XMLSpy para entornos IDE:

```
import "oaidl.idl";
import "ocidl.idl";

// ----- inserte este bloque en su archivo IDL -----
	typedef enum {
		spyEnable = 1,
		spyDisable = 2,
		spyCheck = 4,
		spyUncheck = 8
	} SPYUpdateAction;

// ----- fin del bloque insertado ----
```

 © 2016-2022 Altova GmbH

```
// ----- Por, ej. entrada de interfaz generada automáticamente por el asistente ATL -----
//     [
//                   object,
//                   uuid(AB7CD86A-8145-429A-A1F3-270692EO8AFC),

//                   helpstring("IXMLSpyPlugIn Interface")
//                   pointer_default(unique)
//     ]
//     interface IXMLSpyPlugIn : IUnknown
//     {
//     };

// ----- fin de la entrada de interfaz generada automáticamente


// ----- reemplazar la entrada de interfaz (ver más arriba) generada por el asistente ATL
con este bloque -----

       [
              odl,
              uuid(88F2A622-4B7E-42CD-8D04-3C0E5389DD85),
              helpstring("IXMLSpyPlugIn Interface")
       ]
       interface IXMLSpyPlugIn : IUnknown
              {
                     HRESULT _stdcall OnCommand([in] long nID, [in] IDispatch* pXMLSpy);

                     HRESULT _stdcall OnUpdateCommand([in] long nID, [in] IDispatch*
pXMLSpy, [out, retval] SPYUpdateAction* pAction);

                     HRESULT _stdcall OnEvent([in] long nEventID, [in] SAFEARRAY(VARIANT)*
arrayParameters, [in] IDispatch* pXMLSpy, [out, retval] VARIANT* pReturnValue);

                     HRESULT _stdcall GetUIModifications([out, retval] BSTR*
pModificationsXML);

                     HRESULT _stdcall GetDescription([out, retval] BSTR* pDescription);
              };

// ----- fin del bloque -----


// ----- El código que aparece ahora lo genera automáticamente el asistente ATL y puede ser
diferente en cada caso -----

 [
       uuid(24FE0D1B-3FC0-494E-B36E-1D4CE412B014),
       version(1.0),
       helpstring("XMLSpyIDEPlugInDLL 1.0 Type Library")
 ]
 library XMLSPYIDEPLUGINDLLLib
 {
       importlib("stdole32.tlb");
```

```
	importlib("stdole2.tlb");

	[
		uuid(3800E791-7F6B-4ACD-9E32-2AC184444501),
		helpstring("XMLSpyIDEPlugIn Class")
	]
	coclass XMLSpyIDEPlugIn
	{
		[default] interface IXMLSpyPlugIn;   // ----- definir IXMLSpyPlugIn como
interfaz predeterminada -----
	};
};
```

29.2.4.2 Definición de clase

En la definición de clase del objeto ATL deben realizarse varios cambios. La clase debe derivarse de IXMLSpyPlugIn, el *mapa de la interfaz* necesita una entrada para IXMLSpyPlugIn y los métodos de la interfaz del complemento para entornos IDE debe declararse:

```
#ifndef __XMLSPYIDEPLUGIN_H_
#define __XMLSPYIDEPLUGIN_H_

#include "resource.h"       // símbolos principales

/////////////////////////////////////////////////////////////////////////////
// CXMLSpyIDEPlugIn
class ATL_NO_VTABLE CXMLSpyIDEPlugIn :
	public CComObjectRootEx<CComSingleThreadModel>,
	public CComCoClass<CXMLSpyIDEPlugIn, &CLSID_XMLSpyIDEPlugIn>,
	public IXMLSpyPlugIn
{
public:
	CXMLSpyIDEPlugIn()
	{
	}

DECLARE_REGISTRY_RESOURCEID(IDR_XMLSPYIDEPLUGIN)
DECLARE_NOT_AGGREGATABLE(CXMLSpyIDEPlugIn)

DECLARE_PROTECT_FINAL_CONSTRUCT()

BEGIN_COM_MAP(CXMLSpyIDEPlugIn)
	COM_INTERFACE_ENTRY(IXMLSpyPlugIn)
END_COM_MAP()

// IXMLSpyIDEPlugIn
public:
	virtual HRESULT _stdcall OnCommand(long nID, IDispatch* pXMLSpy);

	virtual HRESULT _stdcall OnUpdateCommand(long nID, IDispatch* pXMLSpy,
SPYUpdateAction* pAction);
```


© 2016-2022 Altova GmbH

```
	virtual HRESULT _stdcall OnEvent(long nEventID, SAFEARRAY **arrayParameters,
IDispatch* pXMLSpy, VARIANT* pReturnValue);

	virtual HRESULT _stdcall GetUIModifications(BSTR* pModificationsXML);

	virtual HRESULT _stdcall GetDescription(BSTR* pDescription);
};

#endif //__XMLSPYIDEPLUGIN_H_
```

29.2.4.3 Implementación

El código de este apartado muestra una sencilla implementación de un complemento de XMLSpy para entornos IDE. Añade un comando de menú y un separador (disponible con XMLSpy) al menú **Herramientas**. Dentro del método OnUpdateCommand() el comando nuevo solamente se habilita cuando el documento activo se abre en la vista **Cuadrícula**. El comando busca el elemento XML que está resaltado y abre cualquier dirección URL empezando con "http://" a partir del valor textual del elemento.

```
////////////////////////////////////////////////////////////////////////////
// CXMLSpyIDEPlugIn

#import "XMLSpy.tlb"
using namespace XMLSpyLib;

HRESULT CXMLSpyIDEPlugIn::OnCommand(long nID, IDispatch* pXMLSpy)
{
	USES_CONVERSION;

	if(nID == 1)  {
		IApplicationPtr	ipSpyApp;

		if(pXMLSpy)   {
			if(SUCCEEDED(pXMLSpy->QueryInterface(__uuidof(IApplication),(void **)
&ipSpyApp))) {
				IDocumentPtr  ipDocPtr = ipSpyApp->ActiveDocument;

				// imaginamos que la vista Cuadrícula está activa
				if(ipDocPtr)  {
					IGridViewPtr  ipGridPtr = ipDocPtr->GridView;

					if(ipGridPtr) {
						IXMLDataPtr		ipXMLData = ipGridPtr-
>CurrentFocus;

						CString		strValue = W2T(ipXMLData-
>TextValue);

						if(!strValue.IsEmpty() && (strValue.Left(7) ==
_T("http://")))
```

```
::ShellExecute(NULL,_T("open"),W2T(ipXMLData->TextValue),NULL,NULL,SW_SHOWNORMAL);
                                        }
                                }
                        }
                }
        }

        return S_OK;
}


HRESULT CXMLSpyIDEPlugIn::OnUpdateCommand(long nID, IDispatch* pXMLSpy, SPYUpdateAction*
pAction)
{
        *pAction = spyDisable;

        if(nID == 1)  {
                IApplicationPtr      ipSpyApp;

                if(pXMLSpy)   {
                        if(SUCCEEDED(pXMLSpy->QueryInterface(__uuidof(IApplication),(void **)
&ipSpyApp)))  {
                                IDocumentPtr  ipDocPtr = ipSpyApp->ActiveDocument;

                                // habilitar solamente si la vista Cuadrícula está activa
                                if((ipDocPtr != NULL) && (ipDocPtr->CurrentViewMode ==
spyViewGrid))
                                        *pAction = spyEnable;
                        }
                }
        }

        return S_OK;
}


HRESULT CXMLSpyIDEPlugIn::OnEvent(long nEventID, SAFEARRAY **arrayParameters, IDispatch*
pXMLSpy, VARIANT* pReturnValue)
{
        return S_OK;
}

HRESULT CXMLSpyIDEPlugIn::GetUIModifications(BSTR* pModificationsXML)
{
        CComBSTR      bstrMods = _T(" \
                                <ConfigurationData>  \
                                        <Modifications>      ");
        // agregar "Abrir URL..." al menú Herramientas
        bstrMods.Append (_T(" \
                                                <Modification> \
                                                        <Action>Add</Action> \
                                                        <UIElement type=\"MenuItem\"> \
```

 © 2016-2022 Altova GmbH

```
                                                <ID>1</ID> \
                                                <Name>Abrir URL...</Name> \
                                                <Place>0</Place> \
                                                <MenuID>129</MenuID> \
                                                <Parent>:Herramientas</Parent> \
                                          </UIElement> \
                                    </Modification>     "));
      // agregar separador al menú Herramientas
      bstrMods.Append (_T(" \
                                    <Modification> \
                                          <Action>Add</Action> \
                                          <UIElement type=\"MenuItem\"> \
                                                <ID>0</ID> \
                                                <Place>1</Place> \
                                                <MenuID>129</MenuID> \
                                                <Parent>:Herramientas</Parent> \
                                          </UIElement> \
                                    </Modification>     "));
      // finalizar descripción de la modificación
      bstrMods.Append (_T(" \
                              </Modifications> \
                        </ConfigurationData>"));

      return bstrMods.CopyTo(pModificationsXML);
}


HRESULT CXMLSpyIDEPlugIn::GetDescription(BSTR* pDescription)
{
      CComBSTR     bstrDescr = _T("ATL C++ XMLSpy IDE PlugIn;Este complemento demuestra la
implementación de un DLL ATL sencillo como complemento de XMLSpy para entornos IDE.");
      return bstrDescr.CopyTo(pDescription);
}
```

29.2.5 IXMLSpyPlugIn

Métodos

OnCommand 1680
OnUpdateCommand 1681
OnEvent 1681
GetUIModifications 1684
GetDescription 1685

Descripción

Si se añade un DLL a XMLSpy como complemento para entornos IDE, debe registrar un componente COM que responda a una interfaz IXMLSpyPlugIn con el uuid(88F2A622-4B7E-42CD-8D04-3C0E5389DD85) reservado. De lo contrario no se reconocerá que es un complemento.

29.2.5.1 OnCommand

Declaración: `OnCommand(nID as long, pXMLSpy as IDispatch)`

Descripción

Al método `OnCommand()` de la implementación de la interfaz se le llama cada vez que se procesa un comando añadido por el complemento, ya sea un comando de menú o un botón de barra de herramientas. `nID` almacena el ID del comando definido por el elemento `ID` del correspondiente elemento `UIElement`.

`pXMLSpy` almacena una referencia a la interfaz de envío del objeto `Application` de XMLSpy.

Ejemplo

```
Public Sub IXMLSpyPlugIn_OnCommand(ByVal nID As Long, ByVal pXMLSpy As Object)
    If (Not (pXMLSpy Is Nothing)) Then
        Dim objDlg
        Dim objDoc As XMLSpyLib.Document
        Dim objSpy As XMLSpyLib.Application
        Set objSpy = pXMLSpy

        If nID = 3 Or nID = 5 Then
            Set objDlg = CreateObject("MSComDlg.CommonDialog")
            objDlg.Filter = "XML Files (*.xml)|*.xml|All Files (*.*)|*.*||"
            objDlg.FilterIndex = 1
            objDlg.ShowOpen

            If Len(objDlg.FileName) > 0 Then
                Set objDoc = objSpy.Documents.OpenFile(objDlg.FileName, False)
                Set objDoc = Nothing
            End If
        End If

        If nID = 4 Or nID = 6 Then
            Set objDlg = CreateObject("MSComDlg.CommonDialog")
            objDlg.Filter = "All Files (*.*)|*.*||"
            objDlg.Flags = cdlOFNPathMustExist
            objDlg.ShowSave

            If Len(objDlg.FileName) > 0 Then
                Set objDoc = objSpy.ActiveDocument

                If Not (objDoc Is Nothing) Then
                    objDoc.SetPathName objDlg.FileName
                    objDoc.Save
                    Set objDoc = Nothing
                End If
            End If
        End If

        Set objSpy = Nothing
    End If
End Sub
```


© 2016-2022 Altova GmbH

29.2.5.2 OnUpdateCommand

Declaración: `OnUpdateCommand(nID as long, pXMLSpy as IDispatch) as SPYUpdateAction`

Descripción
Al método `OnUpdateCommand()` se le llama cada vez que se debe establecer el estado visible de un botón o de un comando de menú. `nID` almacena el ID del comando definido por el elemento `ID` del correspondiente elemento `UIElement`

`pXMLSpy` almacena una referencia a la interfaz de envío del objeto `Application`.

Los valores devueltos posibles para establecer el estado de actualización son:

```
spyEnable        = 1
spyDisable             = 2
spyCheck         = 4
spyUncheck       = 8
```

Ejemplo

```
Public Function IXMLSpyPlugIn_OnUpdateCommand(ByVal nID As Long, ByVal pXMLSpy As Object)
As SPYUpdateAction
    IXMLSpyPlugIn_OnUpdateCommand = spyDisable

    If (Not (pXMLSpy Is Nothing)) Then
        Dim objSpy As XMLSpyLib.Application
        Set objSpy = pXMLSpy

        If nID = 3 Or nID = 5 Then
            IXMLSpyPlugIn_OnUpdateCommand = spyEnable
        End If
        If nID = 4 Or nID = 6 Then
            If objSpy.Documents.Count > 0 Then
                IXMLSpyPlugIn_OnUpdateCommand = spyEnable
            Else
                IXMLSpyPlugIn_OnUpdateCommand = spyDisable
            End If
        End If
    End If
End Function
```

29.2.5.3 OnEvent

Declaración: `OnEvent(nEventID as long, arrayParameters as SAFEARRAY(VARIANT), pXMLSpy as IDispatch) as VARIANT`

Descripción
Al método `OnEvent()` se le llama cada vez que se genera un evento desde XMLSpy.

Los valores posibles para `nEventID` son:

On_BeforeStartEditing	= 1
On_EditingFinished	= 2
On_FocusChanged	= 3
On_Beforedrag	= 4
On_BeforeDrop	= 5
On_OpenProject	= 6
On_OpenDocument	= 7
On_CloseDocument	= 8
On_SaveDocument	= 9

Eventos disponibles desde la versión 4r4 de XMLSpy:

On_DocEditDragOver	= 10
On_DocEditDrop	= 11
On_DocEditKeyDown	= 12
On_DocEditKeyUp	= 13
On_DocEditKeyPressed	= 14
On_DocEditMouseMove	= 15
On_DocEditButtonUp	= 16
On_DocEditButtonDown	= 17
On_DocEditContextMenu	= 18
On_DocEditPaste	= 19
On_DocEditCut	= 20
On_DocEditCopy	= 21
On_DocEditClear	= 22
On_DocEditSelectionChanged	= 23

Eventos disponibles desde la versión 2004 de XMLSpy:

On_DocEditDragOver	= 10

© 2016-2022 Altova GmbH

Eventos disponibles desde la versión 2004r4 de XMLSpy (biblioteca de tipos versión 1.4):

On_BeforeOpenProject	= 25
On_BeforeOpenDocument	= 26
On_BeforeSaveDocument	= 27
On_BeforeCloseDocument	= 28
On_ViewActivation	= 29
On_DocEditKeyboardEvent	= 30
On_DocEditMouseEvent	= 31

Eventos disponibles desde la versión 2006 SP1 de XMLSpy (biblioteca de tipos versión 1.5):

On_BeforeValidate	= 32

Eventos disponibles desde la versión 2007 de XMLSpy (biblioteca de tipos versión 1.6):

On_BeforeShowSuggestions	= 33
On_ProjectOpened	= 34
On_Char	= 35

Eventos disponibles desde la versión 2009 de XMLSpy (biblioteca de tipos versión 2.2):

On_Initialize	= 36
On_Running	= 37
On_Shutdown	= 38

Eventos disponibles desde la versión 2012 de XMLSpy (biblioteca de tipos versión 2.8):

On_AuthenticBeforeSave	= 39
On_AuthenticContextMenuActivated	= 40
On_AuthenticLoad	= 41
On_AuthenticToolbarButtonClicked	= 42
On_AuthenticToolbarButtonExecuted	= 43
On_AuthenticUserAddedXMLNode	= 44

Los eventos se llaman igual que en el entorno de scripting de XMLSpy. Para los complementos para entornos IDe los nombres utilizados son irrelevantes. Los eventos se identifican con el valor `ID`.

`arrayParameters` es una matriz que se rellena con los parámetros del evento generado actualmente. El orden, el tipo y el significado de cada parámetro está disponible en el entorno de scripting de XMLSpy. El módulo Eventos de un proyecto de scripting contiene funciones predefinidas para todos los eventos de todas las versiones anteriores a la versión 4.4. Los parámetros que se pasan a las funciones predefinidas son idénticos a los elementos de matriz del parámetro `arrayParameters`.

Los eventos generados desde la vista Authentic de XMLSpy no pasan ningún parámetro directamente. En su lugar se usan objetos de *evento*. A los objetos de evento se puede acceder desde el objeto *Document* del documento activo.

`pXMLSpy` almacena una referencia a la interfaz de envío del objeto `Application` de XMLSpy.

Si el valor devuelto de `OnEvent()` está definido, entonces el complemento para entornos IDE no obtendrá este evento después. Tampoco un controlador de eventos dentro del entorno de scripting. Recuerde que todos los complementos para entornos IDE obtienen/procesan el evento antes que el entorno de scripting.

29.2.5.4 GetUIModifications

Declaración: `GetUIModifications()` as String

Descripción
Al método `GetUIModifications()` se le llama durante la inicialización del complemento para obtener los datos de configuración XML que definen los cambios en la interfaz gráfica de XMLSpy. Al método se le llama cuando el complemento se carga por primera vez y cada vez que se inicia XMLSpy.

Consulte el apartado *Configuración XML* 1670 para ver cómo se modifica la interfaz gráfica.

Ejemplo

```
Public Function IXMLSpyPlugIn_GetUIModifications() As String
    ' GetUIModifications() obtiene el archivo XML con las modificaciones de la interfaz
gráfica
    ' del archivo config.xml de la carpeta del complemento
    Dim strPath As String
    strPath = App.Path

    If Len(strPath) > 0 Then
        Dim fso As New FileSystemObject
        Dim file As file

        Set file = fso.GetFile(strPath & "\config.xml")

        If (Not (file Is Nothing)) Then
            Dim stream As TextStream
            Set stream = file.OpenAsTextStream(ForReading)

            ' esto reemplaza el símbolo '**path**' de archivo XML
            ' con la ruta de acceso de instalación del complemento para obtener el archivo
de imagen
            Dim strMods As String
            strMods = stream.ReadAll
            strMods = Replace(strMods, "**path**", strPath)

            IXMLSpyPlugIn_GetUIModifications = strMods
        Else
            IXMLSpyPlugIn_GetUIModifications = ""
        End If
    End If
End Function
```


© 2016-2022 Altova GmbH

29.2.5.5 GetDescription

Declaración: `GetDescription()` as String

Descripción

`GetDescription()` sirve para definir la cadena *Description* para las entradas del complemento que aparecen en el cuadro de diálogo "Personalizar".

Ejemplo

```
Public Function IXMLSpyPlugIn_GetDescription() As String
    IXMLSpyPlugIn_GetDescription = "Complemento de muestra para XMLSpy;Este complemento
demuestra la implementación de un DLL Visual Basic sencillo como complemento para XMLSpy."
End Function
```

29.3 Application API

The COM-based API of XMLSpy (also called the Application API from now on) enables other applications to use the functionality of XMLSpy. As a result, it is possible to automate a wide range of tasks, from validating an XML file to modifying complex XML content (with the XMLData[1939] interface).

XMLSpy and its Application API follow the common specifications for automation servers set out by Microsoft. It is possible to access the methods and properties of the Application API from common development environments, such as those using C#, C++, VisualBasic, and Delphi, and with scripting languages like JScript and VBScript.

Execution environments for the Application API

The Application API can be accessed from the following execution environments:

- External programs (described below[1686] and in the Overview[1687] part of this section)
- From within the built-in Scripting Editor of XMLSpy. For a description of the scripting environment, see the section, Scripting Editor[1641].
- XMLSpy allows you to create and integrate your own plug-ins into the application using a special interface for plug-ins. A description of how to create plug-ins is given in the section IDE Plug-ins[1669].
- Via an ActiveX Control, which is available if the integration package[2052] is installed. For more information, see the section ActiveX Integration[2052].

External programs

In the Overview[1687] part of this section, we describe how the functionality of XMLSpy can be accessed and automated from external programs.

Using the Application API from outside XMLSpy requires an instance of XMLSpy to be started first. How this is done depends on the programming language used. See the section, Programming Languages[1688], for information about individual languages.

Essentially, XMLSpy will be started via its COM registration. Then the `Application` object associated with the XMLSpy instance is returned. Depending on the COM settings, an object associated with an already running XMLSpy can be returned. Any programming language that supports creation and invocation of COM objects can be used. The most common of these are listed below.

- JScript and VBScript[1698] script files have a simple syntax and are designed to access COM objects. They can be run directly from a DOS command line or with a double click on Windows Explorer. They are best used for simple automation tasks.
- C#[1701] is a full-fledged programming language that has a wide range of existing functionality. Access to COM objects can be automatically wrapped using C#..
- C++ provides direct control over COM access but requires relatively larger amounts of code than the other languages.
- Java[1711]: Altova products come with native Java classes that wrap the Application API and provide a full Java look-and-feel.
- Other programming languages that make useful alternatives are: Visual Basic for Applications, Perl, and Python.

© 2016-2022 Altova GmbH

Programming points

The following limitations must be considered in your client code:

- Be aware that if your client code crashes, instances of XMLSpy may still remain in the system.
- Don't hold references to objects in memory longer than you need them, especially those from the `XMLData` interface. If the user interacts between two calls of your client, then there is no guarantee that these references are still valid.
- Don't forget to disable dialogs if the user interface is not visible.
- See Error handling in JScript[1693] (and in C#[1709] and Java[1717]) for details of how to avoid annoying error messages.
- Free references explicitly if you are using C# or C++.

This documentation

This documentation section about the Application API is broadly divided into two parts.

- The first part consists of an Overview[1687], which describes the object model for the API and explains how the API is accessed via various programming languages[1688].
- The second part is a reference section (Interfaces[1723] and Enumerations[1987]) that contains descriptions of the interface objects of the Application API.

29.3.1 Overview

This overview of the Application API is organized as follows:

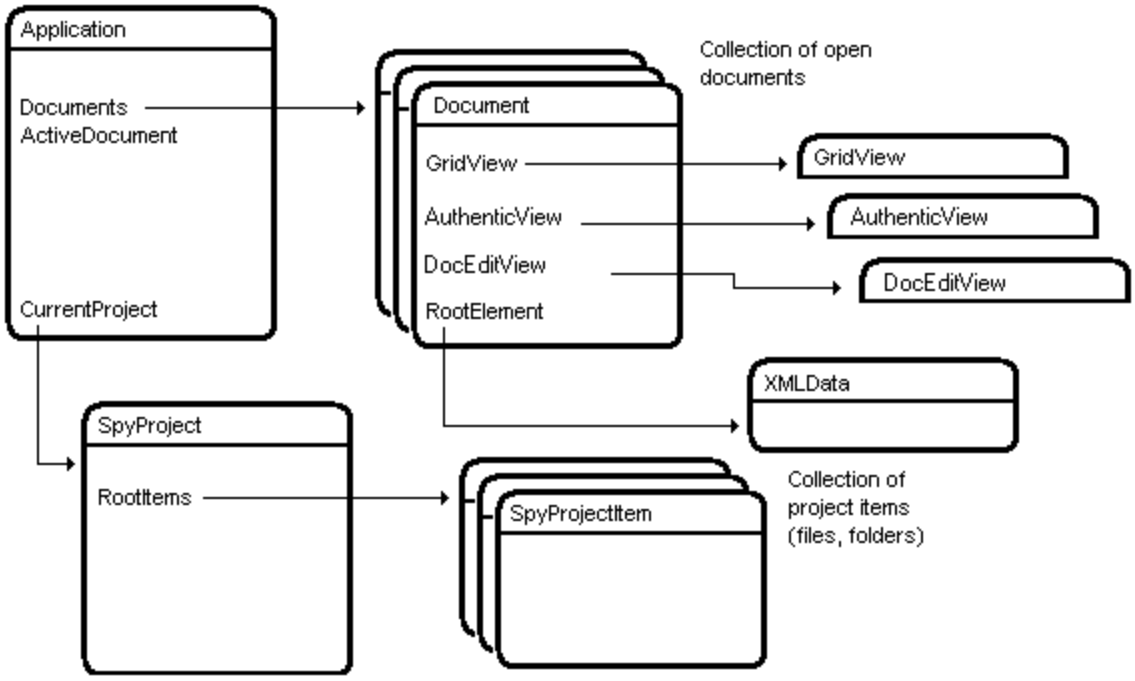
- The Object Model[1687] describes the relationships between the objects of the Application API.
- Programming Languages[1688] explains how the most commonly used programming languages (JScript, VBScript, C#, and Java) can be used to access the functionality of the Application API. Code listings from the example files supplied with your application package are used to describe basic mechanisms.
- The DOM and XMLData[1719] explains the relationship between the Application API's XMLData interface and the DOM.
- Obsolete: Authentic View Row Operations[1722] supplies information about obsolete objects for Authentic View table row operations.
- Obsolete: Authentic View Editing Operations[1722] supplies information about obsolete objects for Authentic View editing operations.

29.3.1.1 Object Model

The starting point for every application which uses the Application API is the `Application`[1724] object. This object contains general methods like import/export support and references to the open documents and any open project.

The `Application` object is created differently in various programming languages. In scripting languages such as JScript or VBScript, this involves calling a function which initializes the application's COM object. For examples, see the Programming Languages[1688] section.

The picture below shows the links between the main objects of the Application API:



The application object consists of the following parts:

1. Document collection and reference to the active document.
2. Reference to current project and methods for creating and opening projects.
3. Methods to support the export to and import from databases, text files, and Word documents.
4. URL management.
5. Methods for macro menu items.

Once you have created an Application object you can start using the functionality of XMLSpy. In most cases, you either open a project and access the documents from there or you directly open a document via the Documents[1843] interface.

29.3.1.2 Programming Languages

Programming languages differ in the way they support COM access. A few examples for the most frequently used languages (*links below*) will help you get started. The code listings in this section show how basic functionality can be accessed. The files in the `API` subfolder of the `Examples` folder can be used to test this functionality:

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Users\<usuario>\Documentos\Altova\XMLSpy\2023\Examples`

© 2016-2022 Altova GmbH

JScript

The JScript listings demonstrate the following basic functionality:

- Start application or attach to a running instance 1690
- Simple document access 1691
- Iteration 1692
- Error handling 1693
- Events 1693
- Import and export of data 1694

VBScript

VBScript is different than JScript only syntactically; otherwise it works in the same way. The link below goes to an example of how VBScript can be used. For more information, refer to the JScript examples 1690.

- Events 1698: Shows how events are handled using VBScript.

C#

C# can be used to access the Application API functionality. The code listings show how to access the API for certain basic functionality.

- Start XMLSpy 1706: Starts XMLSpy, which is registered as an automation server, or activates the program if XMLSpy is already running.
- Open OrgChart.pxf 1707: Locates one of the example documents installed with XMLSpy and opens it. If this document is already open it becomes the active document.
- OnDocumentOpened Event On/Off 1710: Shows how to listen to XMLSpy events. When turned on, a message box will pop up after a document has been opened.
- Open ExpReport.xml 1707: Opens another example document.
- Toggle View Mode 1708: Changes the view of all open documents between Text View and Authentic View. The code shows how to iterate through open documents.
- Validate 1709: Validates the active document and shows the result in a message box. The code shows how to handle errors and COM output parameters.
- Shutdown XMLSpy 1706: Stops XMLSpy.

Java

The XMLSpy API can be accessed from Java code. The Java sub-section of this section 1711 explains how some basic XMLSpy functionality can be accessed from Java code. It is organized into the following sub-sections:

- Mapping Rules for the Java Wrapper 1711
- Example Java Project 1712
- Application Startup and Shutdown 1716
- Simple Document Access 1717
- Iterations 1717
- Use of Out-Parameters 1717
- Event Handlers 1718

29.3.1.2.1 JScript

This section contains listings of JScript code that demonstrate the following basic functionality:

- Start application or attach to a running instance 1690
- Simple document access 1691
- Iteration 1692
- Error handling 1693
- Events 1693
- Import and export of data 1694

Example files

The code listings in this section are available in example files that you can test as is or modify to suit your needs. The JScript example files are located in the `JScript` subfolder of the API Examples folder:

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Users\<usuario>\Documentos\Altova\XMLSpy\2023\Examples`

The example files can be run in one of two ways:

- *From the command line:* Open a command prompt window, change the directory to the path above, and type the name of one of the example scripts (for example, `Start.js`).
- *From Windows Explorer:* In Windows Explorer, browse for the JScript file and double-click it.

The script is executed by Windows Script Host that is packaged with Windows operating system. For more information about Windows Script Host, refer to MSDN documentation (https://msdn.microsoft.com).

29.3.1.2.1.1 Start Application

The JScript below starts the application and shuts it down. If the COM object of the 32-bit XMLSpy cannot be found, the code attempts to get the COM object of the 64-bit application; otherwise, an error is thrown. If an instance of the application is already running, the running instance will be returned.

Nota: en la versión de 32 bits de XMLSpy, el nombre registrado o identificador programático (ProgId) del objeto COM es `XMLSpy.Application`. Para la versión de 64 bits de XMLSpy, el nombre es `XMLSpy_x64.Application`. Sin embargo, debe tener en cuenta que el programa que realiza las llamadas accede a las entradas de registro CLASSES de su propio subárbol o grupo (de 32 o 64 bits). Por tanto, si ejecuta scripts usando la línea de comandos estándar y el explorador de Windows en una instancia Windows de 64 bits, el programa accederá a las entradas de registro de 64 bits, que apuntan a la versión de 64 bits de XMLSpy. Por eso, si tiene instaladas tanto la versión de 32 bits como la versión de 64 bits de XMLSpy necesitará dar un rodeo para llamar a la versión de 32 bits. Por ejemplo, si el programa que realiza las llamadas es Windows Scripting Host, haga lo siguiente:

```
wscript.exe "C:\Users\...
```


© 2016-2022 Altova GmbH

```
\Documents\Altova\StyleVision2023\StyleVisionExamples\API\JScript\start.js"wscript.exe
"C:\Users\...\Documents\Altova\XMLSpy2023\Examples\API\JScript\start.js"
```

```
// Initialize application's COM object. This will start a new instance of the application
and
// return its main COM object. Depending on COM settings, the main COM object of an
already
// running application might be returned.

try {   objSpy = WScript.GetObject("", "XMLSpy.Application");   }
catch(err) {}

if( typeof( objSpy ) == "undefined" )
{
   try   {   objSpy = WScript.GetObject("", "XMLSpy_x64.Application")   }
   catch(err)
   {
      WScript.Echo( "Can't access or create XMLSpy.Application" );
      WScript.Quit();
   }
}

// if newly started, the application will start without its UI visible. Set it to
visible.
objSpy.Visible = true;

WScript.Echo(objSpy.Edition + " has successfully started. ");

objSpy.Visible = false;      // will shutdown application if it has no more COM
connections
//objSpy.Visible = true;   // will keep application running with UI visible
```

The JScript code listed above is available in the sample file `Start.js` (see Example Files 1690).

29.3.1.2.1.2 Simple Document Access

After you have started the application as shown in Start Application 1690, you will most likely want to programmatically open a document in order to work with it. The JScript code listing below illustrates how to open two documents from the XMLSpy Examples folder and set one of them as the active document.

```
// Locate examples via USERPROFILE shell variable. The path needs to be adapted to major
release versions.
objWshShell = WScript.CreateObject("WScript.Shell");
majorVersionYear = objSpy.MajorVersion + 1998
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\My
Documents\\Altova\\XMLSpy" + majorVersionYear + "\\Examples\\";

// Tell XMLSpy to open two documents. No dialogs
```

```
objDoc1 = objSpy.Documents.OpenFile(strExampleFolder + "OrgChart.pxf", false);
objSpy.Documents.OpenFile(strExampleFolder + "ExpReport.xml", false);

// The document currently active can be easily located.
objDoc2 = objSpy.ActiveDocument;

// Let us make sure that the document is shown in grid view.
objDoc2.SwitchViewMode(0);   // SPYViewModes.spyViewGrid = 0

// Now switch back to the document opened first
objDoc1.SetActiveDocument();
```

The JScript code listed above is available in the sample file `DocumentAccess.js` (see Example Files 1690).

29.3.1.2.1.3 Iteration

The JScript listing below shows how to iterate through the open documents. It is assumed that you have already started the application and opened some documents as shown in the previous sections.

```
// go through all open documents using a JScript Enumerator
bRequiresSaving = false;
for (var iterDocs = new Enumerator(objSpy.Documents); !iterDocs.atEnd();
iterDocs.moveNext())
{
   if (iterDocs.item().IsModified)
      bRequiresSaving = true;

   var strErrorText = new Array(1);
   var nErrorNumber = new Array(1);
   var errorData = new Array(1);

   if (!iterDocs.item().IsValid(strErrorText, nErrorNumber, errorData))
   {
       var text = strErrorText;
      // access that XMLData object only if filled in
       if (errorData[0] != null)
         text += "(" + errorData[0].Name + "/" + errorData[0].TextValue + ")";

      WScript.Echo("Document \"" + iterDocs.item().Name +"\" validation error[" +
nErrorNumber + "]: " + text);
   }
   else
   {
      // The COM call succeeded and the document is valid.
      WScript.Echo("Document \"" + iterDocs.item().Name + "\" is valid.");
   }
}

// go through all open documents using index-based access to the document collection
```

 © 2016-2022 Altova GmbH

```
for (i = objSpy.Documents.Count; i > 0; i--)
   objSpy.Documents.Item(i).Close(false);
```

The JScript code listed above is available in the sample file `DocumentAccess.js` (see Example Files 1690).

29.3.1.2.1.4 Error Handling

The Application API returns errors in two different ways:

- The `HRESULT` returned by every API method
- The `IErrorInfo` interface of the Application API

Every API method returns an `HRESULT`. This return value gives the caller information about errors during execution of the method. If the call was successful, the return value is `S_OK`. The `HRESULT` option is commonly used in C/C++ programs.

However, programming languages such as VisualBasic and scripting languages (and other high-level development environments) don't give the programmer access to the `HRESULT` return of a COM call. Such languages use the `IErrorInfo` interface, which is also supported by the Application API. If an error occurs, the Application API creates a new object that implements the `IErrorInfo` interface. The information provided by the `IErrorInfo` interface is imported by the development environment into its own error-handling mechanism.

For example, the JScript code listing below causes an error to be thrown by incorrectly declaring an array. Additional information about the error object is provided by its properties `number` and `description`.

```
try {
   var arr = new Array(-1);
}
catch (err) {
   WScript.Echo("Error : (" + (err.number & 0xffff) + ")" + err.description);
}
```

29.3.1.2.1.5 Events

COM specifies that a client must register itself at a server for callbacks using the connection point mechanism. The automation interface for XMLSpy defines the necessary event interfaces. The way to connect to those events depends on the programming language you use in your client. The following code listing shows how this is done using JScript.

The method `WScript.ConnectObject` is used to receive events.

```
// The event-handler function
function DocEvent_OnBeforeCloseDocument(objDocument)
{
   WScript.Echo("Received event - before closing document");
}
```

```
// Create or connect to XMLSpy (or Authentic Desktop)
try
{
   // Create the environment and XMLSpy (or Authentic Desktop)
   objWshShell = WScript.CreateObject("WScript.Shell");
   objFSO = WScript.CreateObject("Scripting.FileSystemObject");
   objSpy = WScript.GetObject("", "XMLSpy.Application");

// If only Authentic Desktop is installed (and XMLSpy is not installed) use:
// objSpy = WScript.GetObject("", "AuthenticDesktop.Application")

}
catch(err)
   { WScript.Echo ("Can't create WScript.Shell object or XMLSpy"); }

// Create document object and connect to its events
objSpy.Visible = true;
majorVersionYear = objSpy.MajorVersion + 1998
docPath = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\Documents\\Altova\
\XMLSpy" + majorVersionYear + "\\Examples\\ExpReport.xml";
objDoc = objSpy.Documents.OpenFile (docPath, false);
WScript.ConnectObject(objDoc, "DocEvent_");

// Keep running while waiting for the event
// In the meanwhile close this document in XMLSpy (or Authentic Desktop) manually
WScript.Echo ("Sleeping for 10 seconds ...");
WScript.Sleep (10000);

objDoc = null;
WScript.Echo ("Stopped listening for event");
objSpy.Quit();
```

29.3.1.2.1.6 Import and Export of Data

Before you implement your import and export tasks with the Application API, it is good practice to test the connections, parameters, SQL queries and so on in XMLSpy. In this way you are able to verify the results and make quick adjustments to import or export parameters. Most of the methods for importing and exporting data are placed in the `Application`[1724] object; the remaining functions are accessible via the `Document`[1813] interface.

There is some preparatory work necessary before the actual import or export can be started. Every import/export job consists of two parts. You need to define a connection to your data and the specific behavior for the import/export process. In case of an import, the connection is either a database, a text-file, or a Word document. The main issue is which data (columns) to imported into XMLSpy. In case of an export, the connection is either a database or a text file. Specify which data (elements of the XML file) and additional parameters (for example, automatic key generation or number of sub-levels) to use from the XML-structure.

The properties in the `DatabaseConnection`, `TextImportExportSettings` and `ExportSettings` interfaces have default values. See the corresponding descriptions in the `Interfaces`[1723] chapter for further information.

© 2016-2022 Altova GmbH

The sub-sections of this section describe each of these operations in detail.

- Import from Database 1695
- Export to Database 1696
- Import from Text 1696
- Export to Text 1697

Given below are the steps to establish a connection to an existing database for import:

1. Use a `DatabaseConnection` 1803 object and set the following:
   - The method `Application.GetDatabaseSettings` 1731 returns a new object for a database connection: `objImpSettings = objSpy.GetDatabaseSettings();`
   - You have to set either an ADO connection string, `objImpSettings.ADOConnection = strADOConnection` or the path to an existing database file: `objImpSettings.File = strExampleFolder + "Tutorial\\Company.mdb";`
   - To complete the settings you create an SQL SELECT statement to define the data to be queried: `objImpSettings.SQLSelect = "SELECT * FROM Address";`
2. Call `Application.GetDatabaseImportElementList` 1731 to get a collection of the resulting columns of the SQL query: `objElementList = objSpy.GetDatabaseImportElementList(objImpSettings);` This collection gives you the opportunity to control which columns should be imported and specify the datatype of the new elements. Each item of the collection represents one column to import. If you remove an item, the corresponding column will not be imported. You can additionally modify the `ElementListItem.ElementKind` 1853 property to set the datatype of the XML elements for each column. Please consider that `GetDatabaseImportElementList()` executes the SQL query and could initiate a time-consuming call. To avoid this, it is possible to pass a null-pointer as the second parameter to `ImportFromDatabase();` this imports all columns as plain XML elements.
3. Start the import with `Application.ImportFromDatabase` 1735: `objImpDocFromDB = objSpy.ImportFromDatabase(objImpSettings,objElementList);`

```
// Locate examples via USERPROFILE shell variable.
objWshShell = WScript.CreateObject("WScript.Shell");
majorVersionYear = objSpy.MajorVersion + 1998
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\My
Documents\\Altova\\XMLSpy" + majorVersionYear + "\\Examples\\";

try
{
   // specify the source of data import
   objImpSettings = objSpy.GetDatabaseSettings();
   objImpSettings.File = strExampleFolder + "Tutorial\\Company.mdb";
   objImpSettings.SQLSelect = "SELECT * FROM Address";

   // column filter
   objElementList = objSpy.GetDatabaseImportElementList(objImpSettings);

   // import into a new XML file
   objImpDocFromDB = objSpy.ImportFromDatabase(objImpSettings,objElementList);
}
catch(err)
{
   WScript.Echo("Error importing from database.\n\n" +
                    "Error: " + (err.number & 0xffff) + "\n" +
```

```
                    "Description: " + err.description);
}
```

The JScript code listed above is available in the sample file `ImportExport.js` (see Example Files[1690]).

To export data to a database, carry out the steps below:

1. Use a `DatabaseConnection`[1803] object and set the necessary properties. All properties except `SQLSelect` are important for the export. `ADOConnection or File` defines the target for the output. You need to set only one of them.
2. Fill an `ExportSettings`[1853] object with the required values. These properties are the same options as those available in the export dialog of XMLSpy. Select the menu option **Convert | Export to Text files/Database** to see the options and try a combination of export settings. After that it is easy to transfer these settings to the properties of the interface. Call `Application.GetExportSettings`[1732] to get an `ExportSettings` object: `objExpSettings = objSpy.GetExportSettings()`
3. Build an element list with `Document.GetExportElementList`[1831]. The element list enables you to eliminate XML elements from the export process. It also gives you information about the record and field count in the `RecordCount` and `FieldCount` properties. Set the `ExportSettings.ElementList`[1854] property to this collection. It is possible to set the element list to `null/Nothing` (default) to export all elements.
4. Call `Document.ExportToDatabase`[1825] to execute the export. The description of the `ExportToDatabase` method contains also a code example for a database export.

```
   // set the behaviour of the export with ExportSettings
   objExpSettings = objSpy.GetExportSettings()

   //set the destination with DatabaseConnection
   objDB = objSpy.GetDatabaseSettings();
   objDB.CreateMissingTables = true;
   objDB.CreateNew = true;
   objDB.File = "C:\\Temp\\Export.mdb";

try
{
   objImpDocFromDB.ExportToDatabase(objImpDocFromDB.RootElement, objExpSettings, objDB);
}
catch(err)
{
   WScript.Echo("Error exporting to database.\n\n" +
                    "Error: " + (err.number & 0xffff) + "\n" +
                    "Description: " + err.description);
}
```

The JScript code listed above is available in the sample file `ImportExport.js` (see Example Files[1690]).

Importing data from a text file is similar to the import from a database. You must use other interfaces (described in steps 1 to 3 below) with different methods and properties:

1. Use a `TextImportExportSettings`[1897] object and set the properties: The method `Application.GetTextImportExportSettings`[1733] returns a new object to specify a text file for import:

© 2016-2022 Altova GmbH

`objImpSettings = objSpy.GetTextImportExportSettings();` You have to set at least the `ImportFile` property to the path of the file for the import. Another important property is `HeaderRow`. Set it to `False` if the text file does not contain a leading line as a header row: `objImpSettings.ImportFile = strExampleFolder + "Tutorial\\Shapes.txt";`

2. Call `Application.GetTextImportElementList`[1733] to get a collection of all columns inside the text file: `objElementList = objSpy.GetTextImportElementList(objImpSettings);`
3. Start the import with `Application.ImportFromText`[1736]: `objImpDocFromText = objSpy.ImportFromText(objImpSettings,objElementList);`

```
try
{
   // specify the source of data import
   objImpSettings = objSpy.GetTextImportExportSettings();
   objImpSettings.ImportFile = strExampleFolder + "Tutorial\\Shapes.txt";
   objImpSettings.HeaderRow = false;

   // column filter
   objElementList = objSpy.GetTextImportElementList(objImpSettings);

   // import into a new XML file
   objImpDocFromText = objSpy.ImportFromText(objImpSettings,objElementList);
}
catch(err)
{
   WScript.Echo("Error importing from text file.\n\n" +
                "Error: " + (err.number & 0xffff) + "\n" +
                "Description: " + err.description);
}
```

The JScript code listed above is available in the sample file `ImportExport.js` (see Example Files[1690]).

To export data to text, carry out the steps below:

1. Use a `TextImportExportSettings`[1897] object and set the necessary properties.
2. Fill an `ExportSettings`[1853] object with the required values. See Item 2 in Export to database[1696].
3. Build an element list with `Document.GetExportElementList`[1831]. See Item 3 in Export to database[1696].
4. Call `Document.ExportToText`[1826] to execute the export.

```
objExpSettings = objSpy.GetExportSettings();
objExpSettings.ElementList =
objImpDocFromText.GetExportElementList(objImpDocFromText.RootElement, objExpSettings);

objTextExp = objSpy.GetTextImportExportSettings();
objTextExp.HeaderRow = true;
objTextExp.DestinationFolder = "C:\\Temp";

try
{
```

```
      objImpDocFromText.ExportToText(objImpDocFromText.RootElement, objExpSettings,
objTextExp);
}
catch(err)
{
   WScript.Echo("Error exporting to text.\n\n" +
                    "Error: " + (err.number & 0xffff) + "\n" +
                    "Description: " + err.description);
}
```

The JScript code listed above is available in the sample file `ImportExport.js` (see Example Files[1690]).

29.3.1.2.2 VBScript

VBScript is syntactically different than JScript but works in the same way. This section contains a listing showing how events are used with VBScript[1698] and an example[1699].

For information about other functionality, refer to the JScript examples listed below:

- Start application or attach to a running instance[1690]
- Simple document access[1691]
- Iteration[1692]
- Error handling[1693]
- Import and export of data[1694]

29.3.1.2.2.1 Events

COM specifies that a client must register itself at a server for callbacks using the connection point mechanism. The automation interface for XMLSpy defines the necessary event interfaces. The way to connect to those events depends on the programming language you use in your client. The following code listing shows how this is done using VBScript.

The method `WScript.ConnectObject` is used to receive events.

To run this code, paste it into a file with .vbs extension, and either double-click in Windows Explorer, or run it from a command prompt.

```
' the event handler function
Function DocEvent_OnBeforeCloseDocument(objDocument)
   Call WScript.Echo("received event - before closing document")
End Function

' create or connect to XmlSpy
Set objWshShell = WScript.CreateObject("WScript.Shell")
Set objFSO = WScript.CreateObject("Scripting.FileSystemObject")
Set objSpy = WScript.GetObject("", "XMLSpy.Application")
```


© 2016-2022 Altova GmbH

```
' If only Authentic is installed (and XMLSpy is not installed) use:
' Set objSpy = WScript.GetObject("", "AuthenticDesktop.Application")
' If only XMLSpy 64-bit is intalled, use:
' Set objSpy = WScript.GetObject("", "XMLSpy_x64.Application")

' create document object and connect to its events
objSpy.Visible = True

' Find out user's personal folder and locate one of the installed examples.
personalFolder = objWshShell.ExpandEnvironmentStrings("%UserProfile%")
majorVersionYear = objSpy.MajorVersion + 1998
xmlspyExamplesFolder = personalFolder & "\Documents\Altova\XMLSpy" & majorVersionYear  &
"\Examples\"
docPath = xmlspyExamplesFolder & "ExpReport.xml"

' open a document
Set objDoc = objSpy.Documents.OpenFile (docPath, False)
Call WScript.ConnectObject(objDoc, "DocEvent_")

' keep running while waiting on the event
' in the meantime close the document in XMLSPY manually
Call WScript.Echo ("sleeping for 10 seconds ...")
Call WScript.Sleep (10000)

Set objDoc = Nothing
Call WScript.Echo ("stopped listening for event")
Call objSpy.Quit
```

Nota: en la versión de 32 bits de XMLSpy, el nombre registrado o identificador programático (ProgId) del objeto COM es `XMLSpy.Application`. Para la versión de 64 bits de XMLSpy, el nombre es `XMLSpy_x64.Application`. Sin embargo, debe tener en cuenta que el programa que realiza las llamadas accede a las entradas de registro CLASSES de su propio subárbol o grupo (de 32 o 64 bits). Por tanto, si ejecuta scripts usando la línea de comandos estándar y el explorador de Windows en una instancia Windows de 64 bits, el programa accederá a las entradas de registro de 64 bits, que apuntan a la versión de 64 bits de XMLSpy. Por eso, si tiene instaladas tanto la versión de 32 bits como la versión de 64 bits de XMLSpy necesitará dar un rodeo para llamar a la versión de 32 bits. Por ejemplo, si el programa que realiza las llamadas es Windows Scripting Host, haga lo siguiente:

```
wscript.exe "C:\Users\...
\Documents\Altova\StyleVision2023\StyleVisionExamples\API\JScript\start.js"wscript.exe
"C:\Users\...\Documents\Altova\XMLSpy2023\Examples\API\JScript\start.js"
```

29.3.1.2.2.2 Example: Using Events

Authentic View supports event connection on a per-object basis. Implementation of this feature is based on COM connection points and is available in environments that support this mechanism.

The following example is a VBScript code example that shows how to use events from within a VBScript project.

```
' ------------------------------------------------------------------------
' VBScript example that demonstrates how to use events.
' ------------------------------------------------------------------------

' Event handler for OnSelectionChanged event of AuthenticView
Function AuthenticViewEvent_OnSelectionChanged(objAuthenticRange)
    If objAuthenticRange.FirstTextPosition <> objAuthenticRange.LastTextPosition Then
        Call WScript.Echo("Selection: " & objAuthenticRange.Text & vbNewLine & vbNewLine
& "Close this dialog.")
    Else
        Call WScript.Echo("Cursor position: " & objAuthenticRange.FirstTextPosition &
vbNewLine & vbNewLine & "Close this dialog.")
    End If
End Function

' Start/access XMLSpy and connect to its automation interface.
Set WshShell = WScript.CreateObject("WScript.Shell")
Set objSpy = GetObject("", "XMLSpy.Application")
' Make the UI of XMLSpy visible.
objSpy.Visible = True

' Find out user's personal folder and locate one of the installed XMLSpy examples.
personalFolder = WshShell.ExpandEnvironmentStrings("%UserProfile%")
majorVersionYear = objSpy.MajorVersion + 1998
xmlspyExamplesFolder = personalFolder & "\Documents\Altova\XMLSpy" & majorVersionYear  &
"\Examples\"
docPath = xmlspyExamplesFolder & "ExpReport.xml"

' Create object to access windows file system and test if the our document exists.
Set fso = CreateObject("Scripting.FileSystemObject")
If fso.FileExists(docPath) Then
    ' open the document
    Call objSpy.Documents.OpenFile(docPath, False)
    set objDoc = objSpy.ActiveDocument

    ' switch active document to authentic view
    objDoc.SwitchViewMode 4 ' spyViewAuthentic

    ' Register for connection point events on the authentic view of the active document.
    ' Any function with a valid event name prefixed with "AuthenticViewEvent_" will
    ' be called when the corresponding event gets triggered on the specified object.
    set objView = objDoc.AuthenticView
    Call WScript.ConnectObject(objView, "AuthenticViewEvent_")
    Call WScript.Echo("Events are connected." & vbNewLine & vbNewLine & "Now set or move
the cursor in XMLSpy." & vbNewLine & vbNewLine & "Close this dialog to shut down
XMLSpy.")

    ' To disconnect from the events delete the reference to the object.
    set objView = Nothing
Else
    Call WScript.Echo("The file " & docPath & " does not exist.")
End If
```

 © 2016-2022 Altova GmbH

```
' shut down XMLSpy when this script ends
objSpy.Visible = False
```

29.3.1.2.3 C#

The C# programming language can be used to access the Application API functionality. You could use Visual Studio 2012/2013/2015/2017/2019/2022 to create the C# code, saving it in a Visual Studio project. Create the project as follows:

1. In Microsoft Visual Studio, add a new project using **File | New | Project**.
2. Add a reference to the XMLSpy Type Library by clicking **Project | Add Reference**. The Add Reference dialog appears. Browse for the XMLSpy Type Library component, which is located in the XMLSpy application folder, and add it.
3. Enter the code you want.
4. Compile the code and run it.

Example C# project

Your XMLSpy package contains an example C# project, which is located in the `API\C#` subfolder of the `Examples` folder :

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Users\<usuario>\Documentos\Altova\XMLSpy\2023\Examples`

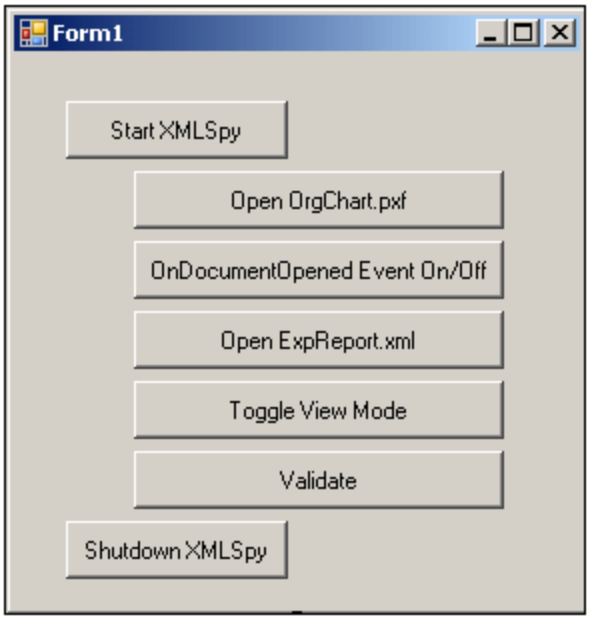
You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022. The code listing below shows how basic application functionality can be used. This code is similar to the example C# project in the API Examples folder of your application package, but might differ slightly.

Platform configuration

If you have a 64-bit operating system and are using a 32-bit installation of XMLSpy, you must add the x86 platform in the solution's Configuration Manager and build the sample using this configuration. A new x86 platform (for the active solution in Visual Studio) can be created in the New Solution Platform dialog (**Build | Configuration Manager | Active solution platform | <New...>**).

What the code listing below does

The example code listing below creates a simple user interface (*screenshot below*) with buttons that invoke basic XMLSpy operations:



- Start XMLSpy[1706]: Starts XMLSpy, which is registered as an automation server, or activates the application if it is already running.
- Open OrgChart.pxf[1707]: Locates one of the example documents installed with XMLSpy and opens it. If this document is already open it becomes the active document.
- OnDocumentOpened Event On/Off[1710]: Shows how to listen to XMLSpy events. When turned on, a message box will pop up after a document has been opened.
- Open ExpReport.xml[1707]: Opens another example document.
- Toggle View Mode[1708]: Changes the view of all open documents between Text View and Authentic View. The code shows how to iterate through open documents.
- Validate[1709]: Validates the active document and shows the result in a message box. The code shows how to handle errors and COM output parameters.
- Shut down XMLSpy[1706]: Stops XMLSpy.

You can modify the code (of the code listing below or of the example C# project in the API Examples folder) in any way you like and run it.

Compiling and running the example

In the API Examples folder, double-click the file `AutomateXMLSpy_VS2008.sln` or the file `AutomateXMLSpy_VS2010.sln` (to open in Visual Studio 2012/2013/2015/2017/2019/2022). Alternatively the file can be opened from within Visual Studio (with **File | Open | Project/Solution**). To compile and run the example, select **Debug | Start Debugging** or **Debug | Start Without Debugging**.

Code listing of the example

Given below is the C# code listing of the basic functionality of the form (`Form1.cs`) created in the `AutomateXMLSpy` example. Note that the code listed below might differ slightly from the code in the API Examples form.The listing below is commented for ease of understanding. Parts of the code are also presented separately in the sub-sections of this section, according to the Application API functionality they access.

The code essentially consists of a series of handlers for the buttons in the user interface shown in the screenshot above.

© 2016-2022 Altova GmbH

```
namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        // An instance of XMLSpy accessed via its automation interface
        XMLSpyLib.Application XMLSpy;

        // Location of examples installed with XMLSpy
        String strExamplesFolder;

        private void Form1_Load(object sender, EventArgs e)
        {
            // Locate examples installed with XMLSpy
            // REMARK: You might need to adapt this if you have a different major version
of the product
            strExamplesFolder = Environment.GetEnvironmentVariable("USERPROFILE") + "\\My
Documents\\Altova\\XMLSpy2012\\Examples\\";
        }

        // Handler for the "Start XMLSpy" button
        private void StartXMLSpy_Click(object sender, EventArgs e)
        {
            if (XMLSpy == null)
            {
                Cursor.Current = Cursors.WaitCursor;

                // If no XMLSpy instance is open, create one and make it visible
                XMLSpy = new XMLSpyLib.Application();
                XMLSpy.Visible = true;

                Cursor.Current = Cursors.Default;
            }
            else
            {
                // If an instance of XMLSpy is already running, make sure it's visible
                if (!XMLSpy.Visible)
                    XMLSpy.Visible = true;
            }
        }

        // Handler for the "Open OrgChart.pxf" button
        private void openOrgChart_Click(object sender, EventArgs e)
        {
            // Make sure there's a running XMLSpy instance, and that it's visible
            StartXMLSpy_Click(null, null);

            // Open one of the example files installed with the product
            XMLSpy.Documents.OpenFile(strExamplesFolder + "OrgChart.pxf", false);
        }
```

```
        // Handler for the "Open ExpReport.xml" button
        private void openExpReport_Click(object sender, EventArgs e)
        {
            // Make sure there's a running XMLSpy instance, and that it's visible
            StartXMLSpy_Click(null, null);

            // Open one of the sample files installed with the product.
            XMLSpy.Documents.OpenFile(strExamplesFolder + "ExpReport.xml", false);
        }

        // Handler for the "Toggle View Mode" button
        private void toggleView_Click(object sender, EventArgs e)
        {
            // Make sure there's a running XMLSpy instance, and that it's visible
            StartXMLSpy_Click(null, null);

            // Iterate through all open documents and toggle view between Text View and
Authentic View
            foreach (XMLSpyLib.Document doc in XMLSpy.Documents)
                if (doc.CurrentViewMode == XMLSpyLib.SPYViewModes.spyViewText)
                    doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewAuthentic);
                else
                    doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewText);
        }

        // Handler for the "Shutdown XMLSpy" button
        // Shut down the application instance by explicitly releasing the COM object
        private void shutdownXMLSpy_Click(object sender, EventArgs e)
        {
            if (XMLSpy != null)
            {
                // Allow shutdown of XMLSpy by releasing the UI
                XMLSpy.Visible = false;

                // Explicitly release the COM object
                try
                {
                    while (System.Runtime.InteropServices.Marshal.ReleaseComObject(XMLSpy)
> 0) ;
                }
                finally
                {
                    // Disallow subsequent access to this object
                    XMLSpy = null;
                }
            }
        }

        // Handler for button "Validate"
        private void validate_Click(object sender, EventArgs e)
        {
            // COM errors are returned to C# as exceptions. We use a try/catch block to
handle them.
            try
            {
```

 © 2016-2022 Altova GmbH

```
                // Method 'IsValid' is one of the few functions that uses output parameters
                // Use 'object' type for these parameters
                object strErrorText = "";
                object nErrorNumber = 0;
                object errorData = null;

                if (!XMLSpy.ActiveDocument.IsValid(ref strErrorText, ref nErrorNumber, ref
errorData))
                {
                    // The COM call succeeded but the document is not valid
                    // A detailed description of the problem is returned in strErrorText,
nErrorNumber and errorData
                    listBoxMessages.Items.Add("Document " + XMLSpy.ActiveDocument.Name + "
is not valid.");
                    listBoxMessages.Items.Add("\tErrorText  : " + strErrorText);
                    listBoxMessages.Items.Add("\tErrorNumber: " + nErrorNumber);
                    listBoxMessages.Items.Add("\tElement    : " + (errorData != null ?
((XMLSpyLib.XMLData)errorData).TextValue : "null"));
                }
                else
                {
                    // The COM call succeeded and the document is valid
                    listBoxMessages.Items.Add("Document " + XMLSpy.ActiveDocument.Name + "
is valid.");
                }
            }
            catch (Exception ex)
            {
                // The COM call was not successful
                // Probably no application instance has been started or no document is
open.
                listBoxMessages.Items.Add("Error validating active document: " +
ex.Message);
            }
        }



        // Event handler for OnDocumentOpened event
        private void handleOnDocumentOpened(XMLSpyLib.Document i_ipDocument)
        {
            MessageBox.Show("Document " + i_ipDocument.Name + " was opened!");
        }

        // Remember if the event handler is currently registered.
        private bool bEventHandlerIsRegistered = false;

        // Handler for button 'OnDocuemntOpened Event On/Off
        private void toggleOnDocumentOpenedEvent_Click(object sender, EventArgs e)
        {
            if (XMLSpy != null)
            {
                if (bEventHandlerIsRegistered)
                    XMLSpy.OnDocumentOpened -= new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
                else
```

```
                    XMLSpy.OnDocumentOpened += new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);

                bEventHandlerIsRegistered = !bEventHandlerIsRegistered;
            }
        }
    }
}
```

29.3.1.2.3.1 Add Reference to XMLSpy API

Add the application's type library as a reference in a .NET project as follows: With the .NET project open, click **Project | Add Reference**. Then browse for the type library, which is called `XMLSpy.tlb`, and is located in the XMLSpy application folder.

Then declare a variable to access the XMLSpy API:

```
// An instance of XMLSpy is accessed via its automation interface.
   XMLSpyLib.Application XMLSpy;
```

29.3.1.2.3.2 Application Startup and Shutdown

In the code snippets below, the methods `StartXMLSpy_Click` and `ShutdownXMLSpy_Click` are those assigned to buttons in the AutomateXMLSpy example[1701] that, respectively, start up and shut down the application. This example is located in the C# subfolder of the API Examples folder (*see the file Form1.cs*):

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Users\<usuario>\Documentos\Altova\XMLSpy\2023\Examples`

You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022.

Starting XMLSpy

The following code snippet from the AutomateXMLSpy example[1701] shows how to start up the application.

```
// Handler for the "Start XMLSpy" button
private void StartXMLSpy_Click(object sender, EventArgs e)
{
    if (XMLSpy == null)
    {
        Cursor.Current = Cursors.WaitCursor;

        // If no XMLSpy instance is running, we create one and make it visible
        XMLSpy = new XMLSpyLib.Application();
        XMLSpy.Visible = true;

        Cursor.Current = Cursors.Default;
```

 © 2016-2022 Altova GmbH

```
            }
            else
            {
                // If an instance of XMLSpy is already running, make sure it's visible
                if (!XMLSpy.Visible)
                    XMLSpy.Visible = true;
            }
        }
```

Shutting down XMLSpy

The following code snippet from the AutomateXMLSpy example[1701] shows how to shut down the application.

```
        // Handler for the "Shutdown XMLSpy" button
        // Shut down the application instance by explicitly releasing the COM object
         private void shutdownXMLSpy_Click(object sender, EventArgs e)
         {
            if (XMLSpy != null)
            {
                // Allow shutdown of XMLSpy by releasing the UI
                XMLSpy.Visible = false;

                // Explicitly release COM object
                try
                {
                    while (System.Runtime.InteropServices.Marshal.ReleaseComObject(XMLSpy)
> 0) ;
                }
                finally
                {
                    // Disallow subsequent access to this object
                    XMLSpy = null;
                }
            }
        }
```

29.3.1.2.3.3 Opening Documents

The code snippets below (from the AutomateXMLSpy example[1701]) show how two files are opened via two separate methods assigned to two buttons in the user interface. Both methods use the same Application API access mechanism: XMLSpy.Documents.OpenFile(string, boolean)[1846].

The AutomateXMLSpy example[1701] (*see the file Form1.cs*) is located in the C# subfolder of the API Examples folder:

Windows 7, Windows 8, Windows 10, Windows 11	C:\Users\<usuario>\Documentos\Altova\XMLSpy\2023\Examples

You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022.

Code snippet

```
// Handler for the "Open OrgChart.pxf" button
 private void openOrgChart_Click(object sender, EventArgs e)
 {
     // Make sure there's a running XMLSpy instance, and that it's visible
     StartXMLSpy_Click(null, null);

     // Open a file from the Examples folder installed with the product
     XMLSpy.Documents.OpenFile(strExamplesFolder + "OrgChart.pxf", false);
 }

 // Handler for the "Open ExpReport.xml" button
 private void openExpReport_Click(object sender, EventArgs e)
 {
     // Make sure there's a running XMLSpy instance, and that it's visible
     StartXMLSpy_Click(null, null);

     // Open a file from the Examples folder installed with the product
     XMLSpy.Documents.OpenFile(strExamplesFolder + "ExpReport.xml", false);
 }
```

The file opened last will be the active file.

29.3.1.2.3.4 Iterating through Open Documents

The code snippet below (from the AutomateXMLSpy example[1701]; *see the file Form1.cs*) shows how to iterate through open documents. A condition is then tested within the iteration loop, and the document view is switched between Text View and Authentic View.

```
// Handler for the "Toggle view mode" button
 private void toggleView_Click(object sender, EventArgs e)
 {
     // Make sure there's a running XMLSpy instance, and that it's visible
     StartXMLSpy_Click(null, null);

     // Iterate through open documents and toggle current view between text and
authentic view.
     foreach (XMLSpyLib.Document doc in XMLSpy.Documents)
         if (doc.CurrentViewMode == XMLSpyLib.SPYViewModes.spyViewText)
             doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewAuthentic);
         else
             doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewText);
 }
```

The AutomateXMLSpy example[1701] example is located in the C# subfolder of the API Examples folder:

© 2016-2022 Altova GmbH

Windows 7, Windows 8, Windows 10, Windows 11	C:\Users\<usuario>\Documentos\Altova\XMLSpy\2023\Examples

You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022.

29.3.1.2.3.5 Errors and COM Output Parameters

The code snippet below (from the AutomateXMLSpy example[1701]) shows how to handle errors and COM output parameters. The method XMLSpy.ActiveDocument.IsValid(ref strErrorText, ref nErrorNumber, ref errorData)[1833] uses output parameters that are used, in the code snippet below, to generate an error-message text.

The AutomateXMLSpy example[1701] (*see the file Form1.cs*) is located in the C# subfolder of the API Examples folder:

Windows 7, Windows 8, Windows 10, Windows 11	C:\Users\<usuario>\Documentos\Altova\XMLSpy\2023\Examples

You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022.

Code snippet

```
        // Handler for button "Validate"
        private void validate_Click(object sender, EventArgs e)
        {
            // COM errors are returned to C# as exceptions. We use a try/catch block to
handle them.
            try
            {
                // Method 'IsValid' is one of the few functions that uses output parameters
                // Use 'object' type for these parameters
                object strErrorText = "";
                object nErrorNumber = 0;
                object errorData = null;

                if (!XMLSpy.ActiveDocument.IsValid(ref strErrorText, ref nErrorNumber, ref
errorData))
                {
                    // The COM call succeeded but the document is not valid
                    // A detailed description of the problem is returned in strErrorText,
nErrorNumber and errorData
                    listBoxMessages.Items.Add("Document " + XMLSpy.ActiveDocument.Name + "
is not valid.");
                    listBoxMessages.Items.Add("\tErrorText   : " + strErrorText);
                    listBoxMessages.Items.Add("\tErrorNumber: " + nErrorNumber);
                    listBoxMessages.Items.Add("\tElement    : " + (errorData != null ?
((XMLSpyLib.XMLData)errorData).TextValue : "null"));
                }
```

```
                else
                {
                    // The COM call succeeded and the document is valid
                    listBoxMessages.Items.Add("Document " + XMLSpy.ActiveDocument.Name + "
is valid.");
                }
            }
            catch (Exception ex)
            {
                // The COM call was not successful
                // Probably no application instance has been started or no document is
open.
                listBoxMessages.Items.Add("Error validating active document: " +
ex.Message);
            }
        }
```

29.3.1.2.3.6 Events

The code snippet below (from the AutomateXMLSpy example[1701]) lists the code for two event handlers. The AutomateXMLSpy example[1701] (*see the file Form1.cs*) is located in the C# subfolder of the API Examples folder:

Windows 7, Windows 8, Windows 10, Windows 11	C:\Users\<usuario>\Documentos\Altova\XMLSpy\2023\Examples

You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022.

Code snippet

```
        // Event handler for OnDocumentOpened event
        private void handleOnDocumentOpened(XMLSpyLib.Document i_ipDocument)
        {
            MessageBox.Show("Document " + i_ipDocument.Name + " was opened!");
        }

        // Remember if the event handler is currently registered.
        private bool bEventHandlerIsRegistered = false;

        // Handler for button 'OnDocuemntOpened Event On/Off
        private void toggleOnDocumentOpenedEvent_Click(object sender, EventArgs e)
        {
            if (XMLSpy != null)
            {
                if (bEventHandlerIsRegistered)
                    XMLSpy.OnDocumentOpened -= new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
                else
```


© 2016-2022 Altova GmbH

```
                    XMLSpy.OnDocumentOpened += new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);

                bEventHandlerIsRegistered = !bEventHandlerIsRegistered;
            }
        }
```

29.3.1.2.4 Java

The Application API can be accessed from Java code. To allow accessing the XMLSpy automation server directly from Java code, the libraries listed below must reside in the `classpath`. They are installed in the folder: `JavaAPI` in the XMLSpy application folder.

- `AltovaAutomation.dll`: a JNI wrapper for Altova automation servers (`AltovaAutomation_x64.dll` in the case of 64-bit versions)
- `AltovaAutomation.jar`: Java classes to access Altova automation servers
- `XMLSpyAPI.jar`: Java classes that wrap the XMLSpy automation interface
- `XMLSpyAPI_JavaDoc.zip`: a `Javadoc` file containing help documentation for the Java API

Note: In order to use the Java API, the DLL and Jar files must be on the Java Classpath.

Example Java project

An example Java project is supplied with your product installation. You can test the Java project and modify and use it as you like. For more details of the example Java project, see the section, Example Java Project 1712.

Rules for mapping the Application API names to Java

The rules for mapping between the Application API and the Java wrapper are as follows:

- ***Classes and class names***
  For every interface of the XMLSpy automation interface a Java class exists with the name of the interface.

- ***Method names***
  Method names on the Java interface are the same as used on the COM interfaces but start with a small letter to conform to Java naming conventions. To access COM properties, Java methods that prefix the property name with `get` and `set` can be used. If a property does not support write-access, no setter method is available. Example: For the `Name` property of the `Document` interface, the Java methods `getName` and `setName` are available.

- ***Enumerations***
  For every enumeration defined in the automation interface, a Java enumeration is defined with the same name and values.

- ***Events and event handlers***
  For every interface in the automation interface that supports events, a Java interface with the same name plus '`Event`' is available. To simplify the overloading of single events, a Java class with default implementations for all events is provided. The name of this Java class is the name of the event interface plus '`DefaultHandler`'. For example:
  `Application`: Java class to access the application

`ApplicationEvents`: Events interface for the Application
`ApplicationEventsDefaultHandler`: Default handler for `ApplicationEvents`

Exceptions to mapping rules

There are some exceptions to the rules listed above. These are listed below:

Interface	Java name
`Document`, method `SetEncoding`	`setFileEncoding`
`AuthenticView`, method `Goto`	`gotoElement`
`AuthenticRange`, method `Goto`	`gotoElement`
`AuthenticRange`, method `Clone`	`cloneRange`

This section

This section explains how some basic XMLSpy functionality can be accessed from Java code. It is organized into the following sub-sections:

- Example Java Project 1712
- Application Startup and Shutdown 1716
- Simple Document Access 1717
- Iterations 1717
- Use of Out-Parameters 1717
- Event Handlers 1718

29.3.1.2.4.1 Example Java Project

The XMLSpy installation package contains an example Java project, located in the the `API\Java` subfolder of the `Examples` folder :

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Users\<usuario>\Documentos\Altova\XMLSpy\2023\Examples`

This folder contains Java examples for the XMLSpy API. You can test it directly from the command line using the batch file `BuildAndRun.bat`, or you can compile and run the example project from within Eclipse. See below for instructions on how to use these procedures.

File list

The Java examples folder contains all the files required to run the example project. These files are listed below. If you are using a 64-bit version of the application, some filenames contain `_x64` in the name. These filenames are indicated with `(_x64)`.

`AltovaAutomation(_x64).dll`	Java-COM bridge: DLL part
`AltovaAutomation.jar`	Java-COM bridge: Java library part
`XMLSpyAPI.jar`	Java classes of the XMLSpy API

© 2016-2022 Altova GmbH

`RunXMLSpy.java`	Java example source code
`BuildAndRun.bat`	Batch file to compile and run example code from the command line prompt. Expects folder where Java Virtual Machine resides as parameter.
`.classpath`	Eclipse project helper file
`.project`	Eclipse project file
`XMLSpyAPI_JavaDoc.zip`	`Javadoc` file containing help documentation for the Java API

What the example does

The example starts up XMLSpy and performs a few operations, including opening and closing documents. When done, XMLSpy stays open. You must close it manually.

- Start XML Spy 1716: Starts XMLSpy, which is registered as an automation server, or activates XMLSpy if it is already running.
- Open OrgChart.pxf 1717: Locates one of the example documents installed with XMLSpy and opens it.
- Iteration and Changing the View Mode 1717: Changes the view of all open documents to Text View. The code also shows how to iterate through open documents.
- Iteration, validation, output parameters 1717: Validates the active document and shows the result in a message box. The code shows how to use output parameters.
- Event Handling 1718: Shows how to handle XMLSpy events.
- Shut down XMLSpy 1716: Shuts down XMLSpy.

You can modify the example in any way you like and run it.

Running the example from the command line

To run the example from the command line, open a command prompt window, go to the Java folder of the API Examples folder (*see above for location*), and then type:

```
buildAndRun.bat "<Path-to-the-Java-bin-folder>"
```

The Java binary folder must be that of a JDK 1.5 or later installation on your computer. Press the **Return** key. The Java source in `RunXMLSpy.java` will be compiled and then executed.

Loading the example in Eclipse

Open Eclipse and use the **Import | Existing Projects into Workspace** command to add the Eclipse project file (`.project`) located in the Java folder of the API Examples folder (*see above for location*). The project `RunXMLSpy` will then appear in your Package Explorer or Navigator. Select the project and then the command **Run as | Java Application** to execute the example.

Note: You can select a class name or method of the Java API and press F1 to get help for that class or method.

Java source code listing

The Java source code in the example file `RunXMLSpy.java` is listed below with comments.

```
001 // Access general JAVA-COM bridge classes
```

```
002 import com.altova.automation.libs.*;
003
004 // Access XMLSpy Java-COM bridge
005 import com.altova.automation.XMLSpy.*;
006 import com.altova.automation.XMLSpy.Enums.SPYViewModes;
007
008 /**
009  * An example that starts XMLSpy COM server and performs view operations on it
010  * Feel free to extend
011  */
012 public class RunXMLSpy
013 {
014   public static void main(String[] args)
015   {
016     // An instance of the application.
017     Application xmlSpy = null;
018
019     // Instead of COM error handling, use Java exception mechanism
020     try
021       {
022       // Start XMLSpy as COM server
023       xmlSpy = new Application();
024
025       // COM servers start up invisible, so make it visible
026       xmlSpy.setVisible(true);
027
028       // Locate samples installed with the product
029       String strExamplesFolder =
030       System.getenv("USERPROFILE") + "\\My Documents\\Altova\\XMLSpy2012\\Examples\\";
031
032       // Open two example files
033       xmlSpy.getDocuments().openFile(strExamplesFolder + "OrgChart.pxf", false);
034       xmlSpy.getDocuments().openFile(strExamplesFolder + "ExpReport.xml", false);
035
036       // Iterate through open documents and set view mode to 'Text'.
037       for (Document doc:xmlSpy.getDocuments())
038       if ( doc.getCurrentViewMode() != SPYViewModes.spyViewText)
039           doc.switchViewMode(SPYViewModes.spyViewText);
040
041       // An alternative iteration mode is index-based
042       // COM indices are typically zero-based
043       Documents documents = xmlSpy.getDocuments();
044       for (int i = 1; i <= documents.getCount();
045           i++)
046           {
047             Document doc = documents.getItem(i);
048
049       // Validation is one of the few methods to have output parameters.
050       // The class JVariant is the correct type for parameters in these cases.
051       // To get values back mark them with the by-reference flag.
052       JVariant validationErrorText = new
053
054       JVariant.JStringVariant("");
055
056       validationErrorText.setByRefFlag();
057       JVariant validationErrorCount = new
```

 © 2016-2022 Altova GmbH

```
058
059       JVariant.JIntVariant(0);
060
061       validationErrorCount.setByRefFlag();
062       JVariant validationErrorXMLData = new
063
064       JVariant.JIDispatchVariant(0);
065
066       validationErrorXMLData.setByRefFlag();
067       if (!doc.isValid(validationErrorText, validationErrorCount,
validationErrorXMLData))
068           System.out.println("Document" + doc.getName() + " is not wellformed - " +
validationErrorText.getStringValue());
069         else
070           System.out.println("Document" + doc.getName() + " is wellformed.");
071       }
072
073       // The following lines attach to the document events using a default
implementation
074       // for the events and override one of its methods.
075       // If you want to override all document events it is better to derive your
listener class
076       // from DocumentEvents and implement all methods of this interface.
077       Document doc = xmlSpy.getActiveDocument();
078       doc.addListener(new
079
080 DocumentEventsDefaultHandler()
081       {
082         @Override
083         public boolean
084
085 onBeforeCloseDocument(Document i_ipDoc) throws AutomationException
086         {
087           System.out.println("Document
088
089 " + i_ipDoc.getName() + " requested closing.");
090
091           // Allow closing of document
092           return true;
093         }
094       });
095       doc.close(true);
096       doc = null;
097
098       System.out.println("Watch XMLSpy!");
099     }
100     catch (AutomationException e)
101     {
102       // e.printStackTrace();
103     }
104     finally
105     {
106       // Make sure that XMLSpy can shut down properly.
107       if (xmlSpy != null)
108           xmlSpy.dispose();
109
```

```
110       // Since the COM server was made visible and still is visible,
111       // it will keep running, and needs to be closed manually.
112       System.out.println("Now close XMLSpy!");
113     }
114   }
115 }
```

29.3.1.2.4.2 Application Startup and Shutdown

The code listings below show how the application can be started up and shut down.

Application startup

Before starting up the application, the appropriate classes must be imported (*see below*).

```
01 // Access general JAVA-COM bridge classes
02 import com.altova.automation.libs.*;
03
04 // Access XMLSpy Java-COM bridge
05 import com.altova.automation.XMLSpy.*;
06 import com.altova.automation.XMLSpy.Enums.SPYViewModes;
07
08 /**
09  * An example that starts XMLSpy COM server and performs view operations on it
10  * Feel free to extend
11  */
12 public class RunXMLSpy
13 {
14   public static void main(String[] args)
15   {
16     // An instance of the application.
17     Application xmlSpy = null;
18
19     // Instead of COM error handling, use Java exception mechanism
20     try
21     {
22         // Start XMLSpy as COM server
23           xmlSpy = new Application();
24         // COM servers start up invisible, so make it visible
25           xmlSpy.setVisible(true);
26
27 ...
28     }
29   }
30 }
```

Application shutdown

The application can be shut down as shown below.

```
01 {
02       // Allow shutdown of XMLSpy by releasing the UI.
03       xmlSpy.setVisible(true);
```


© 2016-2022 Altova GmbH

```
04
05       // Make sure that XMLSpy can shut down properly.
06       if (xmlSpy != null)
07         xmlSpy.dispose();
08
09       // Since the COM server was made visible and still is visible,
10       // it will keep running, and needs to be closed manually.
11       System.out.println("Now close XMLSpy!");
12 }
```

29.3.1.2.4.3 Simple Document Access

The code listing below shows how to open a document.

```
1  // Locate samples installed with the product
2  String strExamplesFolder =
3  System.getenv("USERPROFILE") + "\\My Documents\\Altova\\XMLSpy2012\\Examples\\";
4
5
6  // Open file
7  xmlSpy.getDocuments().openFile(strExamplesFolder + "OrgChart.pxf", false);
```

29.3.1.2.4.4 Iterations

The listing below shows how to iterate through open documents.

```
01  // Iterate through open documents and set view mode to 'Text'.
02  for (Document doc:xmlSpy.getDocuments())
03    if ( doc.getCurrentViewMode() != SPYViewModes.spyViewText)
04            doc.switchViewMode(SPYViewModes.spyViewText);
05
06  // An alternative iteration mode is index-based
07  // COM indices are typically zero-based
08  Documents documents = xmlSpy.getDocuments();
09     for (int i = 1; i <= documents.getCount();
10         i++)
11       {
12         Document doc = documents.getItem(i);
13               ...
14       }
```

29.3.1.2.4.5 Use of Out-Parameters

The code listing below iterates through open documents and validates each of them. For each validation, a message is generated using the output parameters of the Validation method.

```
01 // An alternative iteration mode is index-based
02 // COM indices are typically zero-based
03 Documents documents = xmlSpy.getDocuments();
04 for (int i = 1; i <= documents.getCount();
05       i++)
06     {
07         Document doc = documents.getItem(i);
08
09 // Validation is one of the few methods to have output parameters.
10 // The class JVariant is the correct type for parameters in these cases.
11 // To get values back mark them with the by-reference flag.
12 JVariant validationErrorText = new
13
14 JVariant.JStringVariant("");
15
16 validationErrorText.setByRefFlag();
17       JVariant validationErrorCount = new
18
19 JVariant.JIntVariant(0);
20
21 validationErrorCount.setByRefFlag();
22       JVariant validationErrorXMLData = new
23
24 JVariant.JIDispatchVariant(0);
25
26 validationErrorXMLData.setByRefFlag();
27       if (!doc.isValid(validationErrorText,
28
29         validationErrorCount, validationErrorXMLData))
30         System.out.println("Document
31
32         " + doc.getName() + " is not wellformed - " +
33
34         validationErrorText.getStringValue());
35       else
36         System.out.println("Document
37
38         " + doc.getName() + " is wellformed.");
39     }
```

29.3.1.2.4.6 Event Handlers

The listing below shows how to listen for and use events.

```
01 // The following lines attach to the document events using a default implementation
02 // for the events and override one of its methods.
03 // If you want to override all document events it is better to derive your listener
class
04 // from DocumentEvents and implement all methods of this interface.
05
```

 © 2016-2022 Altova GmbH

```
06 Document doc = xmlSpy.getActiveDocument();
07 doc.addListener(new DocumentEventsDefaultHandler()
08     {
09       @Override
10       public boolean
11       onBeforeCloseDocument(Document i_ipDoc) throws AutomationException
12       {
13         System.out.println("Document " + i_ipDoc.getName() + " requested closing.");
14
15         // Allow closing of document
16         return true;
17       }
18     });
19 doc.close(true);
20 doc = null;
```

29.3.1.3 The DOM and XMLData

The `XMLData` interface gives you full access to the XML structure behind the current document with less methods than DOM and is much simpler. The `XMLData` interface is a minimalist approach to reading and modifying existing, or newly created XML data. You might however, want to use a DOM tree because you can access one from an external source or you just prefer the MSXML DOM implementation.

The `ProcessDOMNode()` and `ProcessXMLDataNode()` functions provided below convert any segments of an XML structure between XMLData and DOM.

To use the `ProcessDOMNode()` function:

- pass the root element of the DOM segment you want to convert in objNode and
- pass the plugin object with the CreateChild() method in objCreator

To use the `ProcessXMLDataNode()` function:

- pass the root element of the XMLData segment in objXMLData and
- pass the DOMDocument object created with MSXML in xmlDoc

```
////////////////////////////////////////////////////////////////
// DOM To XMLData conversion
Function ProcessDOMNode(objNode,objCreator)
{
	var objRoot;
	objRoot = CreateXMLDataFromDOMNode(objNode,objCreator);

	If(objRoot)   {
		If((objNode.nodeValue != Null) && (objNode.nodeValue.length > 0))
			objRoot.TextValue = objNode.nodeValue;
			// add attributes
		If(objNode.attributes) {
			var Attribute;
			var oNodeList = objNode.attributes;

			For(var i = 0;i < oNodeList.length; i++) {
				Attribute = oNodeList.item(i);
```

```
                        var newNode;
                        newNode = ProcessDOMNode(Attribute,objCreator);

                        objRoot.AppendChild(newNode);
                }
        }
        If(objNode.hasChildNodes)   {
                try {
                        // add children
                        var Item;
                        oNodeList = objNode.childNodes;

                        For(var i = 0;i < oNodeList.length; i++) {
                                Item = oNodeList.item(i);

                                var newNode;
                                newNode = ProcessDOMNode(Item,objCreator);

                                objRoot.AppendChild(newNode);
                        }
                }
                catch(err) {
                }
        }
    }
    Return objRoot;
}

Function CreateXMLDataFromDOMNode(objNode,objCreator)
{
    var bSetName = True;
    var bSetValue = True;

    var nKind = 4;

    switch(objNode.nodeType)    {
            Case 2:nKind = 5;break;
            Case 3:nKind = 6;bSetName = False;break;
            Case 4:nKind = 7;bSetName = False;break;
            Case 8:nKind = 8;bSetName = False;break;
            Case 7:nKind = 9;break;
    }
    var objNew = Null;
    objNew = objCreator.CreateChild(nKind);

    If(bSetName)
            objNew.Name = objNode.nodeName;

    If(bSetValue && (objNode.nodeValue != Null))
            objNew.TextValue = objNode.nodeValue;

    Return objNew;
}
////////////////////////////////////////////////////////////
// XMLData To DOM conversion
```

 © 2016-2022 Altova GmbH

```
Function ProcessXMLDataNode(objXMLData,xmlDoc)
{
	var objRoot;
	objRoot = CreateDOMNodeFromXMLData(objXMLData,xmlDoc);

	If(objRoot)	{
		If(IsTextNodeEnabled(objRoot) && (objXMLData.TextValue.length > 0))
			objRoot.appendChild(xmlDoc.createTextNode(objXMLData.TextValue));

		If(objXMLData.HasChildren)	{
			try {
				var objChild;
				objChild = objXMLData.GetFirstChild(-1);

				While(True)	{
					If(objChild)	{
						var newNode;
						newNode = ProcessXMLDataNode(objChild,xmlDoc);

						If(newNode.nodeType == 2)	{
							// child node is an attribute
							objRoot.attributes.setNamedItem(newNode);
						}
						Else
							objRoot.appendChild(newNode);
					}
					objChild = objXMLData.GetNextChild();
				}
			}
			catch(err) {
			}
		}
	}
	Return objRoot;
}

Function CreateDOMNodeFromXMLData(objXMLData,xmlDoc)
{
	switch(objXMLData.Kind)	{
		Case 4:Return xmlDoc.createElement(objXMLData.Name);
		Case 5:Return xmlDoc.createAttribute(objXMLData.Name);
		Case 6:Return xmlDoc.createTextNode(objXMLData.TextValue);
		Case 7:Return xmlDoc.createCDATASection(objXMLData.TextValue);
		Case 8:Return xmlDoc.createComment(objXMLData.TextValue);
		Case 9:Return
xmlDoc.createProcessingInstruction(objXMLData.Name,objXMLData.TextValue);
	}

	Return xmlDoc.createElement(objXMLData.Name);
}
Function IsTextNodeEnabled(objNode)
{
	switch(objNode.nodeType) {
		Case 1:
		Case 2:
```

```
            Case 5:
            Case 6:
            Case 11:Return True;
        }
        Return False;
}
```

29.3.1.4 Obsolete: vista Authentic Row operations

If the schema on which an XML document is based specifies that an element is repeatable, such a structure can be represented in vista Authentic as a table. When represented as a table, rows and their contents can be manipulated individually, thereby allowing you to manipulate each of the repeatable elements individually. Such row operations would be performed by an external script.

If an external script is to perform row operations then two steps must occur:

- The first step checks whether the cursor is currently in a row using a property. Such a check could be, for example, `IsRowInsertEnabled`, which returns a value of either `TRUE` or `FALSE`.
- If the return value is `TRUE` then a row method, such as `RowAppend`, can be called. (`RowAppend` has no parameters and returns no value.)

The following is a list of properties and methods available for table operations. Each property returns a `BOOL`, and the methods have no parameter.

Property	Method	Table operations
IsRowInsertEnabled	**RowInsert**[1984], superseded by `AuthenticRange.InsertRow`[1766]	Insert row operation
IsRowAppendEnabled	**RowAppend**[1982], superseded by `AuthenticRange.AppendRow`[1752]	Append row operation
IsRowDeleteEnabled	**RowDelete**[1983], superseded by `AuthenticRange.DeleteRow`[1755]	Delete row operation
IsRowMoveUpEnabled	**RowMoveUp**[1985], superseded by `AuthenticRange.MoveRowUp`[1773]	Move XML data up one row
IsRowMoveDownEnabled	**RowMoveDown**[1984], superseded by `AuthenticRange.MoveRowDown`[1772]	Move XML data down one row
IsRowDuplicateEnabled	**RowDuplicate**[1983], superseded by `AuthenticRange.DuplicateRow`[1756]	Duplicate currently selected row

29.3.1.5 Obsolete: vista Authentic Editing operations

When XML data is displayed as data in vista Authentic, it is possible to manipulate individual elements using standard editing operations such as cut, copy, and paste. However, not all XML data nodes can be edited. So, in order to carry out an editing operation, first a property is used to test whether editing is possible, and then a method is called to perform the editing operation.

© 2016-2022 Altova GmbH

The only method that does not have a test is the method `EditSelectAll`, which automatically selects all elements displayed in the document.

The following is a list of properties and methods that perform editing operations. Each property returns a `BOOL`, and the methods have no parameter.

Property	Method	Editing operation
IsEditUndoEnabled	**EditUndo**[1970], superseded by `AuthenticView.Undo`[1796]	Undo an editing operation
IsEditRedoEnabled	**EditRedo**[1970], superseded by `AuthenticView.Redo`[1795]	Redo an editing operation
IsEditCopyEnabled	**EditCopy**[1968], superseded by `AuthenticRange.Copy`[1754]	Copy selected text to the clipboard
IsEditCutEnabled	**EditCut**[1969], superseded by `AuthenticRange.Cut`[1755]	Cut selected text to the clipboard
IsEditPasteEnabled	**EditPaste**[1969], superseded by `AuthenticRange.Paste`[1773]	Paste from clipboard to current cursor position
IsEditClearEnabled	**EditClear**[1968], superseded by `AuthenticRange.Delete`[1755]	Clear selected text from XML document

29.3.2 Interfaces

Object Hierarchy

Application[1724]
- SpyProject[1891]
  - SpyProjectItems[1895]
    - SpyProjectItem[1893]
- Documents[1843]
  - Document[1813]
    - GridView[1875]
    - AuthenticView[1778]
      - AuthenticRange[1750]
      - AuthenticDataTransfer[1745] (previously DocEditDataTransfer)
    - OldAuthenticView[1965] (previously DocEditView, **now obsolete**, superseded by AuthenticView[1778] and AuthenticRange[1750])
      - AuthenticSelection[1963] (previously DocEditSelection, **now obsolete**, superseded by AuthenticRange[1750])
      - AuthenticEvent[1950] (previously DocEditEvent, **now obsolete**)
      - AuthenticDataTransfer[1745] (previously DocEditDataTransfer)
    - TextView[1900]
    - XMLData[1939]
- Dialogs[1730]
  - CodeGeneratorDlg[1797]
  - FileSelectionDlg[1856]
  - SchemaDocumentationDlg[1878]

GenerateSampleXMLDlg 1869
DTDSchemaGeneratorDlg 1847
FindInFilesDlg 1857
WSDLDocumentationDlg 1905
WSDL20DocumentationDlg 1917
XBRLDocumentationDlg 1928
DatabaseConnection 1803
ExportSettings 1853
TextImportExportSettings 1897
ElementList 1851
ElementListItem 1852

Enumerations 1987

Description
This chapter contains the reference of the XMLSpy 1.5 Type Library.

Most of the given examples are written in VisualBasic. These code snippets assume that there is a variable defined and set, called **objSpy of type Application**. There are also some code samples written in JavaScript.

29.3.2.1 Application

Methods

GetDatabaseImportElementList 1731
GetDatabaseSettings 1731
GetDatabaseTables 1732
ImportFromDatabase 1735
CreateXMLSchemaFromDBStructure 1729

GetTextImportElementList 1733
GetTextImportExportSettings 1733
ImportFromText 1736

ImportFromWord 1737

ImportFromSchema 1736

GetExportSettings 1732

NewProject 1738
OpenProject 1738

AddMacroMenuItem 1728
ClearMacroMenu 1729

ShowForm 1742

ShowApplication 1741

URLDelete 1742

 © 2016-2022 Altova GmbH

URLMakeDirectory 1742

AddXSLT_XQParameter 1728
GetXSLT_XQParameterCount 1734
GetXSLT_XQParameterName 1734
GetXSLT_XQParameterXPath 1734
RemoveXSLT_XQParameter 1740

FindInFiles 1730

Quit 1739

Properties
Application 1728
Parent 1739

ActiveDocument 1728
Documents 1730

CurrentProject 1729

Dialogs 1730

WarningNumber 1743
WarningText 1743

Status 1742
MajorVersion 1738
MinorVersion 1738
Edition 1730
IsAPISupported 1737
ServicePackVersion 1741

Description
Application is the root for all other objects. It is the only object you can create by CreateObject (VisualBasic) or other similar COM related functions.

Example

```
Dim objSpy As Application
Set objSpy = CreateObject("XMLSpy.Application")
```

29.3.2.1.1 Events

29.3.2.1.1.1 OnBeforeOpenDocument

Event: OnBeforeOpenDocument(*objDialog* as FileSelectionDlg 1856)

Description

This event gets fired whenever a document gets opened via the OpenFile or OpenURL menu command. It is sent after a document file has been selected but before the document gets opened. The file selection dialog object is initialized with the name of the selected document file. You can modify this selection. To continue the opening of the document leave the FileSelectionDlg.DialogAction[1857] property of *io_objDialog* at its default value spyDialogOK[1990]. To abort the opening of the document set this property to spyDialogCancel[1990].

Examples
Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:
```
Function On_BeforeOpenDocument(objDialog)
End Function
```

XMLSpy scripting environment - JScript:
```
function On_BeforeOpenDocument(objDialog)
{
}
```

XMLSpy IDE Plugin:
```
IXMLSpyPlugIn.OnEvent (26, ...) // nEventId = 26
```

29.3.2.1.1.2 OnBeforeOpenProject

Event: OnBeforeOpenProject(*objDialog* as FileSelectionDlg[1856])

Description
This event gets fired after a project file has been selected but before the project gets opened. The file selection dialog object is initialized with the name of the selected project file. You can modify this selection. To continue the opening of the project leave the FileSelectionDlg.DialogAction[1857] property of *io_objDialog* at its default value spyDialogOK[1990]. To abort the opening of the project set this property to spyDialogCancel[1990].

Examples
Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:
```
Function On_BeforeOpenProject(objDialog)
End Function
```

XMLSpy scripting environment - JScript:
```
function On_BeforeOpenProject(objDialog)
{
}
```

XMLSpy IDE Plugin:
```
IXMLSpyPlugIn.OnEvent (25, ...) // nEventId = 25
```


© 2016-2022 Altova GmbH

29.3.2.1.1.3 OnDocumentOpened

Event: OnDocumentOpened(*objDocument* as Document[1813])

Description
This event gets fired whenever a document opens in XMLSpy. This can happen due to opening a file with the OpenFile or OpenURL dialog, creating a new file or dropping a file onto XMLSpy. The new document gets passed as parameter. The operation cannot be canceled.

Examples
Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:
```
Function On_OpenDocument(objDocument)
End Function
```

XMLSpy scripting environment - JScript:
```
function On_OpenDocument(objDocument)
{
}
```

XMLSpy IDE Plugin:
```
IXMLSpyPlugIn.OnEvent (7, ...)   // nEventId = 7
```

29.3.2.1.1.4 OnProjectOpened

Event: OnProjectOpened(*objProject* as SpyProject[1891])

Description
This event gets fired whenever a project gets opened in XMLSpy. The new project gets passed as parameter.

Examples
Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:
```
Function On_OpenProject(objProject)
End Function
```

XMLSpy scripting environment - JScript:
```
function On_OpenProject(objProject)
{
}
```

XMLSpy IDE Plugin:
```
IXMLSpyPlugIn.OnEvent (6, ...)   // nEventId = 6
```

29.3.2.1.2 ActiveDocument

Property: ActiveDocument as Document[1813]

Description
Reference to the active document. If no document is open, ActiveDocument is null (nothing).

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.3 AddMacroMenuItem

Method: AddMacroMenuItem(*strMacro* as String,*strDisplayText* as String)

Description
Adds a menu item to the **Tools** menu. This new menu item invokes the macro defined by strMacro. See also Example Scripting Project[1654].

Errors
1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
1108 Number of macro items is limited to 16 items.

29.3.2.1.4 AddXSLT_XQParameter

Method: AddXSLT_XQParameter(name as String, XPath as String)

Description
Adds an XSLT or XQuery parameter. The parameter's name and value are the two arguments of the method.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.
1124 The XPath expression is not set.
1125 Not a QName.
1126 The specified XPath is not valid. Reason for invalidity appended.
1127 A parameter with the submitted name already exists.

29.3.2.1.5 Application

Property: Application as Application[1724] (read-only)

Description

© 2016-2022 Altova GmbH

Accesses the XMLSpy application object.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.6 ClearMacroMenu

Method: ClearMacroMenu()

Return Value
None

Description
Removes from the **Tools** menu those menu items that were added by calling `AddMacroMenuItem`[1728]. See also Example Scripting Project[1654].

Errors
1111 The application object is no longer valid.

29.3.2.1.7 CreateXMLSchemaFromDBStructure

Method: CreateXMLSchemaFromDBStructure(pImportSettings as DatabaseConnection[1803], pTables as ElementList[1851])

Description
CreateXMLSchemaFromDBStructure creates from a database specified in pImportSettings for the defined tables in pTables new XML Schema document(s) describing the database tables structure.

The parameter pTables specifies which table structures the XML Schema document should contain. This parameter can be NULL, specifying that all table structures will be exported.

See also GetDataBaseTables[1732].

Errors
1112 Invalid database specified.
1120 Database import failed.

29.3.2.1.8 CurrentProject

Property: CurrentProject as SpyProject[1891]

Description
Reference to the active document. If no project is open, CurrentProject is null (nothing).

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.9 Dialogs

Property: Dialogs as Dialogs[1809] (read-only)

Description
Access the built-in dialogs of XMLSpy.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.10 Documents

Property: Documents as Documents[1843]

Description
Collection of all open documents.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.11 Edition

Property: Edition as String

Description
Returns the edition of the application, for example `Altova XMLSpy Enterprise Edition` for the Enterprise edition.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.12 FindInFiles

Method: FindInFiles(*pSettings* as FindInFilesDlg[1857]) as FindInFilesResults[1868]

Description
Returns a FindInFilesResults[1868] object containing information about the files that matched the specified settings.

© 2016-2022 Altova GmbH

Errors

1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.13 GetDatabaseImportElementList

Method: GetDatabaseImportElementList(*pImportSettings* as DatabaseConnection[1803]) as ElementList[1851]

Description

The function returns a collection of ElementListItems where the properties ElementListItem.Name[1853] contain the names of the fields that can be selected for import and the properties ElementListItem.ElementKind[1853] are initialized either to *spyXMLDataAttr* or *spyXMLDataElement*, depending on the value passed in DatabaseConnection.AsAttributes[1804]. This list serves as a filter to what finally gets imported by a future call to ImportFromDatabase[1735]. Use ElementList.RemoveElement[1852] to exclude fields from import.

Properties mandatory to be filled out for the database connection are one of DatabaseConnection.File[1806], DatabaseConnection.ADOConnection[1804] and DatabaseConnection.ODBCConnection[1808], as well as DatabaseConnection.SQLSelect[1809]. Use the property DatabaseConnection.AsAttributes[1804] to initialize ElementListItem.ElementKind[1853] of the resulting element list to either *spyXMLDataAttr* or *spyXMLDataElement*, respectively.

Example

See example at ImportFromDatabase[1735].

Errors

1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
1107 Import from database failed.
1112 Invalid database specified.
1114 Select statement is missing.
1119 database element list import failed.

29.3.2.1.14 GetDatabaseSettings

Method: GetDatabaseSettings() as DatabaseConnection[1803]

Description

GetDatabaseSettings creates a new object of database settings. The object is used to specify database connection parameters for the methods GetDatabaseTables[1732], GetDatabaseImportElementList[1731], ImportFromDatabase[1735], ImportFromSchema[1736] and ExportToDatabase[1825].

Example

See example of ImportFromDatabase[1735].

Errors

1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.15 GetDatabaseTables

Method: GetDatabaseTables(*pImportSettings* as DatabaseConnection[1724]) as ElementList[1851]

Description
GetDatabaseTables reads the table names from the database specified in *pImportSettings*. Properties mandatory to be filled out for the database connection are one of DatabaseConnection.File[1806], DatabaseConnection.ADOConnection[1804] and DatabaseConnection.ODBCConnection[1808]. All other properties are ignored.
The function returns a collection of ElementListItems where the properties ElementListItem.Name[1853] contain the names of tables stored in the specified database. The remaining properties of ElementListItem[1852] are unused.

Errors
1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
1112 Invalid database specified.
1113 Error while reading database table information.
1118 Database table query failed.

Example

```
        Dim objImpSettings As DatabaseConnection
Set objImpSettings = objSpy.GetDatabaseSettings
objImpSettings.ADOConnection = TxtADO.Text

'store table names in list box
ListTables.Clear

Dim objList As ElementList
Dim objItem As ElementListItem
On Error GoTo ErrorHandler
Set objList = objSpy.GetDatabaseTables(objImpSettings)

        For Each objItem In objList
                ListTables.AddItem objItem.Name
        Next
```

29.3.2.1.16 GetExportSettings

Method: GetExportSettings()as ExportSettings[1853] (read-only)

Description
GetExportSettings creates a new object of common export settings. This object is used to pass the parameters to the export functions and defines the behaviour of the export calls. See also the export functions from Document[1813] and the examples at Import and Export[1694].

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.1.17 GetTextImportElementList

Method: GetTextImportElementList(pImportSettings as TextImportExportSettings[1897]) as ElementList[1851]

Description
GetTextImportElementList retrieves importing information about the text-file as specified in pImportSettings. The function returns a collection of ElementListItems where the properties ElementListItem.Name[1853] contain the names of the fields found in the file. The values of remaining properties are undefined.

If the text-file does not contain a column header, set `pImportSettings.HeaderRow`[1899] to `false`. The resulting element list will contain general column names like 'Field1' and so on.

See also Import and export of data[1694].

Errors

1111	The application object is no longer valid.
1100	Invalid parameter or invalid address for the return parameter was specified.
1107	Import from database failed.
1115	Error during text element list import. Cannot create parser for import file.
1116	Error during text element list import.

Example

```
' ------------------------------------------------------
' VBA client code fragment - import selected fields from text file
' ------------------------------------------------------
    Dim objImpSettings As TextImportExportSettings
    Set objImpSettings = objSpy.GetTextImportExportSettings

    objImpSettings.ImportFile = "C:\ImportMe.txt"
    objImpSettings.HeaderRow = False

    Dim objList As ElementList
    Set objList = objSpy.GetTextImportElementList(objImpSettings)

    'exclude first column
    objList.RemoveItem 1

    Dim objImpDoc As Document
    On Error Resume Next
    Set objImpDoc = objSpy.ImportFromText(objImpSettings, objList)
    CheckForError
```

29.3.2.1.18 GetTextImportExportSettings

Method: GetTextImportExportSettings() as TextImportExportSettings[1897] (read-only)

Description
GetTextImportExportSettings creates a new object of common import and export settings for text files. See also the example for Application.GetTextImportElementList[1733] and Import and Export[1694].

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.19 GetXSLT_XQParameterCount

Method: GetXSLT_XQParameterCount() as Long

Description
Returns the number of XSLT and XQuery parameters.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.20 GetXSLT_XQParameterName

Method: GetXSLT_XQParameterName(index as Long) as String

Description
Returns the name of the XSLT or XQuery parameter identified by the supplied index.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.21 GetXSLT_XQParameterXPath

Method: GetXSLT_XQParameterXPath(index as Long) as String

Description
Returns the XPath expression of the XSLT or XQuery parameter identified by the supplied index.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.1.22 ImportFromDatabase

Method: ImportFromDatabase(pImportSettings as DatabaseConnection[1803],pElementList as ElementList[1851]) as Document[1813]

Return Value
Creates a new document containing the data imported from the database.

Description
ImportFromDatabase imports data from a database as specified in pImportSettings and creates a new document containing the data imported from the database. Properties mandatory to be filled out are one of DatabaseConnection.File[1806], DatabaseConnection.ADOConnection[1804] or DatabaseConnection.ODBCConnection[1808] and DatabaseConnection.SQLSelect[1809]. Additionally, you can use DatabaseConnection.AsAttributes[1804], DatabaseConnection.ExcludeKeys[1806], DatabaseConnection.IncludeEmptyElements[1807] and NumberDateTimeFormat[1807] to further parameterize import.

The parameter pElementList specifies which fields of the selected data gets written into the newly created document, and which are created as elements and which as attributes. This parameter can be NULL, specifying that all selected fields will be imported as XML elements.

See GetDatabaseSettings[1731] and GetDatabaseImportElementList[1731] for necessary steps preceding any import of data from a database.

Errors

1111	The application object is no longer valid.
1100	Invalid parameter or invalid address for the return parameter was specified.
1107	Import from database failed.
1112	Invalid database specified.
1114	Select statement is missing.
1117	Transformation to XML failed.
1120	Database import failed.

Example

```
        Dim objImpSettings As DatabaseConnection
        Set objImpSettings = objSpy.GetDatabaseSettings

        objImpSettings.ADOConnection = strADOConnection
        objImpSettings.SQLSelect = "SELECT * FROM MyTable"

        Dim objDoc As Document
        On Error Resume Next
        Set objDoc = objSpy.ImportFromDatabase(objImpSettings,
objSpy.GetDatabaseImportElementList(objImpSettings))
        ' CheckForError here
```

29.3.2.1.23 ImportFromSchema

Method: ImportFromSchema(*pImportSettings* as DatabaseConnection[1803],*strTable* as String,*pSchemaDoc* as Document[1813]) as Document[1813]

Return Value
Creates a new document filled with data from the specified database as specified by the schema definition in *pSchemaDoc*.

Description
ImportFromSchema imports data from a database specified in pImportSettings. Properties mandatory to be filled out are one of DatabaseConnection.File[1806], DatabaseConnection.ADOConnection[1804] or DatabaseConnection.ODBCConnection[1808]. Additionally, you can use DatabaseConnection.AsAttributes[1804], DatabaseConnection.ExcludeKeys[1806] and NumberDateTimeFormat[1807] to further parameterize import. All other properties get ignored.

ImportFromSchema does not use an explicit SQL statement to select the data. Instead, it expects a structure definition of the document to create in form of an XML schema document in *pSchemaDoc*. From this definition the database select statement is automatically deduced. Specify in *strTable* the table name of the import root that will become the root node in the new document.

See GetDatabaseSettings[1731] and GetDatabaseTables[1732] for necessary steps preceding an import from a database based on a schema definition. To create the schema definition file use command 'create database schema' from the 'convert' menu of XMLSpy.

Errors

1111	The application object is no longer valid.
1100	Invalid parameter or invalid address for the return parameter was specified.
1107	Import from database failed.
1112	Invalid database specified.
1120	Database import failed.
1121	Could not create validator for the specified schema.
1122	Failed parsing schema for database import.

29.3.2.1.24 ImportFromText

Method: ImportFromText(*pImportSettings* as TextImportExportSettings[1897],*pElementList* as ElementList[1851]) as Document[1813]

Description
ImportFromText imports the text file as specified in pImportSettings. The parameter pElementList can be used as import filter. Either pass the list returned by a previous call to GetTextImportElementList[1733] or null to import all columns. To avoid import of unnecessary columns use ElementList.RemoveElement[1852] to remove the corresponding field names from `pElementList` before calling `ImportFromText`.
The method returns the newly created document containing the imported data. This document is the same as the active document of XMLSpy.

See also Import and export of data[1694].

Errors

© 2016-2022 Altova GmbH

1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
1107 Import from text file failed.
1117 Transformation to XML failed.

Example

```
' ------------------------------------------------------
' VBA client code fragment - import from text file
' ------------------------------------------------------
        Dim objImpSettings As TextImportExportSettings
        Set objImpSettings = objSpy.GetTextImportExportSettings

        objImpSettings.ImportFile = strFileName
        objImpSettings.HeaderRow = False

        Dim objImpDoc As Document
        On Error Resume Next
        Set objImpDoc = objSpy.ImportFromText(objImpSettings,
objSpy.GetTextImportElementList(objImpSettings))

        CheckForError
```

29.3.2.1.25 ImportFromWord

Method: ImportFromWord(*strFile* as String) as Document[1813]

Description
ImportFromWord imports the MS-Word Document strFile into a new XML document.

Errors
1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
Import from document failed.

29.3.2.1.26 IsAPISupported

Property: IsAPISupported as Boolean

Description
Returns whether the API is supported in this version or not.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.27 MajorVersion

Property: MajorVersion as Integer

Description
Returns the application version's major number, for example `15` for 2013 versions, and `16` for 2014 versions..

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.28 MinorVersion

Property: MinorVersion as Integer

Description
Returns the application version's minor number.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.29 NewProject

Method: NewProject(*strPath* as String,*bDiscardCurrent* as Boolean)

Description
NewProject creates a new project.

If there is already a project open that has been modified and bDiscardCurrent is false, then NewProject() fails.

Errors
1111 The application object is no longer valid.
1102 A project is already open but *bDiscardCurrent* is *true*.
1103 Creation of new project failed.

29.3.2.1.30 OpenProject

Method: OpenProject(*strPath* as String,*bDiscardCurrent* as Boolean,*bDialog* as Boolean)

Parameters
strPath
Path and file name of the project to open. Can be empty if bDialog is true.

bDiscardCurrent

 © 2016-2022 Altova GmbH

Discard currently open project and possibly lose changes.

bDialog
Show dialogs for user input.

Return Value
None

Description
OpenProject opens an existing project. If there is already a project open that has been modified and bDiscardCurrent is false, then OpenProject() fails.

Errors
1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
1101 Cannot open specified project.
1102 A project is already open but *bDiscardCurrent* is *true*.

29.3.2.1.31 Parent

Property: Parent as Application[1724] (read-only)

Description
Accesses the XMLSpy application object.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.32 Quit

Method: Quit()

Return Value
None

Description
This method terminates XMLSpy. All modified documents will be closed without saving the changes. This is also true for an open project.

If XMLSpy was automatically started as an automation server by a client program, the application will not shut down automatically when your client program shuts down if a project or any document is still open. Use the Quit method to ensure automatic shut-down.

Errors
1111 The application object is no longer valid.

29.3.2.1.33 ReloadSettings

Method: ReloadSettings

Return Value

Description
The application settings are reloaded from the registry.

Available with TypeLibrary version 1.5

Errors
1111 The application object is no longer valid.

29.3.2.1.34 RemoveXSLT_XQParameter

Method: RemoveXSLT_XQParameter(index as Long)

Description
Removes the XSLT or XQuery parameter identified by the supplied index.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.35 RunMacro

Method: RunMacro(*strMacro* as String)

Return Value

Description
Calls the specified macro either from the project scripts (if present) or from the global scripts.

Available with TypeLibrary version 1.5

Errors
1111 The application object is no longer valid.

29.3.2.1.36 ScriptingEnvironment

Property: ScriptingEnvironment as IUnknown (read-only)

Description

© 2016-2022 Altova GmbH

Reference to any active scripting environment. This property makes it possible to access the TypeLibrary of the XMLSpyFormEditor.exe application which is used as the current scripting environment.

Available with TypeLibrary version 1.5

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.37 ServicePackVersion

Property: ServicePackVersion as Long

Description
Returns the Service Pack version number of the application. Eg: 1 for 2010 R2 SP1

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.38 ShowApplication

Method: ShowApplication(*bShow* as Boolean)

Return Value
None

Description
The method shows (bShow = True) or hides (bShow = False) XMLSpy.

Errors
1110 The application object is no longer valid.

29.3.2.1.39 ShowFindInFiles

Method: ShowFindInFiles(pSettings as FindInFilesDlg[1857]) as Boolean

Return Value
Returns false if the user pressed the Cancel button, true otherwise.

Description
Displays the FindInFiles dialog preset with the given settings. The user modifications of the settings are stored in the passed dialog object.

Errors
1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.

29.3.2.1.40 ShowForm

Method: ShowForm(*strFormName* as String) as Long

Return Value
Returns zero if the user pressed a Cancel button or the form calls TheView.Cancel().

Description
Displays the form strFormName.

Forms, event handlers and macros can be created with the Scripting Environment. Select "Switch to scripting environment" from the **Tools** menu to invoke the Scripting Environment.

Errors
1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.

29.3.2.1.41 Status

Property: Status as ENUMApplicationStatus 1987

Description
Returns the current status of the running application.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.42 URLDelete

Method: URLDelete(*strURL* as String,*strUser* as String,*strPassword* as String)

Return Value
None

Description
The method deletes the file at the URL strURL.

Errors
1111 The application object is no longer valid.
1109 Error deleting file at specified URL.

29.3.2.1.43 URLMakeDirectory

Method: URLMakeDirectory(*strURL* as String,*strUser* as String,*strPassword* as String)

Return Value

© 2016-2022 Altova GmbH

None

Description
The method creates a new directory at the URL strURL.

Errors
1111 The application object is no longer valid.
1100 Invalid parameter specified.

29.3.2.1.44 Visible

Property: Visible as VARIANT_BOOL

Description
Sets or gets the visibility attribute of XMLSpy. This standard automation property makes usage of ShowApplication[1741] obsolete.

Errors
1110 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.45 WarningNumber

Property: WarningNumber as integer

Description
Some methods fill the property WarningNumber with additional information if an error occurs.

Currently just Documents.OpenFile[1846] fills this property.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.46 WarningText

Property: WarningText as String

Description
Some methods fill the property WarningText with additional information if an error occurs.

Currently just Documents.OpenFile[1846] fills this property.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.2 AuthenticContextMenu

The context menu interface provides the means for the user to customize the context menus shown in Authentic. The interface has the methods listed in this section.

29.3.2.2.1 CountItems

Method: CountItems `()` nItems as long

Return Value
Returns the number of menu items.

Errors
2501 Invalid object.

29.3.2.2.2 DeleteItem

Method: DeleteItem `(IndexPosition as long)`

Return Value
Deletes the menu item that has the index position submitted in the first parameter.

Errors
2501 Invalid object
2502 Invalid index

29.3.2.2.3 GetItemText

Method: GetItemText `(IndexPosition as long)` MenuItemName as string

Return Value
Gets the name of the menu item located at the index position submitted in the first parameter.

Errors
2501 Invalid object
2502 Invalid index

© 2016-2022 Altova GmbH

29.3.2.2.4 InsertItem

Method: InsertItem `(IndexPosition as long, MenuItemName as string, MacroName as string)`

Return Value
Inserts a user-defined menu item at the position in the menu specified in the first parameter and having the name submitted in the second parameter. The menu item will start a macro, so a valid macro name must be submitted.

Errors

2501	Invalid object
2502	Invalid index
2503	No such macro
2504	Internal error

29.3.2.2.5 SetItemText

Method: SetItemText `(IndexPosition as long, MenuItemName as string)`

Return Value
Sets the name of the menu item located at the index position submitted in the first parameter.

Errors

2501	Invalid object
2502	Invalid index

29.3.2.3 AuthenticDataTransfer

Renamed from DocEditDataTransfer to AuthenticDataTransfer

The DocEditView object is renamed to OldAuthenticView.
DocEditSelection is renamed to AuthenticSelection.
DocEditEvent is renamed to AuthenticEvent.
DocEditDataTransfer is renamed to AuthenticDataTransfer.

Their usage—except for AuthenticDataTransfer—is no longer recommended. We will continue to support existing functionality for a yet undefined period of time but no new features will be added to these interfaces. All functionality available up to now in DocEditView[1965], DocEditSelection[1963], DocEditEvent[1950] and DocEditDataTransfer[1746] is now available via AuthenticView[1778], AuthenticRange[1750] and AuthenticDataTransfer[1745]. Many new features have been added.

For examples on migrating from DocEdit to Authentic see the description of the different methods and properties of the different DocEdit objects.

Methods

getData 1746

Properties

dropEffect 1746
ownDrag 1746
type 1747

Description

The events OnDragOver and OnBeforeDrop provide information about the object being dragged with an instance of type AuthenticDataTransfer. It contains a description of the dragged object and its content. The latter is available either as string or a pointer to a COM object supporting the IUnkown interface.

29.3.2.3.1 dropEffect

Property: dropEffect as long

Description
The property stores the drop effect from the default event handler. You can set the drop effect if you change this value and return TRUE for the event handler (or set AuthenticEvent.cancelBubble 1954 to TRUE if you are still using the now obsolete AuthenticEvent interface).

Errors
2101 Invalid address for the return parameter was specified.

29.3.2.3.2 getData

Method: getData() as Variant

Description
Retrieve the data associated with the dragged object. Depending on AuthenticDataTransfer.type 1747, that data is either a string or a COM interface pointer of type IUnknown.

Errors
2101 Invalid address for the return parameter was specified.

29.3.2.3.3 ownDrag

Property: ownDrag as Boolean (read-only)

Description
The property is TRUE if the current dragging source comes from inside vista Authentic.

Errors

© 2016-2022 Altova GmbH

2101 Invalid address for the return parameter was specified.

29.3.2.3.4 type

Property: type as String (read-only)

Description
Holds the type of data you get with the DocEditDataTransfer.getData[1746] method.

Currently supported data types are:

OWN	data from vista Authentic itself
TEXT	plain text
UNICODETEXT	plain text as UNICODE

Errors
2101 Invalid address for the return parameter was specified.

29.3.2.4 AuthenticEventContext

The `EventContext` interface gives access to many properties of the context in which a macro is executed.

29.3.2.4.1 EvaluateXPath

Method: EvaluateXPath (strExpression as string) as strValue as string

Return Value
The method evaluates the XPath expression in the context of the node within which the event was triggered and returns a string.

Description
`EvaluateXPath()` executes an XPath expression with the given event context. The result is returned as a string, in the case of a sequence it is a space-separated string.

Errors
2201 Invalid object.
2202 No context.
2209 Invalid parameter.
2210 Internal error.
2211 XPath error.

29.3.2.4.2 GetEventContextType

Method: GetEventContextType `()` Type as `AuthenticEventContextType` enumeration

Return Value
Returns the context node type.

Description
`GetEventContextType` allows the user to determine whether the macro is in an XML node or in an XPath atomic item context. The enumeration `AuthenticEventContextType` is defined as follows:

authenticEventContextXML,
authenticEventContextAtomicItem,
authenticEventContextOther

If the context is a normal XML node, the `GetXMLNode()` function gives access to it (returns `NULL` if not).

Errors
2201 Invalid object.
2202 No context.
2209 Invalid parameter.

29.3.2.4.3 GetNormalizedTextValue

Method: GetNormalizedTextValue`()` strValue as string

Return Value
Returns the value of the current node as string

Errors
2201 Invalid object.
2202 No context.
2203 Invalid context
2209 Invalid parameter.

29.3.2.4.4 GetVariableValue

Method: GetVariableValue`(strName as string) strValue as string`

Return Value
Gets the value of the variable submitted as the parameter.

Description
`GetVariableValue` gets the variable's value in the scope of the context.

```
nZoom = parseInt( AuthenticView.EventContext.GetVariableValue( 'Zoom' ) );
if ( nZoom > 1 )
{
    AuthenticView.EventContext.SetVariableValue( 'Zoom', nZoom - 1 );
}
```

Errors

© 2016-2022 Altova GmbH

2201 Invalid object.
2202 No context.
2204 No such variable in scope
2205 Variable cannot be evaluated
2206 Variable returns sequence
2209 Invalid parameter

29.3.2.4.5 GetXMLNode

Method: GetXMLNode`()` Node as XMLData object

Return Value
Returns the context XML node or `NULL`

Errors
2201 Invalid object.
2202 No context.
2203 Invalid context
2209 Invalid parameter.

29.3.2.4.6 IsAvailable

Method: IsAvailable`()` as Boolean

Return Value
Returns true if `EventContext` is set, false otherwise.

Errors
2201 Invalid object.

29.3.2.4.7 SetVariableValue

Method: SetVariableValue`(strName as string, strValue as string)`

Return Value
Sets the value (second parameter) of the variable submitted in the first parameter.

Description
`SetVariableValue` sets the variable's value in the scope of the context.

```
nZoom = parseInt( AuthenticView.EventContext.GetVariableValue( 'Zoom' ) );
if ( nZoom > 1 )
```

```
{
    AuthenticView.EventContext.SetVariableValue( 'Zoom', nZoom - 1 );
}
```

Errors

2201	Invalid object.
2202	No context.
2204	No such variable in scope
2205	Variable cannot be evaluated
2206	Variable returns sequence
2207	Variable read-only
2208	No modification allowed

29.3.2.5 AuthenticRange

The first table lists the properties and methods of AuthenticRange that can be used to navigate through the document and select specific portions.

Properties	Methods	
Application 1752	Clone 1754	MoveBegin 1772
FirstTextPosition 1757	CollapsToBegin 1754	MoveEnd 1772
FirstXMLData 1758	CollapsToEnd 1754	NextCursorPosition 1763
FirstXMLDataOffset 1759	ExpandTo 1757	PreviousCursorPosition 1764
LastTextPosition 1769	Goto 1762	Select 1774
LastXMLData 1770	GotoNext 1763	SelectNext 1775
LastXMLDataOffset 1770	GotoPrevious 1764	SelectPrevious 1775
Parent 1773	IsEmpty 1767	SetFromRange 1777
	IsEqual 1767	

The following table lists the content modification methods, most of which can be found on the right/button mouse menu.

Properties	Edit operations	Dynamic table operations
Text 1778	Copy 1754	AppendRow 1752
	Cut 1755	DeleteRow 1755
	Delete 1755	DuplicateRow 1756
	IsCopyEnabled 1766	InsertRow 1766

© 2016-2022 Altova GmbH

	IsCutEnabled 1766	IsFirstRow 1767
	IsDeleteEnabled 1767	IsInDynamicTable 1768
	IsPasteEnabled 1768	IsLastRow 1768
	Paste 1773	MoveRowDown 1772
		MoveRowUp 1773

The following methods provide the functionality of the Authentic entry helper windows for range objects.

Operations of the entry helper windows		
Elements	**Attributes**	**Entities**
CanPerformActionWith 1753	GetElementAttributeValue 1760	GetEntityNames 1762
CanPerformAction 1753	GetElementAttributeNames 1760	InsertEntity 1765
PerformAction 1773	GetElementHierarchy 1760	
	HasElementAttribute 1765	
	IsTextStateApplied 1769	
	SetElementAttributeValue 1776	

Description
AuthenticRange objects are the 'cursor' selections of the automation interface. You can use them to point to any cursor position in the Authentic view, or select a portion of the document. The operations available for AuthenticRange objects then work on this selection in the same way, as the corresponding operations of the user interface do with the current user interface selection. The main difference is that you can use an arbitrary number of AuthenticRange objects at the same time, whereas there is exactly one cursor selection in the user interface.

To get to an initial range object use AuthenticView.Selection 1795, to obtain a range corresponding with the current cursor selection in the user interface. Alternatively, some trivial ranges are accessible via the read/only properties AuthenticView.DocumentBegin 1790, AuthenticView.DocumentEnd 1791, and AuthenticView.WholeDocument 1797. The most flexible method is AuthenticView.Goto 1793, which allows navigation to a specific portion of the document within one call. For more complex selections, combine the above with the various navigation methods on range objects listed in the first table on this page.

Another method to select a portion of the document is to use the position properties of the range object. Two positioning systems are available and can be combined arbitrarily:

- **Absolute** text cursor positions, starting with position 0 at the document beginning, can be set and retrieved for the beginning and end of a range. For more information see FirstTextPosition 1757 and LastTextPosition 1769. This method requires complex internal calculations and should be used with care.

- The **XMLData** element and a text position inside this element, can be set and retrieved for the beginning and end of a range. For more information see FirstXMLData 1758, FirstXMLDataOffset 1759,

LastXMLData[1770], and LastXMLDataOffset[1770]. This method is very efficient but requires knowledge of the underlying document structure. It can be used to locate XMLData objects and perform operations on them otherwise not accessible through the user interface.

Modifications to the document content can be achieved by various methods:

- The Text[1778] property allows you to retrieve the document text selected by the range object. If set, the selected document text gets replaced with the new text.
- The standard document edit functions Cut[1755], Copy[1754], Paste[1773] and Delete[1755].
- Table operations for tables that can grow dynamically.
- Methods that map the functionality of the Authentic entry helper windows.
- Access to the XMLData[1939] objects of the underlying document to modify them directly.

29.3.2.5.1 AppendRow

Method: AppendRow()as Boolean

Description
If the beginning of the range is inside a dynamic table, this method inserts a new row at the end of the selected table. The selection of the range is modified to point to the beginning of the new row. The function returns *true* if the append operation was successful, otherwise *false*.

Errors
2001 The authentic range object or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

Examples

```
' ---------------------------------------------------------
' Scripting environment - VBScript
' Append row at end of current dynamically growable table
' ---------------------------------------------------------
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
If objRange.IsInDynamicTable Then
        objRange.AppendRow
        ' objRange points to beginning of new row
        objRange.Select
End If
```

29.3.2.5.2 Application

Property: Application as Application[1724] (read-only)

Description
Accesses the XMLSpy application object.

© 2016-2022 Altova GmbH

Errors
2001 The authentic range object or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.3 CanPerformAction

Method: CanPerformAction (*eAction* as SPYAuthenticActions[1988], *strElementName* as String) as Boolean

Description
CanPerformAction and its related methods enable access to the entry-helper functions of Authentic. This function allows easy and consistent modification of the document content, without having to know exactly where the modification will take place. The beginning of the range object is used to locate the next valid location where the specified action can be performed. If the location can be found, the method returns *True*, otherwise it returns *False*.

HINT: To find out all valid element names for a given action, use CanPerformActionWith[1753].

Errors
2001 The authentic range object or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.
2007 Invalid action was specified.

Examples
See PerformAction[1773].

29.3.2.5.4 CanPerformActionWith

Method: CanPerformActionWith (*eAction* as SPYAuthenticActions[1988], *out_arrElementNames* as Variant)

Description
PerformActionWith and its related methods, enable access to the entry-helper functions of Authentic. This function allows easy and consistent modification of the document content without having to know exactly where the modification will take place.

This method returns an array of those element names that the specified action can be performed with.

HINT: To apply the action use CanPerformActionWith[1753].

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.
2007 Invalid action was specified.

Examples
See PerformAction[1773].

29.3.2.5.5 Clone

Method: Clone() as AuthenticRange[1750]

Description
Returns a copy of the range object.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.6 CollapsToBegin

Method: CollapsToBegin() as AuthenticRange[1750]

Description
Sets the end of the range object to its begin. The method returns the modified range object.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.7 CollapsToEnd

Method: CollapsToEnd() as AuthenticRange[1750]

Description
Sets the beginning of the range object to its end. The method returns the modified range object.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.8 Copy

Method: Copy() as Boolean

Description
Returns *False* if the range contains no portions of the document that may be copied.
Returns *True* if text, and in case of fully selected XML elements the elements as well, has been copied to the copy/paste buffer.

Errors
2001 The authentic range object or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.5.9 Cut

Method: Cut() as Boolean

Description
Returns *False* if the range contains portions of the document that may not be deleted.
Returns *True* after text, and in case of fully selected XML elements the elements as well, has been deleted from the document and saved in the copy/paste buffer.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.10 Delete

Method: Delete() as Boolean

Description
Returns *False* if the range contains portions of the document that may not be deleted.
Returns *True* after text, and in case of fully selected XML elements the elements as well, has been deleted from the document.

Errors
2001 The authentic range object or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.11 DeleteRow

Method: DeleteRow() as Boolean

Description
If the beginning of the range is inside a dynamic table, this method deletes the selected row. The selection of the range gets modified to point to the next element after the deleted row. The function returns *true,* if the delete operation was successful, otherwise *false*.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

Examples

```
' ----------------------------------------------------------
' Scripting environment - VBScript
' Delete selected row from dynamically growing table
' ----------------------------------------------------------
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection
```

```
' check if we are in a table
If objRange.IsInDynamicTable Then
        objRange.DeleteRow
End If
```

29.3.2.5.12 DuplicateRow

Method: DuplicateRow() as Boolean

Description
If the beginning of the range is inside a dynamic table, this method inserts a duplicate of the current row after the selected one. The selection of the range gets modified to point to the beginning of the new row. The function returns *true* if the duplicate operation was successful, otherwise *false*.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

Examples

```
' ----------------------------------------------------------
' Scripting environment - VBScript
' duplicate row in current dynamically growable table
' ----------------------------------------------------------
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
If objRange.IsInDynamicTable Then
        objRange.DuplicateRow
        ' objRange points to beginning of new row
        objRange.Select
End If
```

29.3.2.5.13 EvaluateXPath

Method: EvaluateXPath (strExpression as string) strValue as string

Return Value
The method returns a string

Description
`EvaluateXPath()` executes an XPath expression with the context node being the beginning of the range selection. The result is returned as a string, in the case of a sequence it is a space-separated string. If XML context node is irrelevant, the user may provide any node, like `AuthenticView.XMLDataRoot`.

Errors
2001 Invalid object
2005 Invalid parameter

© 2016-2022 Altova GmbH

2008 Internal error
2202 Missing context node
2211 XPath error

29.3.2.5.14 ExpandTo

Method: ExpandTo (*eKind* as SPYAuthenticElementKind[1989]), as AuthenticRange[1750]

Description
Selects the whole element of type eKind, that starts at, or contains, the first cursor position of the range. The method returns the modified range object.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2003 Range expansion would be beyond end of document.
2005 Invalid address for the return parameter was specified.

29.3.2.5.15 FirstTextPosition

Property: FirstTextPosition as Long

Description
Set or get the left-most text position index of the range object. This index is always less or equal to LastTextPosition[1769]. Indexing starts with 0 at document beginning, and increments with every different position that the text cursor can occupy. Incrementing the test position by 1, has the same effect as the cursor-right key. Decrementing the test position by 1 has the same effect as the cursor-left key.

If you set FirstTextPosition to a value greater than the current LastTextPosition[1769], LastTextPosition[1769] gets set to the new FirstTextPosition.

HINT: Use text cursor positions with care, since this is a costly operation compared to XMLData based cursor positioning.

Errors
2001 The authentic range object, or its related view object is not valid.
2005 Invalid address for the return parameter was specified.
2006 A text position outside the document was specified.

Examples

```
' ---------------------------------------
' Scripting environment - VBScript
' ---------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

nDocStartPosition = objAuthenticView.DocumentBegin.FirstTextPosition
nDocEndPosition = objAuthenticView.DocumentEnd.FirstTextPosition
```

```
' let's create a range that selects the whole document
' in an inefficient way
Dim objRange
' we need to get a (any) range object first
Set objRange = objAuthenticView.DocumentBegin
objRange.FirstTextPosition = nDocStartPosition
objRange.LastTextPosition = nDocEndPosition

' let's check if we got it right
If objRange.isEqual(objAuthenticView.WholeDocument) Then
        MsgBox "Test using direct text cursor positioning was ok"
Else
        MsgBox "Ooops!"
End If
```

29.3.2.5.16 FirstXMLData

Property: FirstXMLData as XMLData[1939]

Description
Set or get the first XMLData element in the underlying document that is partially, or completely selected by the range. The exact beginning of the selection is defined by the FirstXMLDataOffset[1759] attribute.

Whenever you set FirstXMLData to a new data object, FirstXMLDataOffset[1759] gets set to the first cursor position inside this element. Only XMLData objects that have a cursor position may be used. If you set FirstXMLData / FirstXMLDataOffset[1759] selects a position greater then the current LastXMLData[1770] / LastXMLDataOffset[1770], the latter gets moved to the new start position.

HINT: You can use the FirstXMLData[1758] and LastXMLData[1770] properties to directly access and manipulate the underlying XML document in those cases where the methods available with the AuthenticRange[1750] object are not sufficient.

Errors

2001	The authentic range object, or its related view object is not valid.
2005	Invalid address for the return parameter was specified.
2008	Internal error
2009	The XMLData object cannot be accessed.

Examples

```
' ----------------------------------------------
' Scripting environment - VBScript
' show name of currently selected XMLData element
' ----------------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objXmlData
Set objXMLData = objAuthenticView.Selection.FirstXMLData
' authentic view adds a 'text' child element to elements
' of the document which have content. So we have to go one
```

 © 2016-2022 Altova GmbH

```
' element up.
Set objXMLData = objXMLData.Parent
MsgBox "Current selection selects element " & objXMLData.Name
```

29.3.2.5.17 FirstXMLDataOffset

Property: FirstXMLDataOffset as Long

Description
Set or get the cursor position offset inside FirstXMLData[1758] element for the beginning of the range. Offset positions are based on the characters returned by the Text[1778] property, and start with 0. When setting a new offset, use -1 to set the offset to the last possible position in the element. The following cases require specific attention:

- The textual form of entries in Combo Boxes, Check Boxes and similar controls can be different from what you see on screen. Although the data offset is based on this text, there only two valid offset positions, one at the beginning and one at the end of the entry. An attempt to set the offset to somewhere in the middle of the entry, will result in the offset being set to the end.

- The textual form of XML Entities might differ in length from their representation on the screen. The offset is based on this textual form.

If FirstXMLData / FirstXMLDataOffset[1759] selects a position after the current LastXMLData[1770] / LastXMLDataOffset[1770], the latter gets moved to the new start position.

Errors

2001	The authentic range object, or its related view object is not valid.
2005	Invalid offset was specified.
	Invalid address for the return parameter was specified.

Examples

```
' ---------------------------------------------
' Scripting environment - VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
' ---------------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

' first we use the XMLData based range properties
' to select all text of the first XMLData element
' in the current selection
Dim objRange
Set objRange = objAuthenticView.Selection
objRange.FirstXMLDataOffset = 0  ' start at beginning of element text
objRange.LastXMLData = objRange.FirstXMLData  ' select only one element
objRange.LastXMLDataOffset = -1  ' select till its end

' the same can be achieved with the ExpandTo method
Dim objRange2
```

```
Set objRange2 = objAuthenticView.Selection.ExpandTo(spyAuthenticTag)

' were we successful?
If objRange.IsEqual(objRange2) Then
        objRange.Select()
Else
        MsgBox "Oops"
End If
```

29.3.2.5.18 GetElementAttributeNames

Method: GetElementAttributeNames (*strElementName* as String, *out_arrAttributeNames* as Variant)

Description
Retrieve the names of all attributes for the enclosing element with the specified name. Use the element/attribute pairs, to set or get the attribute value with the methods GetElementAttributeValue[1760] and SetElementAttributeValue[1776].

Errors

2001	The authentic range object, or its related view object is no longer valid.
2005	Invalid element name was specified.
	Invalid address for the return parameter was specified.

Examples
See SetElementAttributeValue[1776].

29.3.2.5.19 GetElementAttributeValue

Method: GetElementAttributeValue (*strElementName* as String, *strAttributeName* as String) as String

Description
Retrieve the value of the attribute specified in strAttributeName, for the element identified with strElementName. If the attribute is supported but has no value assigned, the empty string is returned. To find out the names of attributes supported by an element, use GetElementAttributeNames[1760], or HasElementAttribute[1765].

Errors

2001	The authentic range object, or its related view object is no longer valid.
2005	Invalid element name was specified.
	Invalid attribute name was specified.
	Invalid address for the return parameter was specified.

Examples
See SetElementAttributeValue[1776].

29.3.2.5.20 GetElementHierarchy

Method: GetElementHierarchy (*out_arrElementNames* as Variant)

© 2016-2022 Altova GmbH

Description
Retrieve the names of all XML elements that are parents of the current selection. Inner elements get listed before enclosing elements. An empty list is returned whenever the current selection is not inside a single XMLData element.

The names of the element hierarchy, together with the range object uniquely identify XMLData elements in the document. The attributes of these elements can be directly accessed by GetElementAttributeNames[1760], and related methods.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

C# Examples

```
' ----------------------------------------
' C#
' ----------------------------------------

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            XMLSpyLib.Application app = new XMLSpyLib.Application();

            app.ShowApplication(true);

            XMLSpyLib.AuthenticView view = app.ActiveDocument.AuthenticView;
            XMLSpyLib.AuthenticRange range = view.DocumentBegin;

            object o = null;
            range.GetElementHierarchy(ref o);

            object[] elements = (object[])o;

            foreach (string e in elements)
            {
                Console.WriteLine(e);
            }
        }
    }
}
```

Also see: SetElementAttributeValue[1776].

29.3.2.5.21 GetEntityNames

Method: GetEntityNames (*out_arrEntityNames* as Variant)

Description
Retrieve the names of all defined entities. The list of retrieved entities is independent of the current selection, or location. Use one of these names with the InsertEntity[1765] function.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

Examples
See: GetElementHierarchy[1760] and InsertEntity[1765].

29.3.2.5.22 GetVariableValue

Method: GetVariableValue `(strName as string) strVal as string`

Return Value
Gets the value of the variable named as the method's parameter.

Errors
2001 Invalid object.
2202 No context.
2204 No such variable in scope
2205 Variable cannot be evaluated
2206 Variable returns sequence
2209 Invalid parameter

29.3.2.5.23 Goto

Method: Goto (*eKind* as SPYAuthenticElementKind[1989], *nCount* as Long, *eFrom* as SPYAuthenticDocumentPosition[1988]) as AuthenticRange[1750]

Description
Sets the range to point to the beginning of the nCount element of type eKind. The start position is defined by the parameter eFrom.

Use positive values for nCount to navigate to the document end. Use negative values to navigate to the beginning of the document. The method returns the modified range object.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2003 Target lies after end of document.

© 2016-2022 Altova GmbH

2004 Target lies before begin of document.
2005 Invalid element kind specified.
Invalid start position specified.
Invalid address for the return parameter was specified.

29.3.2.5.24 GotoNext

Method: GotoNext (*eKind* as SPYAuthenticElementKind[1989]) as AuthenticRange[1750]

Description
Sets the range to the beginning of the next element of type eKind. The method returns the modified range object.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2003 Target lies after end of document.
2005 Invalid element kind specified.
Invalid address for the return parameter was specified.

Examples

```
' ----------------------------------------
' Scripting environment - VBScript
' Scan through the whole document word-by-word
' ----------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentBegin
Dim bEndOfDocument
bEndOfDocument = False

On Error Resume Next
While Not bEndOfDocument
        objRange.GotoNext(spyAuthenticWord).Select
        If ((Err.number - vbObjecterror) = 2003) Then
                bEndOfDocument = True
                Err.Clear
        ElseIf (Err.number <> 0) Then
                Err.Raise  ' forward error
        End If
Wend
```

29.3.2.5.25 GotoNextCursorPosition

Method: GotoNextCursorPosition() as AuthenticRange[1750]

Description
Sets the range to the next cursor position after its current end position. Returns the modified object.

Errors

2001 The authentic range object, or its related view object is no longer valid.
2003 Target lies after end of document.
2005 Invalid address for the return parameter was specified.

29.3.2.5.26 GotoPrevious

Method: GotoPrevious (*eKind* as SPYAuthenticElementKind[1989]) as AuthenticRange[1750]

Description

Sets the range to the beginning of the element of type eKind which is before the beginning of the current range. The method returns the modified range object.

Errors

2001 The authentic range object, or its related view object is no longer valid.
2004 Target lies before beginning of document.
2005 Invalid element kind specified.
Invalid address for the return parameter was specified.

Examples

```
' ----------------------------------------
' Scripting environment - VBScript
' Scan through the whole document tag-by-tag
' ----------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bBeginOfDocument
bBeginOfDocument = False

On Error Resume Next
While Not bBeginOfDocument
	objRange.GotoPrevious(spyAuthenticTag).Select
	If ((Err.number - vbObjecterror) = 2004) Then
		bBeginOfDocument = True
		Err.Clear
	ElseIf (Err.number <> 0) Then
		Err.Raise  ' forward error
	End If
Wend
```

29.3.2.5.27 GotoPreviousCursorPosition

Method: GotoPreviousCursorPosition() as AuthenticRange[1750]

© 2016-2022 Altova GmbH

Description
Set the range to the cursor position immediately before the current position. Returns the modified object.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2004 Target lies before begin of document.
2005 Invalid address for the return parameter was specified.

29.3.2.5.28 HasElementAttribute

Method: HasElementAttribute (*strElementName* as String, *strAttributeName* as String) as Boolean

Description
Tests if the enclosing element with name strElementName, supports the attribute specified in strAttributeName.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid element name was specified.
Invalid address for the return parameter was specified.

29.3.2.5.29 InsertEntity

Method: InsertEntity (*strEntityName* as String)

Description
Replace the ranges selection with the specified entity. The specified entity must be one of the entity names returned by GetEntityNames 1762.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Unknown entry name was specified.

Examples

```
' ----------------------------------------------------------
' Scripting environment - VBScript
' Insert the first entity in the list of available entities
' ----------------------------------------------------------
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' first we get the names of all available entities as they
' are shown in the entry helper of XMLSpy
Dim arrEntities
objRange.GetEntityNames arrEntities

' we insert the first one of the list
If UBound(arrEntities) >= 0 Then
        objRange.InsertEntity arrEntities(0)
```

```
Else
        MsgBox "Sorry, no entities are available for this document"
End If
```

29.3.2.5.30 InsertRow

Method: InsertRow() as Boolean

Description
If the beginning of the range is inside a dynamic table, this method inserts a new row before the current one. The selection of the range gets modified to point to the beginning of the newly inserted row. The function returns *true* if the insert operation was successful, otherwise *false*.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

Examples

```
' ------------------------------------------------------------
' Scripting environment - VBScript
' Insert row at beginning of current dynamically growing table
' ------------------------------------------------------------
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
If objRange.IsInDynamicTable Then
        objRange.InsertRow
        ' objRange points to beginning of new row
        objRange.Select
  End If
```

29.3.2.5.31 IsCopyEnabled

Property: IsCopyEnabled as Boolean (read-only)

Description
Checks if the copy operation is supported for this range.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.32 IsCutEnabled

Property: IsCutEnabled as Boolean (read-only)

© 2016-2022 Altova GmbH

Description
Checks if the cut operation is supported for this range.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.33 IsDeleteEnabled

Property: IsDeleteEnabled as Boolean (read-only)

Description
Checks if the delete operation is supported for this range.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.34 IsEmpty

Method: IsEmpty() as Boolean

Description
Tests if the first and last position of the range are equal.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.35 IsEqual

Method: IsEqual (*objCmpRange* as AuthenticRange[1750]) as Boolean

Description
Tests if the start and end of both ranges are the same.

Errors
2001 One of the two range objects being compared, is invalid.
2005 Invalid address for a return parameter was specified.

29.3.2.5.36 IsFirstRow

Property: IsFirstRow as Boolean (read-only)

Description

Test if the range is in the first row of a table. Which table is taken into consideration depends on the extent of the range. If the selection exceeds a single row of a table, the check is if this table is the first element in an embedding table. See the entry helpers of the user manual for more information.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.37 IsInDynamicTable

Method: IsInDynamicTable() as Boolean

Description
Test if the whole range is inside a table that supports the different row operations like 'insert', 'append', duplicate, etc.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.38 IsLastRow

Property: IsLastRow as Boolean (read-only)

Description
Test if the range is in the last row of a table. Which table is taken into consideration depends on the extent of the range. If the selection exceeds a single row of a table, the check is if this table is the last element in an embedding table. See the entry helpers of the user manual for more information.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.39 IsPasteEnabled

Property: IsPasteEnabled as Boolean (read-only)

Description
Checks if the paste operation is supported for this range.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.5.40 IsSelected

Property: IsSelected as Boolean

Description
Returns true() if selection is present. The selection range still can be empty: that happens when e.g. only the cursor is set.

29.3.2.5.41 IsTextStateApplied

Method: IsTextStateApplied (*i_strElementName* as String) as Boolean

Description
Checks if all the selected text is embedded into an XML Element with name i_strElementName. Common examples for the parameter i_strElementName are "strong", "bold" or "italic".

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.42 LastTextPosition

Property: LastTextPosition as Long

Description
Set or get the rightmost text position index of the range object. This index is always greater or equal to FirstTextPosition[1757]. Indexing starts with 0 at the document beginning, and increments with every different position that the text cursor can occupy. Incrementing the test position by 1, has the same effect as the cursor-right key. Decreasing the test position by 1 has the same effect as the cursor-left key.

If you set LastTextPosition to a value less then the current FirstTextPosition[1757], FirstTextPosition[1757] gets set to the new LastTextPosition.

HINT: Use text cursor positions with care, since this is a costly operation compared to XMLData based cursor positioning.

Errors
2001 The authentic range object, or its related view object is not valid.
2005 Invalid address for the return parameter was specified.
2006 A text position outside the document was specified.

Examples

```
' ------------------------------------
' Scripting environment - VBScript
' ------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
```

```
Set objAuthenticView = Application.ActiveDocument.AuthenticView

nDocStartPosition = objAuthenticView.DocumentBegin.FirstTextPosition
nDocEndPosition = objAuthenticView.DocumentEnd.FirstTextPosition

' let's create a range that selects the whole document
' in an inefficient way
Dim objRange
' we need to get a (any) range object first
Set objRange = objAuthenticView.DocumentBegin
objRange.FirstTextPosition = nDocStartPosition
objRange.LastTextPosition = nDocEndPosition

' let's check if we got it right
If objRange.isEqual(objAuthenticView.WholeDocument) Then
        MsgBox "Test using direct text cursor positioning was ok"
Else
        MsgBox "Oops!"
End If
```

29.3.2.5.43 LastXMLData

Property: LastXMLData as XMLData[1939]

Description
Set or get the last XMLData element in the underlying document that is partially or completely selected by the range. The exact end of the selection is defined by the LastXMLDataOffset[1770] attribute.

Whenever you set LastXMLData to a new data object, LastXMLDataOffset[1770] gets set to the last cursor position inside this element. Only XMLData objects that have a cursor position may be used. If you set LastXMLData / LastXMLDataOffset[1770], select a position less then the current FirstXMLData[1758] / FirstXMLDataOffset[1759], the latter gets moved to the new end position.

HINT: You can use the FirstXMLData[1758] and LastXMLData[1770] properties to directly access and manipulate the underlying XML document in those cases, where the methods available with the AuthenticRange[1750] object are not sufficient.

Errors

2001	The authentic range object, or its related view object is not valid.
2005	Invalid address for the return parameter was specified.
2008	Internal error
2009	The XMLData object cannot be accessed.

29.3.2.5.44 LastXMLDataOffset

Property: LastXMLDataOffset as Long

Description
Set or get the cursor position inside LastXMLData[1770] element for the end of the range.

© 2016-2022 Altova GmbH

Offset positions are based on the characters returned by the Text[1778] property and start with 0. When setting a new offset, use -1 to set the offset to the last possible position in the element. The following cases require specific attention:

- The textual form of entries in Combo Boxes, Check Boxes and similar controls can be different from what you see on the screen. Although, the data offset is based on this text, there only two valid offset positions, one at the beginning and one at the end of the entry. An attempt to set the offset to somewhere in the middle of the entry, will result in the offset being set to the end.
- The textual form of XML Entities might differ in length from their representation on the screen. The offset is based on this textual form.

If LastXMLData[1770] / LastXMLDataOffset[1770] selects a position before FirstXMLData[1758] / FirstXMLDataOffset[1759], the latter gets moved to the new end position.

Errors

2001	The authentic range object, or its related view object is not valid.
2005	Invalid offset was specified.
	Invalid address for the return parameter was specified.

Examples

```
' ---------------------------------------------
' Scripting environment - VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
' ---------------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

' first we use the XMLData based range properties
' to select all text of the first XMLData element
' in the current selection
Dim objRange
Set objRange = objAuthenticView.Selection
objRange.FirstXMLDataOffset = 0  ' start at beginning of element text
objRange.LastXMLData = objRange.FirstXMLData  ' select only one element
objRange.LastXMLDataOffset = -1  ' select till its end

' the same can be achieved with the ExpandTo method
Dim objRange2
Set objRange2 = objAuthenticView.Selection.ExpandTo(spyAuthenticTag)

' were we successful?
If objRange.IsEqual(objRange2) Then
        objRange.Select()
Else
        MsgBox "Ooops"
End If
```

29.3.2.5.45 MoveBegin

Method: MoveBegin (*eKind* as SPYAuthenticElementKind[1989], *nCount* as Long) as AuthenticRange[1750]

Description
Move the beginning of the range to the beginning of the nCount element of type eKind. Counting starts at the current beginning of the range object.

Use positive numbers for nCount to move towards the document end, use negative numbers to move towards document beginning. The end of the range stays unmoved, unless the new beginning would be larger than it. In this case, the end is moved to the new beginning. The method returns the modified range object.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2003	Target lies after end of document.
2004	Target lies before beginning of document.
2005	Invalid element kind specified.
	Invalid address for the return parameter was specified.

29.3.2.5.46 MoveEnd

Method: MoveEnd (*eKind* as SPYAuthenticElementKind[1989], *nCount* as Long) as AuthenticRange[1750]

Description
Move the end of the range to the begin of the nCount element of type eKind. Counting starts at the current end of the range object.

Use positive numbers for nCount to move towards the document end, use negative numbers to move towards document beginning. The beginning of the range stays unmoved, unless the new end would be less than it. In this case, the beginning gets moved to the new end. The method returns the modified range object.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2003	Target lies after end of document.
2004	Target lies before begin of document.
2005	Invalid element kind specified.
	Invalid address for the return parameter was specified.

29.3.2.5.47 MoveRowDown

Method: MoveRowDown() as Boolean

Description
If the beginning of the range is inside a dynamic table and selects a row which is not the last row in this table, this method swaps this row with the row immediately below. The selection of the range moves with the row, but does not otherwise change. The function returns *true* if the move operation was successful, otherwise *false*.

Errors

© 2016-2022 Altova GmbH

2001 The authentic range object or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.48 MoveRowUp

Method: MoveRowUp() as Boolean

Description
If the beginning of the range is inside a dynamic table and selects a row which is not the first row in this table, this method swaps this row with the row above. The selection of the range moves with the row, but does not change otherwise. The function returns *true* if the move operation was successful, otherwise *false*.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.49 Parent

Property: Parent as AuthenticView[1778] (read-only)

Description
Access the view that owns this range object.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.50 Paste

Method: Paste() as Boolean

Description
Returns *False* if the copy/paste buffer is empty, or its content cannot replace the current selection.

Otherwise, deletes the current selection, inserts the content of the copy/paste buffer, and returns *True*.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.51 PerformAction

Method: PerformAction (*eAction* as SPYAuthenticActions[1988], *strElementName* as String) as Boolean

Description
PerformAction and its related methods, give access to the entry-helper functions of Authentic. This function allows easy and consistent modification of the document content without a need to know exactly where the

modification will take place. The beginning of the range object is used to locate the next valid location where the specified action can be performed. If no such location can be found, the method returns *False*. Otherwise, the document gets modified and the range points to the beginning of the modification.

HINT: To find out element names that can be passed as the second parameter use CanPerformActionWith[1753].

Errors

2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.
2007 Invalid action was specified.

Examples

```
' ----------------------------------------
' Scripting environment - VBScript
' Insert the innermost element
' ----------------------------------------
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' we determine the elements that can be inserted at the current position
Dim arrElements()
objRange.CanPerformActionWith spyAuthenticInsertBefore, arrElements

' we insert the first (innermost) element
If UBound(arrElements) >= 0 Then
        objRange.PerformAction spyAuthenticInsertBefore, arrElements(0)
        ' objRange now points to the beginning of the inserted element
        ' we set a default value and position at its end
        objRange.Text = "Hello"
        objRange.ExpandTo(spyAuthenticTag).CollapsToEnd().Select
Else
        MsgBox "Can't insert any elements at current position"
End If
```

29.3.2.5.52 Select

Method: Select()

Description

Makes this range the current user interface selection. You can achieve the same result using: '*objRange.Parent.Selection = objRange*'

Errors

2001 The authentic range object or its related view object is no longer valid.

Examples

```
' ----------------------------------
' Scripting environment - VBScript
' ----------------------------------
Dim objAuthenticView
```


© 2016-2022 Altova GmbH

```
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

' set current selection to end of document
objAuthenticView.DocumentEnd.Select()
```

29.3.2.5.53 SelectNext

Method: SelectNext (*eKind* as SPYAuthenticElementKind[1989]) as AuthenticRange[1750]

Description
Selects the element of type eKind after the current end of the range. The method returns the modified range object.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2003	Target lies after end of document.
2005	Invalid element kind specified.
	Invalid address for the return parameter was specified.

Examples

```
' ----------------------------------------
' Scripting environment - VBScript
' Scan through the whole document word-by-word
' ----------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentBegin
Dim bEndOfDocument
bEndOfDocument = False

On Error Resume Next
While Not bEndOfDocument
	objRange.SelectNext(spyAuthenticWord).Select
	If ((Err.number - vbObjecterror) = 2003) Then
		bEndOfDocument = True
		Err.Clear
	ElseIf (Err.number <> 0) Then
		Err.Raise  ' forward error
	End If
Wend
```

29.3.2.5.54 SelectPrevious

Method: GotoPrevious (*eKind* as SPYAuthenticElementKind[1989]) as AuthenticRange[1750]

Description
Selects the element of type eKind before the current beginning of the range. The method returns the modified range object.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2004 Target lies before begin of document.
2005 Invalid element kind specified.
Invalid address for the return parameter was specified.

Examples

```
' ----------------------------------------
' Scripting environment - VBScript
' Scan through the whole document tag-by-tag
' ----------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bBeginOfDocument
bBeginOfDocument = False

On Error Resume Next
While Not bBeginOfDocument
        objRange.SelectPrevious(spyAuthenticTag).Select
        If ((Err.number - vbObjecterror) = 2004) Then
                bBeginOfDocument = True
                Err.Clear
        ElseIf (Err.number <> 0) Then
                Err.Raise  ' forward error
        End If
Wend
```

29.3.2.5.55 SetElementAttributeValue

Method: SetElementAttributeValue (*strElementName* as String, *strAttributeName* as String, *strAttributeValue* as String)

Description
Set the value of the attribute specified in strAttributeName for the element identified with strElementName. If the attribute is supported but has no value assigned, the empty string is returned. To find out the names of attributes supported by an element, use GetElementAttributeNames[1760], or HasElementAttribute[1765].

Errors
2001 The authentic range object or its related view object is no longer valid.
2005 Invalid element name was specified.
Invalid attribute name was specified.

© 2016-2022 Altova GmbH

Invalid attribute value was specified.

Examples

```
' ----------------------------------------
' Scripting environment - VBScript
' Get and set element attributes
' ----------------------------------------
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' first we find out all the elements below the beginning of the range
Dim arrElements
objRange.GetElementHierarchy arrElements

If IsArray(arrElements) Then
	If UBound(arrElements) >= 0 Then
		' we use the top level element and find out its valid attributes
		Dim arrAttrs()
		objRange.GetElementAttributeNames arrElements(0), arrAttrs

		If UBound(arrAttrs) >= 0 Then
			' we retrieve the current value of the first valid attribute
			Dim strAttrVal
			strAttrVal = objRange.GetElementAttributeValue (arrElements(0), arrAttrs(0))
			msgbox "current value of " & arrElements(0) & "//" & arrAttrs(0) & " is: " & strAttrVal

			' we change this value and read it again
			strAttrVal = "Hello"
			objRange.SetElementAttributeValue arrElements(0), arrAttrs(0), strAttrVal
			strAttrVal = objRange.GetElementAttributeValue (arrElements(0), arrAttrs(0))
			msgbox "new value of " & arrElements(0) & "//" & arrAttrs(0) & " is: " & strAttrVal
		End If
	End If
End If
```

29.3.2.5.56 SetFromRange

Method: SetFromRange (*objSrcRange* as AuthenticRange[1750])

Description

Sets the range object to the same beginning and end positions as objSrcRange.

Errors

2001	One of the two range objects, is invalid.
2005	Null object was specified as source object.

29.3.2.5.57 SetVariableValue

Method: SetVariableValue`(strName as string, strValue as string)`

Return Value
Sets the value (second parameter) of the variable named in the first parameter.

Errors

2201	Invalid object.
2202	No context.
2204	No such variable in scope
2205	Variable cannot be evaluated
2206	Variable returns sequence
2207	Variable read-only
2208	No modification allowed

29.3.2.5.58 Text

Property: Text as String

Description
Set or get the textual content selected by the range object.

The number of characters retrieved are not necessarily identical, as there are text cursor positions between the beginning and end of the selected range. Most document elements support an end cursor position different to the beginning cursor position of the following element. Drop-down lists maintain only one cursor position, but can select strings of any length. In the case of radio buttons and check boxes, the text property value holds the string of the corresponding XML element.

If the range selects more then one element, the text is the concatenation of the single texts. XML entities are expanded so that '&' is expected as '&'.

Setting the text to the empty string, does not delete any XML elements. Use Cut[1755], Delete[1755] or PerformAction[1773] instead.

Errors

2001	The authentic range object or its related view object is no longer valid.
2005	Invalid address for a return parameter was specified.

29.3.2.6 AuthenticView

Properties	Methods	Events
Application[1789]	Goto[1793]	OnBeforeCopy[1780]
AsXMLString[1789]	IsRedoEnabled[1793]	OnBeforeCut[1780]
DocumentBegin[1790]	IsUndoEnabled[1794]	OnBeforeDelete[1781]

© 2016-2022 Altova GmbH

DocumentEnd 1791
Event 1792
MarkupVisibility 1794
Parent 1794
Selection 1795
XMLDataRoot 1797
WholeDocument 1797
Print 1794
Redo 1795
Undo 1796
UpdateXMLInstanceEntities 1796
OnBeforeDrop 1781
OnBeforePaste 1782
OnDragOver 1783
OnKeyBoardEvent 1784
OnMouseEvent 1785
OnSelectionChanged 1786

Description

AuthenticView and its child objects AuthenticRange 1750 and AuthenticDataTransfer provide you with an interface for vista Authentic, which allow easy and consistent modification of document contents. These interfaces replace the following interfaces which are marked now as **obsolete:**

OldAuthenticView (old name was DocEditView)
AuthenticSelection (old name was DocEditSelection, superseded by AuthenticRange 1750)
AuthenticEvent (old name was DocEditEvent)

AuthenticView gives you easy access to specific features such as printing, the multi-level undo buffer, and the current cursor selection, or position.

AuthenticView uses objects of type AuthenticRange 1750 to make navigation inside the document straight-forward, and to allow for the flexible selection of logical text elements. Use the properties DocumentBegin 1790, DocumentEnd 1791, or WholeDocument 1797 for simple selections, while using the Goto 1793 method for more complex selections. To navigate relative to a given document range, see the methods and properties of the AuthenticRange 1750 object.

Examples

```
' ---------------------------------------
' XMLSpy scripting environment - VBScript
' secure access to authentic view object
' ---------------------------------------
Dim objDocument
Set objDocument = Application.ActiveDocument
If (Not objDocument Is Nothing) Then
	' we have an active document, now check for view mode
	If (objDocument.CurrentViewMode <> spyViewAuthentic) Then
		If (Not objDocument.SwitchViewMode (spyViewAuthentic)) Then
			MsgBox "Active document does not support authentic view mode"
		Else
			' now it is safe to access the authentic view object
			Dim objAuthenticView
			Set objAuthenticView = objDocument.AuthenticView
			' now use the authentic view object

		End If
	End If
Else
	MsgBox "No document is open"
End If
```

29.3.2.6.1 Events

29.3.2.6.1.1 OnBeforeCopy

Event: OnBeforeCopy() as Boolean

Scripting environment - VBScript:
```
Function On_AuthenticBeforeCopy()
	' On_AuthenticBeforeCopy = False  ' to disable operation
End Function
```

Scripting environment - JScript:
```
function On_AuthenticBeforeCopy()
{
	// return false;  /* to disable operation */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent (21, ...) // nEventId = 21

Description
This event gets triggered before a copy operation gets performed on the document. Return *True* (or nothing) to allow copy operation. Return *False* to disable copying.

29.3.2.6.1.2 OnBeforeCut

Event: OnBeforeCut() as Boolean

Scripting environment - VBScript:
```
Function On_AuthenticBeforeCut()
	' On_AuthenticBeforeCut = False  ' to disable operation
End Function
```

Scripting environment - JScript:
```
function On_AuthenticBeforeCut()
{
	// return false;  /* to disable operation */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent (20, ...) // nEventId = 20

Description
This event gets triggered before a cut operation gets performed on the document. Return *True* (or nothing) to allow cut operation. Return *False* to disable operation.

© 2016-2022 Altova GmbH

29.3.2.6.1.3 OnBeforeDelete

Event: OnBeforeDelete() as Boolean

Scripting environment - VBScript:
```
Function On_AuthenticBeforeDelete()
	' On_AuthenticBeforeDelete = False  ' to disable operation
End Function
```

Scripting environment - JScript:
```
function On_AuthenticBeforeDelete()
{
	// return false;  /* to disable operation */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent (22, ...) // nEventId = 22

Description
This event gets triggered before a delete operation gets performed on the document. Return *True* (or nothing) to allow delete operation. Return *False* to disable operation.

29.3.2.6.1.4 OnBeforeDrop

Event: OnBeforeDrop (*i_nXPos* as Long, *i_nYPos* as Long, *i_ipRange* as AuthenticRange[1750], *i_ipData* as cancelBoolean

Scripting environment - VBScript:
```
Function On_AuthenticBeforeDrop(nXPos, nYPos, objRange, objData)
	' On_AuthenticBeforeDrop = False  ' to disable operation
End Function
```

Scripting environment - JScript:
```
function On_AuthenticBeforeDrop(nXPos, nYPos, objRange, objData)
{
	// return false;  /* to disable operation */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent (11, ...) // nEventId = 11

Description
This event gets triggered whenever a previously dragged object gets dropped inside the application window. All event related information gets passed as parameters.

The first two parameters specify the mouse position at the time when the event occurred. The parameter *objRange* passes a range object that selects the XML element below the mouse position. The value of this parameter might be *NULL*. Be sure to check before you access the range object. The parameter *objData* allows to access information about the object being dragged.

Return *False* to cancel the drop operation. Return *True* (or nothing) to continue normal operation.

Examples

```
' --------------------------------------------------------------------
' VB code snippet - connecting to object level events
' --------------------------------------------------------------------
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnBeforeDrop
' event of object objView
Private Function objView_OnBeforeDrop(ByVal i_nXPos As Long, ByVal i_nYPos As Long,
                                      ByVal i_ipRange As IAuthenticRange,
                                      ByVal i_ipData As IAuthenticDataTransfer) As Boolean

   If (Not i_ipRange Is Nothing) Then
      MsgBox ("Dropping on content is prohibited");
      Return False;
   Else
      Return True;
   End If
End Function

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop
```

29.3.2.6.1.5 OnBeforePaste

Event: OnBeforePaste (*objData* as Variant, *strType* as String) as Boolean

Scripting environment - VBScript:

```
Function On_AuthenticBeforePaste(objData, strType)
        ' On_AuthenticBeforePaste = False  ' to disable operation
End Function
```

Scripting environment - JScript:

```
function On_AuthenticBeforePaste(objData, strType)
{
        // return false;  /* to disable operation */
}
```

IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (19, ...) // nEventId = 19
```

Description

 © 2016-2022 Altova GmbH

This event gets triggered before a paste operation gets performed on the document. The parameter *strType* is one of "TEXT", "UNICODETEXT" or "IUNKNOWN". In the first two cases *objData* contains a string representation of the object that will be pasted. In the later case, *objData* contains a pointer to an IUnknown COM interface.

Return *True* (or nothing) to allow paste operation. Return *False* to disable operation.

29.3.2.6.1.6 OnBeforeSave

Event: OnBeforeSave (SaveAs flag) as Boolean

Description: `OnBeforeSave` gives the opportunity to e.g. warn the user about overwriting the existing XML document, or to make the document read-only when specific circumstances are not met. The event will be fired before the file dialog is shown.

29.3.2.6.1.7 OnDragOver

Event: OnDragOver (*nXPos* as Long, *nYPos* as Long, *eMouseEvent* as SPYMouseEvent[1993], *objRange* as AuthenticRange[1750], *objData* as AuthenticDataTransfer) as Boolean

Scripting environment - VBScript:
```
Function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
        ' On_AuthenticDragOver = False  ' to disable operation
End Function
```

Scripting environment - JScript:
```
function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
{
        // return false;  /* to disable operation */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent (10, ...) // nEventId = 10

Description
This event gets triggered whenever an object from within or outside of vista Authentic gets dragged with the mouse over the application window. All event related information gets passed as parameters.

The first three parameters specify the mouse position, the mouse button status and the status of the virtual keys at the time when the event occurred. The parameter *objRange* passes a range object that selects the XML element below the mouse position. The value of this parameter might be *NULL*. Be sure to check before you access the range object. The parameter *objData* allows to access information about the object being dragged.

Return *False* to cancel the drag operation. Return *True* (or nothing) to continue normal operation.

Examples
```
' ----------------------------------------------------------------------
' VB code snippet - connecting to object level events
```

```
' ---------------------------------------------------------------------
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnDragOver
' event of object objView
Private Function objView_OnDragOver(ByVal i_nXPos As Long, ByVal i_nYPos As Long,
                                    ByVal i_eMouseEvent As SPYMouseEvent,
                                    ByVal i_ipRange As IAuthenticRange,
                                    ByVal i_ipData As IAuthenticDataTransfer) As Boolean

   If (((i_eMouseEvent And spyShiftKeyDownMask) <> 0) And
            (Not i_ipRange Is Nothing)) Then
      MsgBox ("Floating over element " & i_ipRange.FirstXMLData.Parent.Name);
   End If

   Return True;
End Function

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop
```

29.3.2.6.1.8 OnKeyboardEvent

Event: OnKeyboardEvent (*eKeyEvent* as SPYKeyEvent[1992], *nKeyCode* as Long, *nVirtualKeyStatus* as Long) as Boolean

Scripting environment - VBScript:

```
Function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
        ' On_AuthenticKeyboardEvent = True  ' to cancel bubbling of event
End Function
```

Scripting environment - JScript:

```
function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
{
        // return true;  /* to cancel bubbling of event */
}
```

IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (30, ...) // nEventId = 30
```

Description

This event gets triggered for *WM_KEYDOWN*, *WM_KEYUP* and *WM_CHAR* Windows messages.

© 2016-2022 Altova GmbH

The actual message type is available in the *eKeyEvent* parameter. The status of virtual keys is combined in the parameter *nVirtualKeyStatus*. Use the bit-masks defined in the enumeration datatype SPYVirtualKeyMask[2000], to test for the different keys or their combinations.

29.3.2.6.1.9 OnLoad

Event: OnLoad ()

Description: `OnLoad` can be used e.g. to restrict some AuthenticView functionality, as shown in the example below:

```
function On_AuthenticLoad(  )
{
    // We are disabling all entry helpers in order to prevent user from manipulating XML tree
    AuthenticView.DisableElementEntryHelper();
    AuthenticView.DisableAttributeEntryHelper();

    // We are also disabling the markup buttons for the same purpose
    AuthenticView.SetToolbarButtonState( 'AuthenticMarkupSmall', authenticToolbarButtonDisabled );
    AuthenticView.SetToolbarButtonState( 'AuthenticMarkupLarge', authenticToolbarButtonDisabled );
    AuthenticView.SetToolbarButtonState( 'AuthenticMarkupMixed', authenticToolbarButtonDisabled );
}
```

In the example the status of the Markup Small, Markup Large, Markup Mixed toolbar buttons are manipulated with the help of button identifiers. See complete list[1787].

29.3.2.6.1.10 OnMouseEvent

Event: OnMouseEvent (*nXPos* as Long, *nYPos* as Long, *eMouseEvent* as SPYMouseEvent[1993], *objRange* as AuthenticRange[1750]) as Boolean

Scripting environment - VBScript:

```
Function On_AuthenticMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
        ' On_AuthenticMouseEvent = True  ' to cancel bubbling of event
End Function
```

Scripting environment - JScript:

```
function On_AuthenticMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
{
        // return true;  /* to cancel bubbling of event */
}
```

IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (31, ...) // nEventId = 31
```

Description

This event gets triggered for every mouse movement and mouse button Windows message.

The actual message type and the mouse buttons status, is available in the *eMouseEvent* parameter. Use the bit-masks defined in the enumeration datatype SPYMouseEvent[1993] to test for the different messages, button status, and their combinations.

The parameter *objRange* identifies the part of the document found at the current mouse cursor position. The range object always selects a complete tag of the document. (This might change in future versions, when a more precise positioning mechanism becomes available). If no selectable part of the document is found at the current position, the range object is *null*.

29.3.2.6.1.11 OnSelectionChanged

Event: OnSelectionChanged (*objNewSelection* as AuthenticRange[1750])

Scripting environment - VBScript:
Function On_AuthenticSelectionChanged (*objNewSelection*)
End Function

Scripting environment - JScript:
function On_AuthenticSelectionChanged (*objNewSelection*)
{
}

IDE Plugin:
IXMLSpyPlugIn.OnEvent (23, ...) // nEventId = 23

Description
This event gets triggered whenever the selection in the user interface changes.

Examples
```
' ---------------------------------------------------------------------
' VB code snippet - connecting to object level events
' ---------------------------------------------------------------------
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnSelectionChanged
' event of object objView
Private Sub objView_OnSelectionChanged (ByVal i_ipNewRange As XMLSpyLib.IAuthenticRange)
    MsgBox ("new selection: " & i_ipNewRange.Text)
End Sub

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop
```


© 2016-2022 Altova GmbH

29.3.2.6.1.12 OnToolbarButtonClicked

Event: OnToolbarButtonClicked (Button identifier)

Description: `OnToolbarButtonClicked` is fired when a toolbar button was clicked by user. The parameter button identifier helps to determine which button was clicked. The list of predefined button identifiers is below:

- AuthenticPrint
- AuthenticPrintPreview
- AuthenticUndo
- AuthenticRedo
- AuthenticCut
- AuthenticCopy
- AuthenticPaste
- AuthenticClear
- AuthenticMarkupHide
- AuthenticMarkupLarge
- AuthenticMarkupMixed
- AuthenticMarkupSmall
- AuthenticValidate
- AuthenticChangeWorkingDBXMLCell
- AuthenticSave
- AuthenticSaveAs
- AuthenticReload
- AuthenticTableInsertRow
- AuthenticTableAppendRow
- AuthenticTableDeleteRow
- AuthenticTableInsertCol
- AuthenticTableAppendCol
- AuthenticTableDeleteCol
- AuthenticTableJoinCellRight
- AuthenticTableJoinCellLeft
- AuthenticTableJoinCellAbove
- AuthenticTableJoinCellBelow
- AuthenticTableSplitCellHorizontally
- AuthenticTableSplitCellVertically
- AuthenticTableAlignCellContentTop
- AuthenticTableCenterCellVertically
- AuthenticTableAlignCellContentBottom
- AuthenticTableAlignCellContentLeft
- AuthenticTableCenterCellContent
- AuthenticTableAlignCellContentRight
- AuthenticTableJustifyCellContent
- AuthenticTableInsertTable
- AuthenticTableDeleteTable
- AuthenticTableProperties
- AuthenticAppendRow
- AuthenticInsertRow
- AuthenticDuplicateRow
- AuthenticMoveRowUp
- AuthenticMoveRowDown

- AuthenticDeleteRow
- AuthenticDefineEntities
- AuthenticXMLSignature

For custom buttons the user might add his own identifiers. Please, note that the user must take care, as the identifiers are not checked for uniqueness. The same identifiers can be used to identify buttons in the `Set/GetToolbarState()` COM API calls. By adding code for different buttons, the user is in the position to completely redefine the AuthenticView toolbar behavior, adding own methods for table manipulation, etc.

29.3.2.6.1.13 OnToolbarButtonExecuted

Event: OnToolbarButtonExecuted (Button identifier)

Description: `OnToolbarButtonClicked` is fired when a toolbar button was clicked by user. The parameter button identifier helps to determine which button was clicked. See the list of predefined button identifiers 1787.

`OnToolbarButtonExecuted` is fired after the toolbar action was executed. It is useful e.g. to add update code, as shown in the example below:

```
//event fired when a toolbar button action was executed
function On_AuthenticToolbarButtonExecuted( varBtnIdentifier )
{
    // After whatever command user has executed - make sure to update toolbar button states
    UpdateOwnToolbarButtonStates();
}
```

In this case `UpdateOwnToolbarButtonStates` is a user function defined in the Global Declarations.

29.3.2.6.1.14 OnUserAddedXMLNode

Event: OnUserAddedXMLNode (XML node)

Description: `OnUserAddedXMLNode` will be fired when the user adds an XML node as a primary action. This happens in the situations, where the user clicks on

- auto-add hyperlinks (see example OnUserAddedXMLNode.sps)
- the Insert…, Insert After…, Insert Before… context menu items
- Append row, Insert row toolbar buttons
- Insert After…, Insert Before… actions in element entry helper (outside StyleVision)

The event doesn't get fired on Duplicate row, or when the node was added externally (e.g. via COM API), or on Apply (e.g. Text State Icons), or when in XML table operations or in DB operations.

The event parameter is the XML node object, which was added giving the user an opportunity to manipulate the XML node added. An elaborate example for an event handler can be found in the `OnUserAddedXMLNode.sps` file.

© 2016-2022 Altova GmbH

29.3.2.6.2 Application

Property: Application as Application[1724] (read-only)

Description
Accesses the XMLSpy application object.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.3 AsXMLString

Property: AsXMLString as String

Description
Returns or sets the document content as an XML string. Setting the content to a new value does not change the schema file or sps file in use. If the new XMLString does not match the actual schema file error 2011 gets returned.

Errors
2000 The authentic view object is no longer valid.
2011 AsXMLString was set to a value which is no valid XML for the current schema file.

29.3.2.6.4 ContextMenu

Property: ContextMenu() as ContextMenu

Description
The property `ContextMenu` gives access to customize the context menu. The best place to do it is in the event handler `OnContextMenuActivated`.

Errors
2000 Invalid object.
2005 Invalid parameter.

29.3.2.6.5 CreateXMLNode

Method: CreateXMLNode (*nKind* as SPYXMLDataKind[2000]) as XMLData[1939]

Return Value
The method returns the new XMLData[1939] object.

Description
To create a new XMLData object use the CreateXMLNode() method.

Errors
2000 Invalid object.
2012 Cannot create XML node.

29.3.2.6.6 DisableAttributeEntryHelper

Method: DisableAttributeEntryHelper()

Description
DisableAttributeEntryHelper() disables the attribute entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

Errors
2000 Invalid object.

29.3.2.6.7 DisableElementEntryHelper

Method: DisableElementEntryHelper()

Description
DisableElementEntryHelper() disables the element entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

Errors
2000 Invalid object.

29.3.2.6.8 DisableEntityEntryHelper

Method: DisableEntityEntryHelper()

Description
DisableEntityEntryHelper() disables the entity entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

Errors
2000 Invalid object.

29.3.2.6.9 DocumentBegin

Property: DocumentBegin as AuthenticRange 1750 (read-only)

© 2016-2022 Altova GmbH

Description
Retrieve a range object that points to the beginning of the document.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.10 DocumentEnd

Property: DocumentEnd as AuthenticRange[1750] (read-only)

Description
Retrieve a range object that points to the end of the document.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.11 DoNotPerformStandardAction

Method: DoNotPerformStandardAction ()

Description
`DoNotPerformStandardAction()` serves as cancel bubble for macros, and stops further execution after macro has finished.

Errors
2000 Invalid object.

29.3.2.6.12 EvaluateXPath

Method: EvaluateXPath (XMLData as XMLData[1939], strExpression as string) strValue as string

Return Value
The method returns a string

Description
`EvaluateXPath()` executes an XPath expression with the given XML context node. The result is returned as a string, in the case of a sequence it is a space-separated string.

Errors
2000 Invalid object.
2005 Invalid parameter.
2008 Internal error.
2013 XPath error.

29.3.2.6.13 Event

Property: Event as AuthenticEvent (read-only)

Description
This property gives access to parameters of the last event in the same way as OldAuthenticView.event does. Since all events for the scripting environment and external clients are now available with parameters this Event property should only be used from within IDE-Plugins.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.14 EventContext

Property: EventContext() as EventContext

Description
`EventContext` property gives access to the running macros context. See the `EventContext` [1747] interface description for more details.

Errors
2000 Invalid object.

29.3.2.6.15 GetToolbarButtonState

Method: GetToolbarButtonState (`ButtonIdentifier` as string) as `AuthenticToolbarButtonState`

Return Value
The method returns `AuthenticToolbarButtonState`

Description
`Get/SetToolbarButtonState` queries the status of a toolbar button, and lets the user disable or enable the button, identified via its button identifier (see list above [1787]). One usage is to disable toolbar buttons permanently. Another usage is to put `SetToolbarButtonState` in the `OnSelectionChanged` event handler, as toolbar buttons are updated regularly when the selection changes in the document.

Toolbar button states are given by the listed enumerations [1990].

The default state means that the enable/disable of the button is governed by AuthenticView. When the user sets the button state to enable or disable, the button remains in that state as long as the user does not change it.

Errors
2000 Invalid object.

© 2016-2022 Altova GmbH

2005 Invalid parameter.
2008 Internal error.
2014 Invalid button identifier.

29.3.2.6.16 Goto

Method: Goto (*eKind* as SPYAuthenticElementKind[1989], *nCount* as Long, *eFrom* as *SPYAuthenticDocumentPosition*[1988]) as *AuthenticRange*[1750]

Description
Retrieve a range object that points to the beginning of the *nCount* element of type *eKind*. The start position is defined by the parameter *eFrom*. Use positive values for *nCount* to navigate to the document end. Use negative values to navigate towards the beginning of the document.

Errors
2000 The authentic view object is no longer valid.
2003 Target lies after end of document.
2004 Target lies before beginning of document.
2005 Invalid element kind specified.
The document position to start from is not one of *spyAuthenticDocumentBegin* or *spyAuthenticDocumentEnd*.
Invalid address for the return parameter was specified.

Examples

```
' ---------------------------------------
' Scripting environment - VBScript
' ---------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

On Error Resume Next
Dim objRange
' goto beginning of first table in document
Set objRange = objAuthenticView.Goto (spyAuthenticTable, 1, spyAuthenticDocumentBegin)
If (Err.number = 0) Then
        objRange.Select()
Else
        MsgBox "No table found in document"
End If
```

29.3.2.6.17 IsRedoEnabled

Property: IsRedoEnabled as Boolean (read-only)

Description
True if redo steps are available and Redo[1795] is possible.

Errors

2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.18 IsUndoEnabled

Property: IsUndoEnabled as Boolean (read-only)

Description
True if undo steps are available and Undo[1796] is possible.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.19 MarkupVisibility

Property: MarkupVisibility as SPYAuthenticMarkupVisibility[1989]

Description
Set or get current visibility of markup.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid enumeration value was specified.
Invalid address for the return parameter was specified.

29.3.2.6.20 Parent

Property: Parent as Document[1813] (read-only)

Description
Access the document shown in this view.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.21 Print

Method: Print (*bWithPreview* as Boolean, *bPromptUser* as Boolean)

Description

© 2016-2022 Altova GmbH

Print the document shown in this view. If *bWithPreview* is set to *True*, the print preview dialog pops up. If *bPromptUser* is set to *True*, the print dialog pops up. If both parameters are set to *False*, the document gets printed without further user interaction.

Errors
2000 The authentic view object is no longer valid.

29.3.2.6.22 Redo

Method: Redo() as Boolean

Description
Redo the modification undone by the last undo command.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.23 Selection

Property: Selection as AuthenticRange 1750

Description
Set or get current text selection in user interface.

Errors
2000 The authentic view object is no longer valid.
2002 No cursor selection is active.
2005 Invalid address for the return parameter was specified.

Examples

```
' ---------------------------------------
' Scripting environment - VBScript
' ---------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

' if we are the end of the document, re-start at the beginning
If (objAuthenticView.Selection.IsEqual(objAuthenticView.DocumentEnd)) Then
        objAuthenticView.Selection = objAuthenticView.DocumentBegin
Else
        ' objAuthenticView.Selection = objAuthenticView.Selection.GotoNextCursorPosition()
        ' or shorter:
         objAuthenticView.Selection.GotoNextCursorPosition().Select
End If
```

29.3.2.6.24 SetToolbarButtonState

Method: SetToolbarButtonState (`ButtonIdentifier` as string, `AuthenticToolbarButtonState` state)

Description
`Get/SetToolbarButtonState` queries the status of a toolbar button, and lets the user disable or enable the button, identified via its button identifier (see list above 1787). One usage is to disable toolbar buttons permanently. Another usage is to put `SetToolbarButtonState` in the `OnSelectionChanged` event handler, as toolbar buttons are updated regularly when the selection changes in the document.

Toolbar button states are given by the listed enumerations 1990.

The default state means that the enable/disable of the button is governed by AuthenticView. When the user sets the button state to enable or disable, the button remains in that state as long as the user does not change it.

Errors
2000 Invalid object.
2008 Internal error.
2014 Invalid button identifier.

29.3.2.6.25 Undo

Method: Undo() as Boolean

Description
Undo the last modification of the document from within this view.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.26 UpdateXMLInstanceEntities

Method: UpdateXMLInstanceEntities()

Description
Updates the internal representation of the declared entities, and refills the entry helper. In addition, the validator is reloaded, allowing the XML file to validate correctly. Please note that this may also cause schema files to be reloaded.

Errors
The method never returns an error.

Example
// ----------

© 2016-2022 Altova GmbH

```
// Scripting environment - JavaScript
// ----------------------------------------
if(Application.ActiveDocument && (Application.ActiveDocument.CurrentViewMode == 4))
{
	var objDocType;
	objDocType = Application.ActiveDocument.DocEditView.XMLRoot.GetFirstChild(10);

	if(objDocType)
	{
		var objEntity = Application.ActiveDocument.CreateChild(14);
		objEntity.Name = "child";
		objEntity.TextValue = "SYSTEM \"child.xml\"";
		objDocType.AppendChild(objEntity);

		Application.ActiveDocument.AuthenticView.UpdateXMLInstanceEntities();
	}
}
```

29.3.2.6.27 WholeDocument

Property: WholeDocument as AuthenticRange[1750] (read-only)

Description
Retrieve a range object that selects the whole document.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.28 XMLDataRoot

Property: XMLDataRoot as XMLData[1939] (read-only)

Description
Returns or sets the top-level XMLData element of the current document. This element typically describes the document structure and would be of kind spyXMLDataXMLDocStruct, spyXMLDataXMLEntityDocStruct or spyXMLDataDTDDocStruct..

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.7 CodeGeneratorDlg

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Properties and Methods

Standard automation properties
Application[1798]
Parent[1802]

Programming language selection properties
ProgrammingLanguage[1802]
TemplateFileName[1803]

Settings for C++ code
CPPSettings_DOMType[1799]
CPPSettings_LibraryType[1800]
CPPSettings_UseMFC[1800]
CPPSettings_GenerateVC6ProjectFile[1799]
CPPSettings_GenerateVSProjectFile[1800]

Settings for C# code
CSharpSettings_ProjectType[1801]

Dialog handling for above code generation properties
PropertySheetDialogAction[1803]

Output path selection properties
OutputPath[1801]
OutputPathDialogAction[1801]

Presentation of result
OutputResultDialogAction[1802]

Description
Use this object to configure the generation of program code for schema files. The method GenerateProgramCode[1829] expects a CodeGeneratorDlg as parameter to configure code generation as well as the associated user interactions.

29.3.2.7.1 Application

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Property: Application as Application[1724] (read-only)

Description
Access the XMLSpy application object.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.7.2 CompatibilityMode (obsolete)

Property: CompatibilityMode as Boolean
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Set to true to generate code compatible to XMLSpy 2005R3. Set to false to use newly added code-generation features.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.3 CPPSettings_DOMType

Property: CPPSettings_DOMType as SPYDOMType 1990
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines one of the settings that configure generation of C++ code.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.4 CPPSettings_GenerateVC6ProjectFile

Property: CPPSettings_GenerateVC6ProjectFile as Boolean
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines one of the settings that configure generation of C++ code.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.5 CPPSettings_GenerateGCCMakefile

Property: CPPSettings_GenerateGCCMakefile as Boolean
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Creates makefiles to compile the generated code under Linux with GCC.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.6 CPPSettings_GenerateVSProjectFile

Property: CSharpSettings_GenerateVSProjectFile as SPYProjectType[1995]
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines one of the settings that configure generation of C++ code. Only `spyVisualStudio2005Project (=4)` and `spyVisualStudio2008Project (=5)` and `spyVisualStudio2010Project (=6)` are valid project types.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.7 CPPSettings_LibraryType

Property: CPPSettings_LibraryType as SPYLibType[1993]
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines one of the settings that configure generation of C++ code.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.8 CPPSettings_UseMFC

Property: CPPSettings_UseMFC as Boolean
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

© 2016-2022 Altova GmbH

Description
Defines one of the settings that configure generation of C++ code.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.9 CSharpSettings_ProjectType

Property: CSharpSettings_ProjectType as SPYProjectType[1995]
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines the only setting to configure generation of C# code.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.10 OutputPath

Property: OutputPath as String
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Selects the base directory for all generated code.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.7.11 OutputPathDialogAction

Property: OutputPathDialogAction as SPYDialogAction[1990]
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines how the sub-dialog for selecting the code generation output path gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current value of the OutputPath[1801] property as default. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors
2200 The object is no longer valid.

2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.12 OutputResultDialogAction

Property: OutputResultDialogAction as SPYDialogAction[1990]
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines how the sub-dialog that asks to show the result of the code generation process gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.13 Parent

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Property: Parent as Dialogs[1809] (read-only)

Description
Access the parent of the object.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.7.14 ProgrammingLanguage

Property: ProgrammingLanguage as ProgrammingLanguage[1994]
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Selects the output language for the code to be generated.

CAUTION: Setting this property to one of C++, C# or Java, changes the property TemplateFileName[1803] to the appropriate template file delivered with XMLSpy as well. If you want to generate C++, C# or Java code based on your own templates, set first the programming language and then select your template file.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.7.15 PropertySheetDialogAction

Property: PropertySheetDialogAction as SPYDialogAction[1990]
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines how the sub-dialog that configures the code generation process gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current values as defaults. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors

2200	The object is no longer valid.
2201	Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.16 TemplateFileName

Property: TemplateFileName as String
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Selects the code generation template file. XMLSpy comes with template files for C++, C# or Java in the SPL folder of your installation directory.

Setting this property to one of the code generation template files of your XMLSpy installation automatically sets the ProgrammingLanguage[1802] property to its appropriate value.

Errors

2200	The object is no longer valid.
2201	Invalid address for the return parameter was specified.

29.3.2.8 DatabaseConnection

Properties for import and export
File[1806] or
ADOConnection[1804] or
ODBCConnection[1808]

Properties for import only
DatabaseKind[1805]
SQLSelect[1809]
AsAttributes[1804]
ExcludeKeys[1806]
IncludeEmptyElements[1807]
NumberDateTimeFormat[1807]
NullReplacement[1807]
CommentIncluded[1805]

Properties for export only
CreateMissingTables 1805
CreateNew 1805
TextFieldLen 1809
DatabaseSchema 1806

Properties for XML Schema from DB Structure generation
PrimaryKeys 1808
ForeignKeys 1806
UniqueKeys 1809
SchemaExtensionType 1808
SchemaFormat 1808
ImportColumnsType 1807

Description
DatabaseConnection specifies the parameters for the database connection.

Please note that the properties of the DatabaseConnection interface are referring to the settings of the import and export dialogs of XMLSpy.

29.3.2.8.1 ADOConnection

Property: ADOConnection as String

Description
The property ADOConnection contains a connection string. Either use this property or ODBCConnection 1808 or File 1806 to refer to a database.

Errors
No error codes are returned.

Example

```
Dim objSpyConn As DatabaseConnection
Set objSpyConn = objSpy.GetDatabaseSettings

Dim objADO As DataLinks
Set objADO = CreateObject("DataLinks")

If Not (objADO Is Nothing) Then
        Dim objConn As Connection
        Set objConn = objADO.PromptNew
        objSpyConn.ADOConnection = objConn.ConnectionString
End If
```

29.3.2.8.2 AsAttributes

Property: AsAttributes as Boolean

© 2016-2022 Altova GmbH

Description
Set AsAttributes to true if you want to initialize all import fields to be imported as attributes. Default is false and will initialize all fields to be imported as elements. This property is used only in calls to Application.GetDatabaseImportElementList[1731].

Errors
No error codes are returned.

29.3.2.8.3 CommentIncluded

Property: CommentIncluded as Boolean

Description
This property tells whether additional comments are added to the generated XML. Default is true. This property is used only when importing from databases.

Errors
No error codes are returned.

29.3.2.8.4 CreateMissingTables

Property: CreateMissingTables as Boolean

Description
If CreateMissingTables is true, tables which are not already defined in the export database will be created during export. Default is true. This property is used only when exporting to databases.

Errors
No error codes are returned.

29.3.2.8.5 CreateNew

Property: CreateNew as Boolean

Description
Set CreateNew true if you want to create a new database on export. Any existing database will be overwritten. See also DatabaseConnection.File[1806]. Default is false. This property is used only when exporting to databases.

Errors
No error codes are returned.

29.3.2.8.6 DatabaseKind

Property: DatabaseKind as SPYDatabaseKind[1990]

Description

Select the kind of database that gets access. The default value is spyDB_Unspecified(7) and is sufficient in most cases. This property is used only when importing from databases.

Errors
No error codes are returned.

29.3.2.8.7 DatabaseSchema

Property: DatabaseSchema as String

Description
This property specifies the Schema used for export in Schema aware databases. Default is "". This property is used only when exporting to databases.

Errors
No error codes are returned.

29.3.2.8.8 ExcludeKeys

Property: ExcludeKeys as Boolean

Description
Set ExcludeKeys to true if you want to exclude all key columns from the import data. Default is false. This property is used only when importing from databases.

Errors
No error codes are returned.

29.3.2.8.9 File

Property: File as String

Description
The property File sets the path for the database during export or import. This property can only be used in conjunction with a Microsoft Access database. Either use this property or ODBCConnection[1808] or ADOConnection[1804] to refer to the database.

See also Import and Export[1694].

Errors
No error codes are returned.

29.3.2.8.10 ForeignKeys

Property: ForeignKeys as Boolean

© 2016-2022 Altova GmbH

Description
Specifies whether the Foreign Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

Errors
No error codes are returned.

29.3.2.8.11 ImportColumnsType

Property: ImportColumnsType as SPYImportColumnsType[1992]

Description
Defines if column information from the DB is saved as element or attribute in the XML Schema. Default is as element. This property is used only when creating a XML Schema from a DB structure.

Errors
No error codes are returned.

29.3.2.8.12 IncludeEmptyElements

Property: IncludeEmptyElements as Boolean

Description
Set IncludeEmptyElements to false if you want to exclude all empty elements. Default is true. This property is used only when importing from databases.

Errors
No error codes are returned.

29.3.2.8.13 NullReplacement

Property: NullReplacement as String

Description
This property contains the text value that is used during import for empty elements (null values). Default is "". This property is used only when importing from databases.

Errors
No error codes are returned.

29.3.2.8.14 NumberDateTimeFormat

Property: NumberDateTimeFormat as SPYNumberDateTimeFormat[1994]

Description
The property NumberDateTimeFormat sets the format of numbers and date- and time-values. Default is spySystemLocale[1994]. This property is used only when importing from databases.

Errors
No error codes are returned.

29.3.2.8.15 ODBCConnection

Property: ODBCConnection as String

Description
The property ODBCConnection contains a ODBC connection string. Either use this property or ADOConnection[1804] or File[1806] to refer to a database.

Errors
No error codes are returned.

29.3.2.8.16 PrimaryKeys

Property: PrimaryKeys as Boolean

Description
Specifies whether the Primary Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

Errors
No error codes are returned.

29.3.2.8.17 SchemaExtensionType

Property: SchemaExtensionType as SPYSchemaExtensionType[1997]

Description
Defines the Schema extension type used during the Schema generation. This property is used only when creating a XML Schema from a DB structure.

See also Create XML Schema from DB Structure[1461].

Errors
No error codes are returned.

29.3.2.8.18 SchemaFormat

Property: SchemaFormat as SPYSchemaFormat[1997]

Description
Defines the Schema format used during the Schema generation. This property is used only when creating a XML Schema from a DB structure.

© 2016-2022 Altova GmbH

See also Create XML Schema from DB Structure[1461].

Errors
No error codes are returned.

29.3.2.8.19 SQLSelect

Property: SQLSelect as String

Description
The SQL query for the import is stored in the property SQLSelect. This property is used only when importing from databases. See also Import and Export[1694].

Errors
No error codes are returned.

29.3.2.8.20 TextFieldLen

Property: TextFieldLen as long

Description
The property TextFieldLen sets the length for created text fields during the export. Default is 255. This property is used only when exporting to databases.

Errors
No error codes are returned.

29.3.2.8.21 UniqueKeys

Property: UniqueKeys as Boolean

Description
Specifies whether the Unique Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

Errors
No error codes are returned.

29.3.2.9 Dialogs

Properties and Methods

Standard automation properties
Application[1810]
Parent[1811]

Various dialog objects

CodeGeneratorDlg 1810
FileSelectionDlg 1810
SchemaDocumentationDlg 1811
GenerateSampleXMLDlg 1811
DTDSchemaGeneratorDlg 1812
FindInFilesDlg 1812
WSDLDocumentationDlg 1812
WSDL20DocumentationDlg 1812
XBRLDocumentationDlg 1813

Description
The Dialogs object provides access to different built-in dialogs of XMLSpy. These dialog objects allow to initialize the fields of user dialogs before they get presented to the user or allow to simulate complete user input by your program.

29.3.2.9.1 Application

Property: Application as Application 1724 (read-only)

Description
Access the XMLSpy application object.

Errors
2300 The object is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.2 CodeGeneratorDlg

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Property: CodeGeneratorDlg as CodeGeneratorDlg 1797 (read-only)

Description
Get a new instance of a code generation dialog object. You will need this object to pass the necessary parameters to the code generation methods. Initial values are taken from last usage of the code generation dialog.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.3 FileSelectionDlg

Property: FileSelectionDlg as FileSelectionDlg 1856 (read-only)

Description
Get a new instance of a file selection dialog object.

 © 2016-2022 Altova GmbH

File selection dialog objects are passed to you with the some events that signal opening or saving of documents and projects.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.4 Parent

Property: Parent as Application[1724] (read-only)

Description
Access the XMLSpy application object.

Errors
2300 The object is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.5 SchemaDocumentationDlg

Property: SchemaDocumentationDlg as SchemaDocumentationDlg[1878] (read-only)

Description
Get a new instance of a dialog object that parameterizes generation of schema documentation. See Document.GenerateSchemaDocumentation[1829] for its usage.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.6 GenerateSampleXMLDlg

Property: GenerateSampleXMLDlg as GenerateSampleXMLDlg[1869] (read-only)

Description
Get a new instance of a dialog object that parameterizes generation of a sample XML based on a W3C schema or DTD. See GenerateSampleXML[1829] for its usage.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.7 DTDSchemaGeneratorDlg

Property: DTDSchemaGeneratorDlg as DTDSchemaGeneratorDlg[1847] (read-only)

Description
Get a new instance of a dialog object that parameterizes generation of a schema or DTD. See Document.GenerateDTDOrSchemaEx[1828] for its usage.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.8 FindInFilesDlg

Property: FindInFilesDlg as FindInFilesDlg[1857] (read-only)

Description
Get a new instance of a dialog object that parameterizes the search (or replacement) of strings in files. See Application.FindInFiles[1730] for its usage.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.9 WSDLDocumentationDlg

Property: WSDLDocumentationDlg as WSDLDocumentationDlg[1905] (read-only)

Description
Get a new instance of a dialog object that parameterizes generation of WSDL documentation. See Document.GenerateWSDLDocumentation[1830] for its usage.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.10 WSDL20DocumentationDlg

Property: WSDL20DocumentationDlg as WSDL20DocumentationDlg[1917] (read-only)

Description
Get a new instance of a dialog object that parameterizes generation of WSDL 2.0 documentation. See Document.GenerateWSDL20Documentation[1830] for its usage.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.9.11 XBRLDocumentationDlg

Property: XBRLDocumentationDlg as XBRLDocumentationDlg[1928] (read-only)

Description
Get a new instance of a dialog object that parameterizes generation of XBRL documentation. See Document.GenerateXBRLDocumentation[1830] for its usage.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.10 Document

Properties and Methods

Standard automation properties
Application[1818]
Parent[1836]

Various document properties and methods
SetActiveDocument[1838]
Encoding[1824]
SetEncoding (obsolete)[1838]
Suggestions[1840]

XML validation
IsValid[1833]
SetExternalIsValid[1839]

Document conversion and transformation
AssignDTD[1818]
AssignSchema[1818]
AssignXSL[1818]
AssignXSLFO[1819]
ConvertDTDOrSchema[1820]
ConvertDTDOrSchemaEx[1821]
GenerateDTDOrSchema[1828]
GenerateDTDOrSchemaEx[1828]
FlattenDTDOrSchema[1827]
CreateSchemaDiagram[1823]
ExecuteXQuery[1825]
TransformXSL[1841]
TransformXSLEx[1841]
TransformXSLFO[1842]
GenerateProgramCode[1829] (Enterprise Edition only)
GenerateSchemaDocumentation[1829]

GenerateSampleXML 1829
GenerateWSDL20Documentation 1830
GenerateWSDLDocumentation 1830
GenerateXBRLDocumentation 1831
ConvertToWSDL20 1821

Document export
GetExportElementList 1831
ExportToText 1826
ExportToDatabase 1825
CreateDBStructureFromXMLSchema 1822
GetDBStructureList 1831

File saving and naming
FullName 1828
Name 1835
Path 1836
GetPathName (obsolete) 1832
SetPathName (obsolete) 1839
Title 1841
IsModified 1833
Saved 1837
SaveAs 1837
Save 1837
SaveInString 1837
SaveToURL 1838
Close 1820

View access
CurrentViewMode 1823
SwitchViewMode 1840
AuthenticView 1819
GridView 1832
DocEditView (obsolete) 1824

Access to XMLData
RootElement 1836
DataRoot 1823
CreateChild 1822
UpdateViews 1842
StartChanges 1840
EndChanges 1824
UpdateXMLData 1842

Description
Document objects represent XML documents opened in XMLSpy.

Use one of the following properties to access documents that are already open XMLSpy:
Application.ActiveDocument 1728
Application.Documents 1730

Use one of the following methods to open a new document in XMLSpy:
Documents.OpenFile 1846

© 2016-2022 Altova GmbH

Documents.OpenURL 1846
Documents.OpenURLDialog 1847
Documents.NewFile 1845
Documents.NewFileFromText 1845
SpyProjectItem.Open 1894
Application.ImportFromDatabase 1735
Application.ImportFromSchema 1736
Application.ImportFromText 1736
Application.ImportFromWord 1737
Document.ConvertDTDOrSchema 1820
Document.GenerateDTDOrSchema 1828

29.3.2.10.1 Events

29.3.2.10.1.1 OnBeforeSaveDocument

Event: OnBeforeSaveDocument(*objDocument* as Document 1813, *objDialog* as FileSelectionDlg 1856)

XMLSpy scripting environment - VBScript:
Function On_BeforeSaveDocument(*objDocument*, *objDialog*)
End Function

' old handler - now obsolete
' return string to save to new file name
' return empty string to cancel save operation
' return nothing to save to original name
Function On_SaveDocument(*objDocument*, *strFilePath*)
End Function

XMLSpy scripting environment - JScript:
function On_BeforeSaveDocument(*objDocument*, *objDialog*)
{
}

// old handler - now obsolete
// return string to save to new file name
// return empty string to cancel save operation
// return nothing to save to original name
function On_SaveDocument(*objDocument*, *strFilePath*)
{
}

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (27, ...) // nEventId = 27

Description
This event gets fired on any attempt to save a document. The file selection dialog object is initialized with the name chosen for the document file. You can modify this selection. To continue saving the document leave the

FileSelectionDlg.DialogAction[1857] property of *io_objDialog* at its default value spyDialogOK[1990]. To abort saving of the document set this property to spyDialogCancel[1990].

29.3.2.10.1.2 OnBeforeCloseDocument

Event: OnBeforeCloseDocument(*objDocument* as Document[1813])as Boolean

XMLSpy scripting environment - VBScript:
```
Function On_BeforeCloseDocument(objDocument)
	' On_BeforeCloseDocument = False  ' to prohibit closing of document
End Function
```

XMLSpy scripting environment - JScript:
```
function On_BeforeCloseDocument(objDocument)
{
	// return false;  /* to prohibit closing of document */
}
```

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (28, ...) // nEventId = 28

Description
This event gets fired on any attempt to close a document. To prevent the document from being closed return false.

29.3.2.10.1.3 OnBeforeValidate

Event: OnBeforeValidate(*objDocument* as Document[1813], *bOnLoading* as Boolean, *bOnCommand* as Boolean) as Boolean

XMLSpy scripting environment - VBScript:
```
Function On_BeforeValidate(objDocument, bOnLoading, bOnCommand)
	On_BeforeValidate = bCancelDefaultValidation     'set by the script if necessary
End Function
```

XMLSpy scripting environment - JScript:
```
function On_BeforeValidate(objDocument, bOnLoading, bOnCommand)
{
	return bCancelDefaultValidation   //set by the script if necessary
}
```

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (32, ...) // nEventId = 32

Description
This event gets fired before the document is validated. It is possible to suppress the default validation by returning false from the event handler. In this case the script should also set the validation result using the SetExternalIsValid[1839] method.

© 2016-2022 Altova GmbH

bOnLoading is true if the event is raised on the initial validation on loading the document.

bOnCommand is true whenever the user selected the Validate command from the Toolbar or menu.

Available with TypeLibrary version 1.5

29.3.2.10.1.4 OnCloseDocument

Event: OnCloseDocument(*objDocument* as Document[1813])

XMLSpy scripting environment - VBScript:
Function On_Close Document(*objDocument*)
End Function

XMLSpy scripting environment - JScript:
function On_Close Document(*objDocument*)
{
}

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (8, ...) // nEventId = 8

Description
This event gets fired as a result of closing a document. Do not modify the document from within this event.

29.3.2.10.1.5 OnViewActivation

Event: OnViewActivation(*objDocument* as Document[1813], *eViewMode* as SPYViewModes[1999], *bActivated* as Boolean)

XMLSpy scripting environment - VBScript:
Function On_ViewActivation(*objDocument*, *eViewMode*, *bActivated*)
End Function

XMLSpy scripting environment - JScript:
function On_ViewActivation(*objDocument*, *eViewMode*, *bActivated*)
{
}

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (29, ...) // nEventId = 29

Description
This event gets fired whenever a view of a document becomes visible (i.e. becomes the active view) or invisible (i.e. another view becomes the active view or the document gets closed). However, the first view activation event after a document gets opened cannot be received, since there is no document object to get the event from. Use the Application.OnDocumentOpened[1727] event instead.

29.3.2.10.2 Application

Property: Application as Application[1724] (read-only)

Description
Accesses the XMLSpy application object.

Errors
1400 The object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.3 AssignDTD

Method: AssignDTD(*strDTDFile* as String, *bDialog* as Boolean)

Description
The method places a reference to the DTD file "strDTDFile" into the document. Note that no error occurs if the file does not exist, or is not accessible. If bDialog is true XMLSpy presents a dialog to set the file.

Errors
1400 The object is no longer valid.
1409 You are not allowed to assign a DTD to the document.

29.3.2.10.4 AssignSchema

Method: AssignSchema (*strSchemaFile* as String, *bDialog* as Boolean)

Description
The method places a reference to the schema file "strSchemaFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If bDialog is true XMLSpy presents a dialog to set the file.

Errors
1400 The object is no longer valid.
1409 You are not allowed to assign a schema file to the document.

29.3.2.10.5 AssignXSL

Method: AssignXSL (*strXSLFile* as String, *bDialog* as Boolean)

Description
The method places a reference to the XSL file "strXSLFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If bDialog is true XMLSpy presents a dialog to set the file.

Errors
1400 The object is no longer valid.
1409 You are not allowed to assign an XSL file to the document.

© 2016-2022 Altova GmbH

29.3.2.10.6 AssignXSLFO

Method: AssignXSLFO (*strXSLFOFile* as String, *bDialog* as Boolean)

Description
The method places a reference to the XSLFO file "strXSLFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If bDialog is true XMLSpy presents a dialog to set the file.

Errors
1400 The object is no longer valid.
1409 You are not allowed to assign an XSL file to the document.

29.3.2.10.7 AsXMLString

Property: AsXMLString as String

Description
This property can be used to get or set the document content.

Errors
1400 The document object is no longer valid.
1404 Cannot create XMLData object.
1407 View mode cannot be switched.

29.3.2.10.8 AuthenticView

Method: AuthenticView as AuthenticView[1778] (read-only)

Description
Returns an object that gives access to properties and methods specific to Authentic view. The object returned is only valid if the current document is opened in Authentic view mode. The lifetime of an object ends with the next view switch. Any attempt to access objects or any of its children afterwards will result in an error indicating that the object is invalid.

AuthenticView[1778] and DocEditView[1965] both provide automation access to the Authentic view mode of XMLSpy. Functional overlap is intentional. A future version of Authentic View will include all functionality of DocEditView[1965] and its sub-objects, thereby making usage of DocEditView[1965] obsolete.

Errors
1400 The object is no longer valid.
1417 Document needs to be open in authentic view mode.

Examples

```
' ----------------------------------------
' XMLSpy scripting environment - VBScript
' secure access to authentic view object
' ----------------------------------------
Dim objDocument
```

```
Set objDocument = Application.ActiveDocument
If (Not objDocument Is Nothing) Then
	' we have an active document, now check for view mode
	If (objDocument.CurrentViewMode <> spyViewAuthentic) Then
		If (Not objDocument.SwitchViewMode (spyViewAuthentic)) Then
			MsgBox "Active document does not support authentic view mode"
		Else
			' now it is safe to access the authentic view object
			Dim objAuthenticView
			Set objAuthenticView = objDocument.AuthenticView
			' now use the authentic view object

		End If
	End If
Else
	MsgBox "No document is open"
End If
```

29.3.2.10.9 Close

Method: Close (*bDiscardChanges* as Boolean)

Description
To close the document call this method. If bDiscardChanges is true and the document is modified, the document will be closed but not saved.

Errors
1400 The object is no longer valid.
1401 Document needs to be saved first.

29.3.2.10.10 ConvertDTDOrSchema

Method: ConvertDTDOrSchema (*nFormat* as SPYDTDSchemaFormat[1991], *nFrequentElements* as SPYFrequentElements[1992])

Parameters
nFormat
Sets the schema output format to DTD or W3C.

nFrequentElements
Create complex elements as elements or complex types.

Description
ConvertDTDOrSchema takes an existing schema format and converts it into a different format.
For a finer tuning of DTD/XSD conversion, use ConvertDTDOrSchemaEx[1821].

Errors

© 2016-2022 Altova GmbH

1400 The object is no longer valid.
1412 Error during conversion. In the case of DTD to DTD or XSD to XSD conversion, the following errors are returned: *DTD to DTD conversion is not supported. Please use function FlattenDTDOrSchema instead* and *Schema to schema conversion is not supported. Please use function FlattenDTDOrSchema instead*.

29.3.2.10.11 ConvertDTDOrSchemaEx

Method: ConvertDTDOrSchemaEx (*nFormat* as SPYDTDSchemaFormat[1991], *nFrequentElements* as SPYFrequentElements[1992], sOutputPath as String, nOutputPathDialogAction as SPYDialogAction[1990])

Parameters
nFormat
Sets the schema output format to DTD, or W3C.

nFrequentElements
Create complex elements as elements or complex types.

`sOutputPath`
The file path for the newly generated file.

`nOutputPathDialogAction`
Defines the dialog interaction for this call.

Description
`ConvertDTDOrSchemaEx` takes an existing schema format and converts it into a different format.

Errors
1400 The object is no longer valid.
1412 Error during conversion. In the case of DTD to DTD or XSD to XSD conversion, the following errors are returned: *DTD to DTD conversion is not supported. Please use function FlattenDTDOrSchema instead* and *Schema to schema conversion is not supported. Please use function FlattenDTDOrSchema instead*.

29.3.2.10.12 ConvertToWSDL20

Method: ConvertToWSDL20 (*sFilePath* as String, *bShowDialogs* as Boolean)

Parameters
sFilePath

This specifies the file name of the converted WSDL. In case the source WSDL includes files which also must be converted, then only the directory part of the given path is used and the file names are generated automatically.

bShowDialogs
Defines whether file/folder selection dialogs are shown.

Description
Converts the WSDL 1.1 document to a WSDL 2.0 file. It will also convert any referenced WSDL files that are referenced from within this document. Note that this functionality is limited to WSDL View only. See Document.CurrentViewMode[1823]. and SPYViewModes[1999].

Errors

1400	The document object is no longer valid.
1407	Invalid parameters have been passed or an empty file name has been specified as output target.
1417	The document is not opened in WSDL view, maybe it is not an '.wsdl' file.
1421	Feature is not available in this edition.
1433	WSDL 1.1 to WSDL 2.0 conversion failed.

29.3.2.10.13 CreateChild

Method: CreateChild (*nKind* as SPYXMLDataKind[2000]) as XMLData[1939]

Return Value
The method returns the new XMLData object.

Description
To create a new XMLData object use the CreateChild() method.

Errors

1400	The object is no longer valid.
1404	Cannot create XMLData object.
1407	Invalid address for the return parameter was specified.

29.3.2.10.14 CreateDBStructureFromXMLSchema

Method: CreateDBStructureFromXMLSchema (*pDatabase* as DatabaseConnection[1803], *pTables* as ElementList[1851], *bDropTableWithExistingName* as Boolean) as String

Description
CreateDBStructureFromXMLSchema exports the given tables to the specified database. The function returns the SQL statements that were necessary to perform the changes.

See also GetDBStructureList[1831].

Errors

1429	Database selection missing.
1430	Document export failed.

© 2016-2022 Altova GmbH

29.3.2.10.15 CreateSchemaDiagram

Method: CreateSchemaDiagram (*nKind* as SPYSchemaDefKind[1996], *strName* as String, *strFile* as String)

Return Value
None.

Description
The method creates a diagram of the schema type strName of kind nKind and saves the output file into strFile. Note that this functionality is limited to Schema View only. See Document.CurrentViewMode[1823]. and SPYViewModes[1999].

Errors
1400 The object is no longer valid.
1414 Failed to save diagram.
1415 Invalid schema definition type specified.

29.3.2.10.16 CurrentViewMode

Method: CurrentViewMode as SPYViewModes[1999]

Description
The property holds the current view mode of the document. See also Document.SwitchViewMode[1840].

Errors
1400 The object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.17 DataRoot

Property: DataRoot as XMLData[1939] (read-only)

Description
This property provides access to the document's first XMLData object of type *spyXMLDataElement*. This is typically the root element for all document content data. See XMLSpyDocument.RootElement[1836] to get the root element of the whole document including XML prolog data. If the CurrentViewMode[1823] is not *spyViewGrid* or *spyViewAuthentic* an UpdateXMLData[1842] may be necessary to get access to the latest XMLData[1939].

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.18 DocEditView

Method: DocEditView as DocEditView

Description
Holds a reference to the current vista Authentic object.

Errors
1400 The object is no longer valid.
1407 Invalid address for the return parameter was specified.
1417 Document needs to be open in authentic view mode.

29.3.2.10.19 Encoding

Property: Encoding as String

Description
This property provides access to the document's encoding value. However, this property can only be accessed when the document is opened in *spyViewGrid*, *spyViewText* or *spyViewAuthentic*. See CurrentViewMode[1823] on how to detect a document's actual view mode.

This property makes the method SetEncoding[1838] obsolete.

Possible values are, for example:

8859-1,
8859-2,
ASCII, ISO-646,
850,
1252,
1255,
SHIFT-JIS, MS-KANJI,
BIG5, FIVE,
UTF-7,
UTF-8,
UTF-16

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.
1416 Operation not supported in current view mode.

29.3.2.10.20 EndChanges

Method: EndChanges()

© 2016-2022 Altova GmbH

Description
Use the method EndChanges to display all changes since the call to Document.StartChanges[1840].

Errors
1400 The object is no longer valid.

29.3.2.10.21 ExecuteXQuery

Method: ExecuteXQuery (*strXMLFileName* as String)

Description
Execute the XQuery statements contained in the document of the document object. Either an XQuery execution or an XQuery Update is performed depending on the file extension of the document. Use the XML file specified in the argument as the XML target document that the XQuery document processes.

- If the document has an XQuery file extension as defined in the Options dialog of XMLSpy, then an XQuery execution is performed. By default: `.xq`, `.xql`, and `.xquery` are set as XQuery file extensions in XMLSpy.
- If the document has an XQuery Update file extension as defined in the Options dialog of XMLSpy, then an XQuery Update action is performed. By default: `.xqu` is set as an XQuery Update file extension in XMLSpy.

If your XQuery script does not use an XML source, set the parameter `strXMLFileName` to an empty string.

Errors
1400 The document object is no longer valid.
1423 XQuery transformation error.
1424 Not all files required for operation could be loaded. Most likely, the file specified in `strXMLFileName` does not exist or is not valid.

29.3.2.10.22 ExportToDatabase

Method: ExportToDatabase (*pFromChild* as XMLData[1939], *pExportSettings* as ExportSettings[1853], *pDatabase* as DatabaseConnection[1803])

Description
ExportToDatabase exports the XML document starting with the element pFromChild. The parameter pExportSettings defines the behaviour of the export (see Application.GetExportSettings[1732]). The parameter pDatabase specifies the destination of the export (see Application.GetDatabaseSettings[1731]). `UpdateXMLData()`[1842] might be indirectly needed as you have to pass the `XMLData`[1939] as parameter to this function.

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.
1416 Error during export.
1429 Database selection missing.

1430 Document export failed.

Example

```
Dim objDoc As Document
Set objDoc = objSpy.ActiveDocument

'set the behaviour of the export with ExportSettings
Dim objExpSettings As ExportSettings
Set objExpSettings = objSpy.GetExportSettings

'set the destination with DatabaseConnection
Dim objDB As DatabaseConnection
Set objDB = objSpy.GetDatabaseSettings

objDB.CreateMissingTables = True
objDB.CreateNew = True
objDB.File = "C:\Export.mdb"

objDoc.ExportToDatabase objDoc.RootElement, objExpSettings, objDB
If Err.Number <> 0 Then
        a = MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) &
                "Description: " & Err.Description)
End If
```

29.3.2.10.23 ExportToText

Method: ExportToText (*pFromChild* as XMLData[1939], *pExportSettings* as ExportSettings[1853], *pTextSettings* as TextImportExportSettings[1897])

Description
ExportToText exports tabular information from the document starting at pFromChild into one or many text files. Columns of the resulting tables are generated in alphabetical order of the column header names. Use GetExportElementList[1831] to learn about the data that will be exported. The parameter pExportSettings defines the specifics for the export. Set the property ExportSettings.ElementList[1854] to the - possibly modified - list returned by GetExportElementList[1831] to avoid exporting all contained tables. The parameter pTextSettings defines the options specific to text export and import. You need to set the property TextImportExportSettings.DestinationFolder[1898] before you call ExportToText. `UpdateXMLData()`[1842] might be indirectly needed as you have to pass the `XMLData`[1939] as parameter to this function.

See also Import and export of data[1694].

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.
1416 Error during export.
1430 Document export failed.

Example

 © 2016-2022 Altova GmbH

```
' ------------------------------------------------------
' VBA client code fragment - export document to text files
' ------------------------------------------------------
        Dim objDoc As Document
        Set objDoc = objSpy.ActiveDocument

        Dim objExpSettings As ExportSettings
        Set objExpSettings = objSpy.GetExportSettings
        objExpSettings.ElementList = objDoc.GetExportElementList(
                                                        objDoc.RootElement,
                                                        objExpSettings)

        Dim objTextExp As TextImportExportSettings
        Set objTextExp = objSpy.GetTextImportExportSettings
        objTextExp.HeaderRow = True
        objTextExp.DestinationFolder = "C:\Exports"

        On Error Resume Next
        objDoc.ExportToText objDoc.RootElement, objExpSettings, objTextExp

        If Err.Number <> 0 Then
                a = MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) & "Description: "
                & Err.Description)
        End If
```

29.3.2.10.24 FlattenDTDOrSchema

Method: FlattenDTDOrSchema (sOutputPath as String, nOutputPathDialogAction as SPYDialogAction[1990])

Parameters

`sOutputPath`
The file path for the newly generated file.

`nOutputPathDialogAction`
Defines the dialog interaction for this call.

Description

`FlattenDTDOrSchema` takes an existing DTD or schema, generates a flattened file, and saves the generated file at the specified location. In the case of DTDs, flattening removes parameter entities and produces a single DTD from a collection of modules; sections marked `IGNORE` are suppressed and unused parameter entities are deleted. When an XML Schema is flattened, (i) the components of all included schemas are added as global components of the active schema, and (ii) included schemas are deleted.

Errors

1400 The object is no longer valid.
1412 Error during conversion.

29.3.2.10.25 FullName

Property: FullName as String

Description
This property can be used to get or set the full file name - including the path - to where the document gets saved. The validity of the name is not verified before the next save operation.

This property makes the methods GetPathName[1832] and SetPathName[1839] obsolete.

Errors
1400 The document object is no longer valid.
1402 Empty string has been specified as full file name.

29.3.2.10.26 GenerateDTDOrSchema

Method: GenerateDTDOrSchema (*nFormat* as SPYDTDSchemaFormat[1991], *nValuesList* as integer, *nDetection* as SPYTypeDetection[1998], *nFrequentElements* as SPYFrequentElements[1992])

Parameters
nFormat
Sets the schema output format to DTD, or W3C.

nValuesList
Generate not more than this amount of enumeration-facets per type. Set to -1 for unlimited.

nDetection
Specifies granularity of simple type detection.

nFrequentElements
Shall the types for all elements be defined as global? Use the value *spyGlobalComplexType* to define them on global scope. Otherwise, use the value *spyGlobalElements*.

Description
Use this method to automatically generate a DTD or schema for the current XML document.
For a finer tuning of DTD / schema generation, use GenerateDTDOrSchemaEx[1828].
Note that this functionality is not available in ZIP View only. See Document.CurrentViewMode[1823]. and SPYViewModes[1999].

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.

29.3.2.10.27 GenerateDTDOrSchemaEx

Method: GenerateDTDOrSchemaEx (*objDlg* as DTDSchemaGeneratorDlg[1847]) as Document[1813]

© 2016-2022 Altova GmbH

Description
Use this method to automatically generate a DTD or schema for the current XML document. A DTDSchemaGeneratorDlg[1847] object is used to pass information to the schema/DTD generator. The generation process can be configured to allow user interaction or run without further user input.
Note that this functionality is not available in ZIP View only. See Document.CurrentViewMode[1823]. and SPYViewModes[1999].

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.

29.3.2.10.28 GenerateProgramCode

Method: GenerateProgramCode (*objDlg* as CodeGeneratorDlg[1797])
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Generate Java, C++ or C# class files from the XML Schema definitions in your document. A CodeGeneratorDlg[1797] object is used to pass information to the code generator. The generation process can be configured to allow user interaction or run without further user input.

Errors
1400 The document object is no longer valid.
1407 An empty file name has been specified.
1421 Feature not available in this edition

29.3.2.10.29 GenerateSampleXML

Method: GenerateSampleXML (*objDlg* as GenerateSampleXMLDlg[1869]) as Document[1813]

Description
Generates a sample XML if the document is a schema or DTD. Use Dialogs.GenerateSampleXMLDlg[1811] to get an initialized set of options.

Available with TypeLibrary version 1.5

Errors
1400 The document object is no longer valid.

29.3.2.10.30 GenerateSchemaDocumentation

Method: GenerateSchemaDocumentation (*objDlg* as SchemaDocumentationDlg[1878])

Description
Generate documentation for a schema definition file in HTML, MS-Word, or RTF format. The parameter objDlg is used to parameterize the generation process. Use Dialogs.SchemaDocumentationDlg[1811] to get an initialized

set of options. As a minimum, you will need to set the property SchemaDocumentationDlg.OutputFile[1885] before starting the generation process. Note that this functionality is limited to Schema View only. See Document.CurrentViewMode[1823] and SPYViewModes[1999].

Errors

1400 The document object is no longer valid.
1407 Invalid parameters have been passed or an empty file name has been specified as output target.
1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
1421 Feature is not available in this edition.
1422 Error during generation

29.3.2.10.31 GenerateWSDL20Documentation

Method: GenerateWSDL20Documentation (*objDlg* as WSDL20DocumentationDlg[1917])

Description

Generate documentation for a WSDL definition file in HTML, MS-Word, or RTF format. The parameter objDlg is used to parameterize the generation process. Use Dialogs.WSDL20DocumentationDlg[1812] to get an initialized set of options. As a minimum, you will need to set the property WSDL20DocumentationDlg.OutputFile[1922] before starting the generation process. Note that this functionality is limited to WSDL View only. See Document.CurrentViewMode[1823] and SPYViewModes[1999].

Errors

1400 The document object is no longer valid.
1407 Invalid parameters have been passed or an empty file name has been specified as output target.
1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
1421 Feature is not available in this edition.
1422 Error during generation

29.3.2.10.32 GenerateWSDLDocumentation

Method: GenerateWSDLDocumentation (*objDlg* as WSDLDocumentationDlg[1905])

Description

Generate documentation for a WSDL definition file in HTML, MS-Word, or RTF format. The parameter objDlg is used to parameterize the generation process. Use Dialogs.WSDLDocumentationDlg[1812] to get an initialized set of options. As a minimum, you will need to set the property WSDLDocumentationDlg.OutputFile[1911] before starting the generation process. Note that this functionality is limited to WSDL View only. See Document.CurrentViewMode[1823] and SPYViewModes[1999].

Errors

1400 The document object is no longer valid.
1407 Invalid parameters have been passed or an empty file name has been specified as output target.
1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
1421 Feature is not available in this edition.

© 2016-2022 Altova GmbH

1422 Error during generation

29.3.2.10.33 GenerateXBRLDocumentation

Method: GenerateXBRLDocumentation (*objDlg* as XBRLDocumentationDlg[1928])

Description
Generate documentation for an XBRL file in HTML, MS-Word, or RTF format. The parameter objDlg is used to parameterize the generation process. Use Dialogs.XBRLDocumentationDlg[1813] to get an initialized set of options. As a minimum, you will need to set the property XBRLDocumentationDlg.OutputFile[1933] before starting the generation process. Note that this functionality is limited to XBRL View only. See Document.CurrentViewMode[1823] and SPYViewModes[1999].

Errors
1400 The document object is no longer valid.
1407 Invalid parameters have been passed or an empty file name has been specified as output target.
1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
1421 Feature is not available in this edition.
1422 Error during generation

29.3.2.10.34 GetDBStructureList

Method: GetDBStructureList (*pDatabase* as DatabaseConnection[1803]) as ElementList[1851]

Description
GetDBStructureList creates a collection of elements from the Schema document for which tables in the specified database are created. The function returns a collection of ElementListItems where the properties ElementListItem.Name[1853] contain the names of the tables.

See also CreateDBStructureFromXMLSchema[1822].

Errors
1400 The object is no longer valid.
1427 Failed creating parser for the specified XML.
1428 Export of element list failed.
1429 Database selection missing.

29.3.2.10.35 GetExportElementList

Method: GetExportElementList (*pFromChild* as XMLData[1939], *pExportSettings* as ExportSettings[1853]) as ElementList[1851]

Description
GetExportElementList creates a collection of elements to export from the document, depending on the settings in pExportSettings and starting from the element pFromChild. The function returns a collection of ElementListItems where the properties ElementListItem.Name[1853] contain the names of the tables that can be exported from the document. The property ElementListItem.FieldCount[1853] contains the number of columns in the table. The property ElementListItem.RecordCount[1853] contains the number of records in the table. The property ElementListItem.ElementKind[1853] is unused. `UpdateXMLData()`[1842] might be indirectly needed as you have to pass the `XMLData`[1939] as parameter to this function.

See also Import and export of data[1694].

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.
1427 Failed creating parser for the specified XML.
1428 Export of element list failed.

29.3.2.10.36 GetPathName (obsolete)

Superseded by Document.FullName[1828]

```
// ----- javascript sample -----
// instead of:
// strPathName = Application.ActiveDocument.GetPathName();
// use now:
strPathName = Application.ActiveDocument.FullName;
```

Method: GetPathName() as String

Description
The method GetPathName gets the path of the active document.

See also Document.SetPathName[1839] (obsolete).

29.3.2.10.37 GridView

Property: GridView as GridView[1875]

Description
This property provides access to the grid view functionality of the document.

Errors
1400 The object is no longer valid.
1407 Invalid address for the return parameter was specified.
1417 Document needs to be open in enhanced grid view mode.

© 2016-2022 Altova GmbH

29.3.2.10.38 IsModified

Property: IsModified as Boolean

Description
True if the document is modified.

Errors
1400 The object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.39 IsValid

Method: HRESULT IsValid([in, out] VARIANT *strError, [in, out] VARIANT *nErrorPos, [in, out] VARIANT *pBadData, [out,retval] VARIANT_BOOL *bValid);

Return Value
True if the document is valid, false if not. To call `IsValid()`, the application GUI must be visible. (If you wish to validate without the GUI being visible, please use Altova RaptorXML Server.)

Description
`IsValid` validates the document against its associated schema or DTD. `strError` gives you the same error message as when you validate the file within the GUI.

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.
1408 Unable to validate file.

Example
The following C++ code snippet provides an example of how to use the **`IsValid`** method.

```
#import "XMLSpy.tlb"


CComPtr< XMLSpyLib::IDocument12> ipDoc = ipXMLSpy->GetActiveDocument();
if ( ipDoc )
{
	// prepare in/out parameters for IsValid call
	CComVariant variantError;
	CComVariant variantErrorPos;
	CComVariant variantBadData;
	// IsValid always shows a dialog with the validation result. This cannot be turned
off.
	bool bIsValid = ipDoc->IsValid( &variantError, &variantErrorPos, &variantBadData )
== VARIANT_TRUE;
```

```
	if ( !bIsValid )
	{
		// retrieve values from out parameters
		CString strError = (V_VT( &variantError ) == VT_BSTR ?
V_BSTR( &variantError ) : _T( "" ));
		long npos = (V_VT( &variantErrorPos ) == VT_I4 ? V_I4( &variantErrorPos ) : -
1);
		CComQIPtr< XMLSpyLib::IXMLData > ipXMLBadData = (V_VT( &variantBadData ) ==
VT_DISPATCH ? V_DISPATCH( &variantBadData ) : nullptr);

		if ( ipXMLBadData )
			strError += CString( _T("\n\n Node: ") ) + (LPCWSTR)ipXMLBadData-
>GetName();

		if ( !strError.IsEmpty() )
			AfxMessageBox( "Validation failed - " + strError );
	}
}
```

29.3.2.10.40 IsValidEx

Method: IsValidEx (*nXSDVersion* as SPYValidateXSDVersion[1999], *nErrorLimit* as `int`, *nErrorFormat* as SPYValidateErrorFormat[1999], out *strError* as Variant) as Boolean

Return Value
True if the document is valid, false if not.

Description
`IsValidEx` validates the document against its associated schema or DTD.

In parameters:
`nXSDVersion` which is an enumeration value of SPYValidateXSDVersion[1999] that selects the XSD version to validate against.
`nErrorLimit` which is an integer. Values must be `1` to `999`.
`nErrorFormat` which is an enumeration value of SPYValidateErrorFormat[1999] that selects the XSD version to validate against.

Out parameter:
`strError` is the error message, and is the same as that received when validating the file within the GUI.

Errors

1400	The object is no longer valid.
1407	Invalid parameter or invalid address for the return parameter was specified.
1408	Unable to validate file.

Example
The following C++ code snippet provides an example of how to use the **`IsValidEx`** method.

© 2016-2022 Altova GmbH

```
#import "XMLSpy.tlb"


CComPtr< XMLSpyLib::IDocument12> ipDoc = ipXMLSpy->GetActiveDocument();
if ( ipDoc )
{
	CComVariant variantErrorEx;
	bool bIsValidEx = ipDoc->IsValidEx( XMLSpyLib::spyValidateXSDVersion_AutoDetect,
100, XMLSpyLib::SPYValidateErrorFormat_LongXML, &variantErrorEx ) == VARIANT_TRUE;

	//	// retrieve values from out parameters
	CString strErrorEx = (V_VT( &variantErrorEx ) == VT_BSTR ?
V_BSTR( &variantErrorEx ) : _T( "" ));
	if ( !strErrorEx.IsEmpty() )
		AfxMessageBox( "Validation failed - " + strErrorEx );
}
```

29.3.2.10.41 IsWellFormed

Method: IsWellFormed (*pData* as XMLData, *bWithChildren* as Boolean, *strError* as Variant, *nErrorPos* as Variant, *pBadXMLData* as Variant) as Boolean

Return Value
True if the document is well formed.

Description
IsWellFormed checks the document for well-formedness starting at the element pData.

If the document is not well formed, strError contains an error message, nErrorPos the position in the file and pBadXMLData holds a reference to the element which breaks the well-formedness. These out-parameters are defined as VARIANTs to support scripting languages like VBScript.

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.

Example

See IsValid.

29.3.2.10.42 Name

Property: Name as String (read-only)

Description
Use this property to retrieve the name - not including the path - of the document file. To change the file name for a document use the property FullName[1828].

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.43 Parent

Property: Parent as Documents[1843] (read-only)

Description
Access the parent of the document object.

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

Property: Parent as Application[1724] (read-only)

29.3.2.10.44 Path

Property: Path as String (read-only)

Description
Use this property to retrieve the path - not including the file name - of the document file. To change the file name and path for a document use the property FullName[1828].

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.45 RootElement

Property: RootElement as XMLData[1939] (read-only)

Description
The property RootElement provides access to the root element of the XML structure of the document including the XML prolog data. To access the first element of a document's content navigate to the first child of kind *spyXMLDataElement* or use the Document.DataRoot[1823] property. If the CurrentViewMode[1823] is not *spyViewGrid* or *spyViewAuthentic* an UpdateXMLData[1842] may be necessary to get access to the latest XMLData[1939].

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.10.46 Save

Method: Save()

Description
The method writes any modifications of the document to the associated file. See also Document.FullName[1828].

Errors
1400 The document object is no longer valid.
1407 An empty file name has been specified.
1403 Error when saving file, probably the file name is invalid.

29.3.2.10.47 SaveAs

Method: SaveAs (*strFileName* as String)

Description
Save the document to the file specified. If saving was successful, the FullName[1828] property gets set to the specified file name.

Errors
1400 The document object is no longer valid.
1407 An empty file name has been specified.
1403 Error when saving file, probably the file name is invalid.

29.3.2.10.48 Saved

Property: Saved as Boolean (read-only)

Description
This property can be used to check if the document has been saved after the last modifications. It returns the negation of IsModified[1833].

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.49 SaveInString

Method: SaveInString (*pData* as XMLData[1939], *bMarked* as Boolean) as String

Parameters
pData
XMLData element to start. Set pData to Document.RootElement[1836] if you want to copy the complete file.

bMarked

If bMarked is true, only the elements selected in the grid view are copied.

Return Value
Returns a string with the XML data.

Description
SaveInString starts at the element pData and converts the XMLData objects to a string representation. UpdateXMLData() 1842 might be indirectly needed as you have to pass the XMLData 1939 as parameter to this function.

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.

29.3.2.10.50 SaveToURL

Method: SaveToURL (*strURL* as String, *strUser* as String, *strPassword* as String)

Return Value

Description
SaveToURL() writes the document to the URL strURL. This method does not set the permanent file path of the document.

Errors
1400 The object is no longer valid.
1402 Invalid URL specified.
1403 Error while saving to URL.

29.3.2.10.51 SetActiveDocument

Method: SetActiveDocument()

Description
The method sets the document as the active and brings it to the front.

Errors
1400 The object is no longer valid.

29.3.2.10.52 SetEncoding (obsolete)

Superseded by Document.Encoding 1824

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.SetEncoding("UTF-16");
// use now:
```


© 2016-2022 Altova GmbH

```
Application.ActiveDocument.Encoding = "UTF-16";
```

Method: SetEncoding (*strEncoding* as String)

Description
SetEncoding sets the encoding of the document like the menu item "File/Encoding..." in XMLSpy. Possible values for strEncoding are, for example:

8859-1,
8859-2,
ASCII, ISO-646,
850,
1252,
1255,
SHIFT-JIS, MS-KANJI,
BIG5, FIVE,
UTF-7,
UTF-8,
UTF-16

29.3.2.10.53 SetExternalIsValid

Method: SetExternalIsValid (*bValid* as Boolean)

Parameters

bValid
Sets the result of an external validation process.

Description
The internal information set by this method is only queried on cancelling the default validation in any OnBeforeValidate 1816 handler.

Available with TypeLibrary version 1.5

Errors
1400 The object is no longer valid.

29.3.2.10.54 SetPathName (obsolete)

Superseded by Document.FullName 1828

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.SetPathName("C:\\myXMLFiles\\test.xml");
```

```
// use now:
Application.ActiveDocument.FullName = "C:\\myXMLFiles\\test.xml";
```

Method: SetPathName (*strPath* as String)

Description
The method SetPathName sets the path of the active document. SetPathName only copies the string and does not check if the path is valid. All succeeding save operations are done into this file.

29.3.2.10.55 StartChanges

Method: StartChanges()

Description
After StartChanges is executed XMLSpy will not update its editor windows until Document.EndChanges[1824] is called. This increases performance of complex tasks to the XML structure.

Errors
1400 The object is no longer valid.

29.3.2.10.56 Suggestions

Property: Suggestions as Array

Description
This property contains the last valid user suggestions for this document. The XMLSpy generated suggestions can be modified before they are shown to the user in the OnBeforeShowSuggestions[1900] event.

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.

29.3.2.10.57 SwitchViewMode

Method: SwitchViewMode (*nMode* as SPYViewModes[1999]) as Boolean

Return value
Returns true if view mode is switched.

Description
The method sets the current view mode of the document in XMLSpy. See also Document.CurrentViewMode[1823].

Errors
1400 The object is no longer valid.
1407 Invalid address for the return parameter was specified.
1417 Invalid view mode specified.

© 2016-2022 Altova GmbH

29.3.2.10.58 TextView

Property: TextView as TextView[1900]

Description
This property provides access to the text view functionality of the document.

Errors
1400 The object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.59 Title

Property: Title as String (read-only)

Description
Title contains the file name of the document. To get the path and filename of the file use FullName[1828].

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.60 TransformXSL

Method: TransformXSL()

Description
TransformXSL processes the XML document via the associated XSL file. See Document.AssignXSL[1818] on how to place a reference to a XSL file into the document.

Errors
1400 The document object is no longer valid.
1411 Error during transformation process.

29.3.2.10.61 TransformXSLEx

Method: TransformXSLEx(*nAction* as SPYDialogAction[1990])

Description
TransformXSLEx processes the XML document via the associated XSL file. The parameter specifies whether a dialog asking for the result document name should pop up or not. See Document.AssignXSL[1818] on how to place a reference to a XSL file into the document.

Errors
1400 The document object is no longer valid.
1411 Error during transformation process.

29.3.2.10.62 TransformXSLFO

Method: TransformXSLFO()

Description
TransformXSLFO processes the XML document via the associated XSLFO file. See AssignXSLFO[1819] on how to place a reference to a XSLFO file into the document. You need to assign a FOP processor to XMLSpy before you can use this method.

Errors
1400 The document object is no longer valid.
1411 Error during transformation process.

29.3.2.10.63 TreatXBRLInconsistenciesAsErrors

Property: TreatXBRLInconsistenciesAsErrors as Boolean

Description
If this is set to `true` the `Document.IsValid()` method will return `false` for XBRL instances containing inconsistencies as defined by the XBRL Specification. The default value of this property is `false`.

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.64 UpdateViews

Method: UpdateViews()

Description
To redraw the Enhanced Grid View and the Tree View call UpdateViews. This can be important after you changed the XMLData structure of a document. This method does not redraw the text view of XMLSpy.

Errors
1400 The document object is no longer valid.

29.3.2.10.65 UpdateXMLData

Method: UpdateXMLData() as Boolean

Description
The XMLData[1939] tree is updated from the current view. Please note that this can fail in case of the TextView if the current XML text is not well-formed. This is not necessary if CurrentViewMode[1823] is *spyViewGrid* or *spyViewAuthentic* because these views keep the XMLData[1939] updated.

© 2016-2022 Altova GmbH

Available with TypeLibrary version 1.5

Errors
1400 The document object is no longer valid.

29.3.2.11 Documents

Properties
Count 1844
Item 1844

Methods
NewAuthenticFile 1844
NewFile 1845
NewFileFromText 1845
OpenAuthenticFile 1845
OpenFile 1846
OpenURL 1846
OpenURLDialog 1847

Description
This object represents the set of documents currently open in XMLSpy. Use this object to open further documents or iterate through already opened documents.

Examples
```
' ----------------------------------------
' XMLSpy scripting environment - VBScript
' iterate through open documents
' ----------------------------------------
Dim objDocuments
Set objDocuments = Application.Documents

For Each objDoc In objDocuments
        'do something useful with your document
        objDoc.SetActiveDocument()
Next
```

```
// ----------------------------------------
// XMLSpy scripting environment - JScript
// close all open documents
// ----------------------------------------
for (var iter = new Enumerator (Application.Documents);
    ! iter.atEnd();
    iter.moveNext())
{
        // MsgBox ("Closing file " + iter.item().Name);
        iter.item().Close (true);
```

}

29.3.2.11.1 Count

Property: Count as long

Description
Count of open documents.

Errors
1600 Invalid `Documents` object
1601 Invalid input parameter

29.3.2.11.2 Item

Method: Item (*n* as long) as Document[1813]

Description
Gets the document with the index n in this collection. Index is 1-based.

Errors
1600 Invalid `Documents` object
1601 Invalid input parameter

29.3.2.11.3 NewAuthenticFile

Method: NewAuthenticFile (*strSPSPath* as String, *strXMLPath* as String) as Document[1813]

Parameters
strSPSPath
The path to the SPS document.

strXMLPath
The new XML document name.

Return Value
The method returns the new document.

Description
NewAuthenticFile creates a new XML file and opens it in Authentic View using SPS design strSPSPath.

© 2016-2022 Altova GmbH

29.3.2.11.4 NewFile

Method: NewFile (*strFile* as String, *strType* as String) as Document[1813]

Parameters
strFile
Full path of new file.

strType
Type of new file as string (i.e. "xml", "xsd", ...)

Return Value
Returns the new file.

Description
NewFile creates a new file of type strType (i.e. "xml"). The newly created file is also the ActiveDocument.

29.3.2.11.5 NewFileFromText

Method: NewFileFromText (*strText* as String, *strType* as String) as Document[1813]

Parameters
strText
The content of the new document in plain text.

strType
Type of the document to create (i.e. "xml").

Return Value
The method returns the new document.

Description
NewFileFromText creates a new document with strText as its content.

29.3.2.11.6 OpenAuthenticFile

Method: OpenAuthenticFile (*strSPSPath* as String, *strXMLPath* as String) as Document[1813]

Parameters
strSPSPath
The path to the SPS document.

strXMLPath
The path to the XML document (can be empty).

Return Value
The method returns the new document.

Description
OpenAuthenticFile opens an XML file or database in Authentic View using SPS design strSPSPath.

29.3.2.11.7 OpenFile

Method: OpenFile (*strPath* as String, *bDialog* as Boolean) as Document[1813]

Parameters
strPath
Path and file name of file to open.

bDialog
Show dialogs for user input.

Return Value
Returns the opened file on success.

Description
OpenFile opens the file strPath. If bDialog is TRUE, a file-dialog will be displayed.

Example

```
Dim objDoc As Document
Set objDoc = objSpy.Documents.OpenFile(strFile, False)
```

29.3.2.11.8 OpenURL

Method: OpenURL (*strURL* as String, *nURLType* as SPYURLTypes[1998], *nLoading* as SPYLoading[1993], *strUser* as String, *strPassword* as String) as Document[1813]

Parameters
strURL
URL to open as document.

nURLType
Type of document to open. Set to -1 for auto detection.

nLoading
Set nLoading to 0 (zero) if you want to load it from cache or proxy. Otherwise set nLoading to 1.

strUser
Name of the user if required. Can be empty.

strPassword
Password for authentification. Can be empty.

Return Value
The method returns the opened document.

Description

© 2016-2022 Altova GmbH

OpenURL opens the URL strURL.

29.3.2.11.9 OpenURLDialog

Method: OpenURLDialog (*strURL* as String, *nURLType* as SPYURLTypes[1998], *nLoading* as SPYLoading[1993], *strUser* as String, *strPassword* as String) as Document[1813]

Parameters
strURL
URL to open as document.

nURLType
Type of document to open. Set to -1 for auto detection.

nLoading
Set nLoading to 0 (zero) if you want to load it from cache or proxy. Otherwise set nLoading to 1.

strUser
Name of the user if required. Can be empty.

strPassword
Password for authentification. Can be empty.

Return Value
The method returns the opened document.

Description
OpenURLDialog displays the "open URL" dialog to the user and presets the input fields with the given parameters.

29.3.2.12 DTDSchemaGeneratorDlg

Properties and Methods

Standard automation properties
Application[1848]
Parent[1850]

DTDSchemaFormat[1848]
ValueList[1851]
TypeDetection[1851]
FrequentElements[1848]
MergeAllEqualNamed[1849]
ResolveEntities[1851]
AttributeTypeDefinition[1848]
GlobalAttributes[1849]
OnlyStringEnums[1850]
MaxEnumLength[1849]
OutputPath[1850]
OutputPathDialogAction[1850]

Description
Use this object to configure the generation of a schema or DTD. The method GenerateDTDOrSchemaEx[1828] expects a DTDSchemaGeneratorDlg as parameter to configure the generation as well as the associated user interactions.

29.3.2.12.1 Application

Property: Application as Application[1724] (read-only)

Description
Access the XMLSpy application object.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.2 AttributeTypeDefinition

Property: AttributeTypeDefinition as SPYAttributeTypeDefinition[1988]

Description
Specifies how attribute definitions get merged.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.3 DTDSchemaFormat

Property: DTDSchemaFormat as SPYDTDSchemaFormat[1991]

Description
Sets the schema output format to DTD, or W3C.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.4 FrequentElements

Property: FrequentElements as SPYFrequentElements[1992]

Description

© 2016-2022 Altova GmbH

Shall the types for all elements be defined as global? Use the value *spyGlobalComplexType* to define them on global scope. Otherwise, use the value *spyGlobalElements*.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.5 GlobalAttributes

Property: GlobalAttributes as Boolean

Description
Shall attributes with same name and type be resolved globally?

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.6 MaxEnumLength

Property: MaxEnumLength as Integer

Description
Specifies the maximum number of characters allowed for enumeration names. If one value is longer than this, no enumeration will be generated.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.7 MergeAllEqualNamed

Property: MergeAllEqualNamed as Boolean

Description
Shall types of all elements with the same name be merged into one type?

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.8 OnlyStringEnums

Property: OnlyStringEnums as Boolean

Description
Specifies if enumerations will be created only for plain strings or all types of values.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.9 OutputPath

Property: OutputPath as String

Description
Selects the file name for the generated schema/DTD.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.10 OutputPathDialogAction

Property: OutputPathDialogAction as SPYDialogAction 1990

Description
Defines how the sub-dialog for selecting the schema/DTD output path gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current value of the OutputPath 1850 property as default. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.11 Parent

Property: Parent as Dialogs 1809 (read-only)

Description
Access the parent of the object.

Errors
3000 The object is no longer valid.

© 2016-2022 Altova GmbH

3001 Invalid address for the return parameter was specified.

29.3.2.12.12 ResolveEntities

Property: ResolveEntities as Boolean

Description
Shall all entities be resolved before generation starts? If yes, an info-set will be built.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.13 TypeDetection

Property: TypeDetection as SPYTypeDetection 1998

Description
Specifies granularity of simple type detection.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.14 ValueList

Property: ValueList as Integer

Description
Generate not more than this amount of enumeration-facets per type. Set to -1 for unlimited.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.13 ElementList

Properties
Count 1852
Item 1852

Methods
RemoveElement 1852

Description
Element lists are used for different purposes during export and import of data. Depending on this purpose, different properties of ElementListItem[1852] are used.

It can hold
- a list of table names returned by a call to Application.GetDatabaseTables[1732],
- a list of field names retuned by a call to Application.GetDatabaseImportElementList[1731] or Application.GetTextImportElementList[1733],
- a field name filter list used in Application.ImportFromDatabase[1735] and Application.ImportFromText[1736],
- a list of table names and counts for their rows and columns as returned by calls to GetExportElementList[1831] or
- a field name filter list used in Document.ExportToDatabase[1825] and Document.ExportToText[1826].

29.3.2.13.1 Count

Property: Count as long (read-only)

Description
Count of elements in this collection.

29.3.2.13.2 Item

Method: Item(n as long) as ElementListItem[1852]

Description
Gets the element with the index n from this collection. The first item has index 1.

29.3.2.13.3 RemoveElement

Method: RemoveElement(Index as long)

Description
RemoveElement removes the element Index from the collection. The first Item has index 1.

29.3.2.14 ElementListItem

Properties
Name[1853]

ElementKind[1853]

FieldCount[1853]
RecordCount[1853]

Description
An element in an ElementList[1851]. Usage of its properties depends on the purpose of the element list. For details see ElementList[1851].

© 2016-2022 Altova GmbH

29.3.2.14.1 ElementKind

Property: ElementKind as SPYXMLDataKind 2000

Description
Specifies if a field should be imported as XML element (data value of spyXMLDataElement) or attribute (data value of spyXMLDataAttr).

29.3.2.14.2 FieldCount

Property: FieldCount as long (read-only)

Description
Count of fields (i.e. columns) in the table described by this element. This property is only valid after a call to Document.GetExportElementList 1831.

29.3.2.14.3 Name

Property: Name as String (read-only)

Description
Name of the element. This is either the name of a table or a field, depending on the purpose of the element list.

29.3.2.14.4 RecordCount

Property: RecordCount as long (read-only)

Description
Count of records (i.e. rows) in the table described by this element. This property is only valid after a call to Document.GetExportElementList 1831.

29.3.2.15 ExportSettings

Properties

ElementList 1854

EntitiesToText 1854

ExportAllElements 1854
SubLevelLimit 1856

FromAttributes 1855
FromSingleSubElements 1855
FromTextValues 1855

CreateKeys 1854
IndependentPrimaryKey 1855

Namespace 1855

ExportCompleteXML 1855
StartFromElement 1856

Description
ExportSettings contains options used during export of XML data to a database or text file. See Import and export of data 1694 for a general overview.

29.3.2.15.1 CreateKeys

Property: CreateKeys as Boolean

Description
This property turns creation of keys (i.e. primary key and foreign key) on or off. Default is True.

29.3.2.15.2 ElementList

Property: ElementList as ElementList 1851

Description
Default is empty list. This list of elements defines which fields will be exported. To get the list of available fields use Document.GetExportElementList 1831. It is possible to prevent exporting columns by removing elements from this list with ElementList.RemoveElement 1852 before passing it to Document.ExportToDatabase 1825 or Document.ExportToText 1826.

29.3.2.15.3 EntitiesToText

Property: EntitiesToText as Boolean

Description
Defines if XML entities should be converted to text or left as they are during export. Default is True.

29.3.2.15.4 ExportAllElements

Property: ExportAllElements as Boolean

Description
If set to true, all elements in the document will be exported. If set to false, then ExportSettings.SubLevelLimit 1856 is used to restrict the number of sub levels to export. Default is true.

© 2016-2022 Altova GmbH

29.3.2.15.5 ExportCompleteXML

Property: ExportCompleteXML as Boolean

Description
Defines whether the complete XML is exported or only the element specified by StartFromElement[1856] and its children. Default is True.

29.3.2.15.6 FromAttributes

Property: FromAttributes as Boolean

Description
Set FromAttributes to false if no export data should be created from attributes. Default is True.

29.3.2.15.7 FromSingleSubElements

Property: FromSingleSubElements as Boolean

Description
Set FromSingleSubElements to false if no export data should be created from elements. Default is True.

29.3.2.15.8 FromTextValues

Property: FromTextValues as Boolean

Description
Set FromTextValues to false if no export data should be created from text values. Default is True.

29.3.2.15.9 IndependentPrimaryKey

Property: IndependentPrimaryKey as Boolean

Description
Turns creation of independent primary key counter for every element on or off. If ExportSettings.CreateKeys[1854] is False, this property will be ignored. Default is True.

29.3.2.15.10 Namespace

Property: Namespace as SPYExportNamespace[1991]

Description
The default setting removes all namespace prefixes from the element names. In some database formats the colon is not a legal character. Default is spyNoNamespace.

29.3.2.15.11 StartFromElement

Property: StartFromElement as String

Description
Specifies the start element for the export. This property is only considered when ExportCompleteXML[1855] is false.

29.3.2.15.12 SubLevelLimit

Property: SubLevelLimit as Integer

Description
Defines the number of sub levels to include for the export. Default is 0. This property is ignored if ExportSettings.ExportAllElements[1854] is true.

29.3.2.16 FileSelectionDlg

Properties and Methods

Standard automation properties
Application[1856]
Parent[1857]

Dialog properties
FullName[1857]

Acceptance or cancellation of action that caused event
DialogAction[1857]

Description
The dialog object allows you to receive information about an event and pass back information to the event handler in the same way as with a user dialog. Use the FileSelectionDlg.FullName[1857] to select or modify the file path and set the FileSelectionDlg.DialogAction[1857] property to cancel or agree with the action that caused the event.

29.3.2.16.1 Application

Property: Application as Application[1724] (read-only)

Description
Access the XMLSpy application object.

Errors
2400 The object is no longer valid.
2401 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.16.2 DialogAction

Property: DialogAction as SPYDialogAction[1990]

Description
If you want your script to perform the file selection operation without any user interaction necessary, simulate user interaction by either setting the property to *spyDialogOK(0)* or *spyDialogCancel(1)*.
To allow your script to fill in the default values but let the user see and react on the dialog, use the value *spyDialogUserInput(2)*. If you receive a FileSelectionDlg object in an event handler, *spyDialogUserInput(2)* is not supported and will be interpreted as *spyDialogOK(0)*.

Errors

2400	The object is no longer valid.
2401	Invalid value for dialog action or invalid address for the return parameter was specified.

29.3.2.16.3 FullName

Property: FullName as String

Description
Access the full path of the file the gets selected by the dialog. Most events that pass a FileSelectionDlg object to you allow you modify this value and thus influence the action that caused the event (e.g. load or save to a different location).

Errors

2400	The object is no longer valid.
2401	Invalid address for the return parameter was specified.

29.3.2.16.4 Parent

Property: Parent as Dialogs[1809] (read-only)

Description
Access the parent of the object.

Errors

2400	The object is no longer valid.
2401	Invalid address for the return parameter was specified.

29.3.2.17 FindInFilesDlg

Properties and Methods

Standard automation properties
Application[1858]
Parent[1860]

Find 1859
RegularExpression 1860
Replace 1861
DoReplace 1859
ReplaceOnDisk 1861
MatchWholeWord 1860
MatchCase 1860
SearchLocation 1861
StartFolder 1862
IncludeSubfolders 1859
SearchInProjectFilesDoExternal 1861
FileExtension 1859
AdvancedXMLSearch 1858
XMLElementNames 1863
XMLElementContents 1863
XMLAttributeNames 1862
XMLAttributeContents 1862
XMLComments 1863
XMLCData 1863
XMLPI 1864
XMLRest 1864
ShowResult 1862

Description
Use this object to configure the search (or replacement) for strings in files. The method FindInFiles 1730 expects a FindInFilesDlg as parameter.

29.3.2.17.1 AdvancedXMLSearch

Property: AdvancedXMLSearch as Boolean

Description
Specifies if the XML search properties (XMLElementNames 1863, XMLElementContents 1863, XMLAttributeNames 1862, XMLAttributeContents 1863, XMLComments 1863, XMLCData 1863, XMLPI 1864 and XMLRest 1864) are considered. The default is false.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.2 Application

Property: Application as Application 1724 (read-only)

Description
Access the XMLSpy application object.

Errors
3500 The object is no longer valid.

© 2016-2022 Altova GmbH

3501 Invalid address for the return parameter was specified.

29.3.2.17.3 DoReplace

Property: DoReplace as Boolean

Description
Specifies if the matched string is replaced by the string defined in Replace[1861]. The default is false.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.4 FileExtension

Property: FileExtension as String

Description
Specifies the file filter of the files that should be considered during the search. Multiple file filters must be delimited with a semicolon (eg: *.xml;*.dtd;a*.xsd). Use the wildcards * and ? to define the file filter.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.5 Find

Property: Find as String

Description
Specifies the string to search for.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.6 IncludeSubfolders

Property: IncludeSubfolders as Boolean

Description
Specifies if subfolders are searched too. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.7 MatchCase

Property: MatchCase as Boolean

Description
Specifies if the search is case sensitive. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.8 MatchWholeWord

Property: MatchWholeWord as Boolean

Description
Specifies whether the whole word or just a part of it must match. The default is false.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.9 Parent

Property: Parent as Dialogs[1809] (read-only)

Description
Access the parent of the object.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.10 RegularExpression

Property: RegularExpression as Boolean

Description
Specifies if Find[1859] contains a regular expression. The default is false.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.17.11 Replace

Property: Replace as String

Description
Specifies the replacement string. The matched string is only replaced if DoReplace[1859] is set true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.12 ReplaceOnDisk

Property: ReplaceOnDisk as Boolean

Description
Specifies if the replacement is done directly on disk. The modified file is not opened. The default is false.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.13 SearchInProjectFilesDoExternal

Property: SearchInProjectFilesDoExternal as Boolean

Description
Specifies if the external folders in the open project are searched, when a project search is performed. The default is false.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.14 SearchLocation

Property: SearchLocation as SPYFindInFilesSearchLocation[1991]

Description
Specifies the location of the search. The default is spyFindInFiles_Documents.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.15 ShowResult

Property: ShowResult as Boolean

Description
Specifies if the result is displayed in the Find in Files output window. The default is false.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.16 StartFolder

Property: StartFolder as String

Description
Specifies the folder where the disk search starts.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.17 XMLAttributeContents

Property: XMLAttributeContents as Boolean

Description
Specifies if attribute contents are searched when AdvancedXMLSearch[1858] is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.18 XMLAttributeNames

Property: XMLAttributeNames as Boolean

Description
Specifies if attribute names are searched when AdvancedXMLSearch[1858] is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.17.19 XMLCData

Property: XMLCData as Boolean

Description
Specifies if CData tags are searched when AdvancedXMLSearch[1858] is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.20 XMLComments

Property: XMLComments as Boolean

Description
Specifies if comments are searched when AdvancedXMLSearch[1858] is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.21 XMLElementContents

Property: XMLElementContents as Boolean

Description
Specifies if element contents are searched when AdvancedXMLSearch[1858] is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.22 XMLElementNames

Property: XMLElementNames as Boolean

Description
Specifies if element names are searched when AdvancedXMLSearch[1858] is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.23 XMLPI

Property: XMLPI as Boolean

Description
Specifies if XML processing instructions are searched when AdvancedXMLSearch[1858] is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.24 XMLRest

Property: XMLRest as Boolean

Description
Specifies if the rest of the XML (which is not covered by the other XML search properties) is searched when AdvancedXMLSearch[1858] is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.18 FindInFilesResult

Properties and Methods

Standard automation properties
Application[1864]
Parent[1865]

Count[1865]
Item[1865]

Path[1866]
Document[1865]

Description
This object represents a file that matched the search criteria. It contains a list of FindInFilesResultMatch[1866] objects that describe the matching position.

29.3.2.18.1 Application

Property: Application as Application[1724] (read-only)

Description

© 2016-2022 Altova GmbH

Access the XMLSpy application object.

Errors
3700 The object is no longer valid.
3701 Invalid address for the return parameter was specified.

29.3.2.18.2 Count

Property: Count as long (read-only)

Description
Count of elements in this collection.

29.3.2.18.3 Document

Property: Path as Document[1813] (read-only)

Description
This property returns the Document[1813] object if the matched file is already open in XMLSpy.

Errors
3700 The object is no longer valid.
3701 Invalid address for the return parameter was specified.

29.3.2.18.4 Item

Method: Item(n as long) as FindInFilesResultMatch[1866]

Description
Gets the element with the index n from this collection. The first item has index 1.

29.3.2.18.5 Parent

Property: Parent as FindInFilesResults[1868] (read-only)

Description
Access the parent of the object.

Errors
3700 The object is no longer valid.
3701 Invalid address for the return parameter was specified.

29.3.2.18.6 Path

Property: Path as String (read-only)

Description
Returns the path of the file that matched the search criteria.

Errors
3700 The object is no longer valid.
3701 Invalid address for the return parameter was specified.

29.3.2.19 FindInFilesResultMatch

Properties and Methods

Standard automation properties
Application[1866]
Parent[1867]

Line[1867]
Position[1867]
Length[1866]
LineText[1867]
Replaced[1868]

Description
Contains the exact position in the file of the matched string.

29.3.2.19.1 Application

Property: Application as Application[1724] (read-only)

Description
Access the XMLSpy application object.

Errors
3800 The object is no longer valid.
3801 Invalid address for the return parameter was specified.

29.3.2.19.2 Length

Property: Length as Long (read-only)

Description
Returns the length of the matched string.

© 2016-2022 Altova GmbH

Errors
3800 The object is no longer valid.
3801 Invalid address for the return parameter was specified.

29.3.2.19.3 Line

Property: Line as Long (read-only)

Description
Returns the line number of the match. The line numbering starts with 0.

Errors
3800 The object is no longer valid.
3801 Invalid address for the return parameter was specified.

29.3.2.19.4 LineText

Property: LineText as String (read-only)

Description
Returns the text of the line.

Errors
3800 The object is no longer valid.
3801 Invalid address for the return parameter was specified.

29.3.2.19.5 Parent

Property: Parent as FindInFilesResult 1864 (read-only)

Description
Access the parent of the object.

Errors
3800 The object is no longer valid.
3801 Invalid address for the return parameter was specified.

29.3.2.19.6 Position

Property: Position as Long (read-only)

Description

Returns the start position of the match in the line. The position numbering starts with 0.

Errors
3800 The object is no longer valid.
3801 Invalid address for the return parameter was specified.

29.3.2.19.7 Replaced

Property: Replaced as Boolean (read-only)

Description
True if the matched string was replaced.

Errors
3800 The object is no longer valid.
3801 Invalid address for the return parameter was specified.

29.3.2.20 FindInFilesResults

Properties and Methods

Standard automation properties
Application[1868]
Parent[1869]

Count[1869]
Item[1869]

Description
This is the result of the FindInFiles[1730] method. It is a list of FindInFilesResult[1864] objects.

29.3.2.20.1 Application

Property: Application as Application[1724] (read-only)

Description
Access the XMLSpy application object.

Errors
3600 The object is no longer valid.
3601 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.20.2 Count

Property: Count as long (read-only)

Description
Count of elements in this collection.

29.3.2.20.3 Item

Method: Item(n as long) as FindInFilesResult[1864]

Description
Gets the element with the index n from this collection. The first item has index 1.

29.3.2.20.4 Parent

Property: Parent as Application[1724] (read-only)

Description
Access the parent of the object.

Errors
3600 The object is no longer valid.
3601 Invalid address for the return parameter was specified.

29.3.2.21 GenerateSampleXMLDlg

Properties and Methods

Standard automation properties
Application[1870]
Parent[1873]

NonMandatoryAttributes[1872]
NonMandatoryElements[1872]
RepeatCount[1873]
FillAttributesWithSampleData[1871]
FillElementsWithSampleData[1871]
ContentOfNillableElementsIsNonMandatory[1871]
TryToUseNonAbstractTypes[1874]
SchemaOrDTDAssignment[1874]
LocalNameOfRootElement[1872]
NamespaceURIOfRootElement[1872]
OptionsDialogAction[1873]

Properties that are no longer supported
TakeFirstChoice - obsolete[1874]

FillWithSampleData - obsolete 1871
Optimization - obsolete 1873

Description
Used to set the parameters for the generation of sample XML instances based on a W3C schema or DTD.

29.3.2.21.1 Application

Property: Application as Application 1724 (read-only)

Description
Access the XMLSpy application object.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.2 ChoiceMode

Property: `ChoiceMode` as `SPYSampleXMLGenerationChoiceMode` 1995

Description
Specifies which elements will be generated.

Errors

2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.3 ConsiderSampleValueHints

Property: `ConsiderSampleValueHints` as Boolean

Description
Selects whether to use `SampleValueHints` 1874 or not.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.21.4 ContentOfNillableElementsIsNonMandatory

Property: ContentOfNillableElementsIsNonMandatory as Boolean

Description
If true, the contents of elements that are nillable will not be treated as mandatory.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.5 FillAttributesWithSampleData

Property: FillAttributesWithSampleData as Boolean

Description
If true, attributes will have sample content.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.6 FillElementsWithSampleData

Property: FillElementsWithSampleData as Boolean

Description
If true, elements will have sample content.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.7 FillWithSampleData - obsolete

Property: FillWithSampleData as Boolean

Description
Do no longer access this property. Use FillAttributesWithSampleData 1871 and FillElementsWithSampleData 1871 , instead.

Errors
0001 The property is no longer accessible.

29.3.2.21.8 LocalNameOfRootElement

Property: LocalNameOfRootElement as String

Description
Specifies the local name of the root element for the generated sample XML.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.9 NamespaceURIOfRootElement

Property: NamespaceURIOfRootElement as String

Description
Specifies the namespace URI of the root element for the generated sample XML.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.10 NonMandatoryAttributes

Property: NonMandatoryAttributes as Boolean

Description
If true attributes which are not mandatory are created in the sample XML instance file.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.11 NonMandatoryElements

Property: NonMandatoryElements as Boolean

Description
If `true`, elements which are not mandatory are created in the sample XML instance file.

Errors
2200 The object is no longer valid.
2201 Invalid address was specified for the return parameter.

© 2016-2022 Altova GmbH

29.3.2.21.12 Optimization - obsolete

Property: Optimization as SPYSampleXMLGenerationOptimization[1995]

Description
Do not use this property any longer. Use ChoiceMode and NonMandatoryElements.

Errors
0001 The property is no longer accessible.

29.3.2.21.13 OptionsDialogAction

Property: OptionsDialogAction as SPYDialogAction[1990]

Description
To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
2200 The object is no longer valid.
2201 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.21.14 Parent

Property: Parent as Dialogs[1809] (read-only)

Description
Access the parent of the object.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.15 RepeatCount

Property: RepeatCount as long

Description
Number of elements to create for repeated types.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.16 SampleValueHints

Property: `SampleValueHints` as `SPYSampleXMLGenerationSampleValueHints` 1996

Description
Specifies how to select data for the generated sample file.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.17 SchemaOrDTDAssignment

Property: SchemaOrDTDAssignment as SPYSampleXMLGenerationSchemaOrDTDAssignment 1996

Description
Specifies in which way a reference to the related schema or DTD - which is this document - will be generated into the sample XML.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.18 TakeFirstChoice - obsolete

Property: TakeFirstChoice as Boolean

Description
Do no longer use this property.

Errors
0001 The property is no longer accessible.

29.3.2.21.19 TryToUseNonAbstractTypes

Property: TryToUseNonAbstractTypes as Boolean

Description
If true, tries to use a non-abstract type for xsi:type, if element has an abstract type.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.22 GridView

Methods
Deselect 1877
Select 1878

SetFocus 1878

Properties
CurrentFocus 1877

IsVisible 1878

Description
GridView Class

29.3.2.22.1 Events

29.3.2.22.1.1 OnBeforeDrag

Event: OnBeforeDrag() as Boolean

XMLSpy scripting environment - VBScript:
```
Function On_BeforeDrag()
        ' On_BeforeStartEditing = False  ' to prohibit dragging
End Function
```

XMLSpy scripting environment - JScript:
```
function On_BeforeDrag()
{
        // return false;  /* to prohibit dragging */
}
```

XMLSpy IDE Plugin:
```
IXMLSpyPlugIn.OnEvent (4, ...)   // nEventId = 4
```

Description
This event gets fired on an attempt to drag an XMLData element on the grid view. Return *false* to prevent dragging the data element to a different position.

29.3.2.22.1.2 OnBeforeDrop

Event: OnBeforeDrop(*objXMLData* as XMLData 1939) as Boolean

XMLSpy scripting environment - VBScript:
```
Function On_BeforeDrop(objXMLData)
```

```
        ' On_BeforeStartEditing = False  ' to prohibit dropping
End Function
```

XMLSpy scripting environment - JScript:

```
function On_BeforeDrop(objXMLData)
{
        // return false;  /* to prohibit dropping */
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (5, ...)   // nEventId = 5
```

Description

This event gets fired on an attempt to drop a previously dragged XMLData element on the grid view. Return *false* to prevent the data element to be moved from its original position to the drop destination position.

29.3.2.22.1.3 OnBeforeStartEditing

Event: OnBeforeStartEditing(*objXMLData* as XMLData[1939], *bEditingName* as Boolean)as Boolean

XMLSpy scripting environment - VBScript:

```
Function On_BeforeStartEditing(objXMLData, bEditingName)
        ' On_BeforeStartEditing = False  ' to prohibit editing the field
End Function
```

XMLSpy scripting environment - JScript:

```
function On_BeforeStartEditing(objXMLData, bEditingName)
{
        // return false;  /* to prohibit editing the field */
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (1, ...)   // nEventId = 1
```

Description

This event gets fired before the editing mode for a grid cell gets entered. If the parameter *bEditingName* is true, the name part of the element will be edited, it its value is false, the value part will be edited.

29.3.2.22.1.4 OnEditingFinished

Event: OnEditingFinished(*objXMLData* as XMLData[1939], *bEditingName* as Boolean)

XMLSpy scripting environment - VBScript:

```
Function On_EditingFinished(objXMLData, bEditingName)
End Function
```


© 2016-2022 Altova GmbH

XMLSpy scripting environment - JScript:
function On_EditingFinished(*objXMLData*, *bEditingName*)
{
}

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (2, ...) // nEventId = 2

Description
This event gets fired when the editing mode of a grid cell is exited. The parameter *bEditingName* specifies if the name part of the element has been edited.

29.3.2.22.1.5 OnFocusChanged

Event: OnFocusChanged(*objXMLData* as XMLData[1939], *bSetFocus* as Boolean, *bEditingName* as Boolean)

XMLSpy scripting environment - VBScript:
Function On_FocusChanged(*objXMLData*, *bSetFocus*, *bEditingName*)
End Function

XMLSpy scripting environment - JScript:
function On_FocusChanged(*objXMLData*, *bSetFocus*, *bEditingName*)
{
}

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (3, ...) // nEventId = 3

Description
This event gets fired whenever a grid cell receives or loses the cursor focus. If the parameter *bEditingName* is *true*, focus of the name part of the grid element has changed. Otherwise, focus of the value part has changed.

29.3.2.22.2 CurrentFocus

Property: CurrentFocus as XMLData[1939]

Description
Holds the XML element with the current focus. This property is read-only.

29.3.2.22.3 Deselect

Method: Deselect(*pData* as XMLData[1939])

Description
Deselects the element pData in the grid view.

29.3.2.22.4 IsVisible

Property: IsVisible as Boolean

Description
True if the grid view is the active view of the document. This property is read-only.

29.3.2.22.5 Select

Method: Select (*pData* as XMLData[1939])

Description
Selects the XML element pData in the grid view.

29.3.2.22.6 SetFocus

Method: SetFocus (*pFocusData* as XMLData[1939])

Description
Sets the focus to the element pFocusData in the grid view.

29.3.2.23 SchemaDocumentationDlg

Properties and Methods

Standard automation properties
Application[1880]
Parent[1886]

Interaction and visibility properties
OutputFile[1885]
OutputFileDialogAction[1886]
OptionsDialogAction[1885]
ShowProgressBar[1889]
ShowResult[1889]

Document generation options and methods
OutputFormat[1886]
UseFixedDesign[1891]
SPSFile[1891]
EmbedDiagrams[1881]
DiagramFormat[1880]
MultipleOutputFiles[1885]
EmbedCSSInHTML[1880]
CreateDiagramsFolder[1880]
GenerateRelativeLinks[1881]

© 2016-2022 Altova GmbH

IncludeAll 1881
IncludeIndex 1883
IncludeGlobalAttributes 1882
IncludeGlobalElements 1882
IncludeLocalAttributes 1883
IncludeLocalElements 1884
IncludeGroups 1883
IncludeComplexTypes 1882
IncludeSimpleTypes 1884
IncludeAttributeGroups 1882
IncludeRedefines 1884
IncludeReferencedSchemas 1884

AllDetails 1879
ShowDiagram 1887
ShowNamespace 1888
ShowType 1890
ShowChildren 1887
ShowUsedBy 1891
ShowProperties 1889
ShowSingleFacets 1890
ShowPatterns 1889
ShowEnumerations 1888
ShowAttributes 1887
ShowIdentityConstraints 1888
ShowAnnotations 1886
ShowSourceCode 1890

Description
This object combines all options for schema document generation as they are available through user interface dialog boxes in XMLSpy. The document generation options are initialized with the values used during the last generation of schema documentation. However, before using the object you have to set the SetOutputFile 1885 property to a valid file path. Use OptionsDialogAction 1885, OutputFileDialogAction 1886 and ShowProgressBar 1889 to specify the level of user interaction desired. You can use IncludeAll 1881 and AllDetails 1879 to set whole option groups at once or the individual properties to operate on a finer granularity.

29.3.2.23.1 AllDetails

Method: AllDetails (i_bDetailsOn as Boolean)

Description
Use this method to turn all details options on or off.

Errors
2900 The object is no longer valid.

29.3.2.23.2 Application

Property: Application as Application[1724] (read-only)

Description
Access the XMLSpy application object.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.3 CreateDiagramsFolder

Property: CreateDiagramsFolder as Boolean

Description
Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is false.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.4 DiagramFormat

Property: DiagramFormat as SPYImageKind[1992]

Description
This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is PNG.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.5 EmbedCSSInHTML

Property: EmbedCSSInHTML as Boolean

Description
Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized

© 2016-2022 Altova GmbH

with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.6 EmbedDiagrams

Property: EmbedDiagrams as Boolean

Description

Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.7 GenerateRelativeLinks

Property: GenerateRelativeLinks as Boolean

Description

Set this property to true, to create relative paths to local files. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is false.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.8 IncludeAll

Method: IncludeAll (i_bInclude as Boolean)

Description

Use this method to mark or unmark all include options.

Errors

2900 The object is no longer valid.

29.3.2.23.9 IncludeAttributeGroups

Property: IncludeAttributeGroups as Boolean

Description
Set this property to true, to include attribute groups in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.10 IncludeComplexTypes

Property: IncludeComplexTypes as Boolean

Description
Set this property to true, to include complex types in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.11 IncludeGlobalAttributes

Property: IncludeGlobalAttributes as Boolean

Description
Set this property to true, to include global attributes in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.12 IncludeGlobalElements

Property: IncludeGlobalElements as Boolean

Description

© 2016-2022 Altova GmbH

Set this property to true, to include global elements in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.13 IncludeGroups

Property: IncludeGroups as Boolean

Description
Set this property to true, to include groups in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.14 IncludeIndex

Property: IncludeIndex as Boolean

Description
Set this property to true, to include an index in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.15 IncludeLocalAttributes

Property: IncludeLocalAttributes as Boolean

Description
Set this property to true, to include local attributes in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.16 IncludeLocalElements

Property: IncludeLocalElements as Boolean

Description
Set this property to true, to include local elements in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.17 IncludeRedefines

Property: IncludeRedefines as Boolean

Description
Set this property to true, to include redefines in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.18 IncludeReferencedSchemas

Property: IncludeReferencedSchemas as Boolean

Description
Set this property to true, to include referenced schemas in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.19 IncludeSimpleTypes

Property: IncludeSimpleTypes as Boolean

Description

© 2016-2022 Altova GmbH

Set this property to true, to include simple types in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.20 MultipleOutputFiles

Property: MultipleOutputFiles as Boolean

Description

Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is false.

Errors

2900 The object is no longer valid.
2901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.23.21 OptionsDialogAction

Property: OptionsDialogAction as SPYDialogAction[1990]

Description

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

2900 The object is no longer valid.
2901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.23.22 OutputFile

Property: OutputFile as String

Description

Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to Document.GenerateSchemaDocumentation[1829].

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.23 OutputFileDialogAction

Property: OutputFileDialogAction as SPYDialogAction[1990]

Description
To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in OutputFile[1885] should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
2900 The object is no longer valid.
2901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.23.24 OutputFormat

Property: OutputFormat as SPYSchemaDocumentationFormat[1997]

Description
Defines the kind of documentation that will be generated: HTML (value=`0`), MS-Word (value=`1`), or RTF (value=`2`). The property gets initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is HTML.

Errors
2900 The object is no longer valid.
2901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.23.25 Parent

Property: Parent as Dialogs[1809] (read-only)

Description
Access the parent of the object.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.26 ShowAnnotations

Property: ShowAnnotations as Boolean

Description

© 2016-2022 Altova GmbH

Set this property to true, to show the annotations to a type definition in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.27 ShowAttributes

Property: ShowAttributes as Boolean

Description

Set this property to true, to show the type definitions attributes in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.28 ShowChildren

Property: ShowChildren as Boolean

Description

Set this property to true, to show the children of a type definition as links in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.29 ShowDiagram

Property: ShowDiagram as Boolean

Description

Set this property to true, to show type definitions as diagrams in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

29.3.2.23.30 ShowEnumerations

Property: ShowEnumerations as Boolean

Description
Set this property to true, to show the enumerations contained in a type definition in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.31 ShowIdentityConstraints

Property: ShowIdentityConstraints as Boolean

Description
Set this property to true, to show a type definitions identity constraints in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.32 ShowNamespace

Property: ShowNamespace as Boolean

Description
Set this property to true, to show the namespace of type definitions in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.23.33 ShowPatterns

Property: ShowPatterns as Boolean

Description
Set this property to true, to show the patterns of a type definition in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.34 ShowProgressBar

Property: ShowProgressBar as Boolean

Description
Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.35 ShowProperties

Property: ShowProperties as Boolean

Description
Set this property to true, to show the type definition properties in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.36 ShowResult

Property: ShowResult as Boolean

Description
Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in XMLSpy. To show Word documentation, MS-Word will be started. The

property gets initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.37 ShowSingleFacets

Property: ShowSingleFacets as Boolean

Description

Set this property to true, to show the facets of a type definition in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.38 ShowSourceCode

Property: ShowSourceCode as Boolean

Description

Set this property to true, to show the XML source code for type definitions in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.39 ShowType

Property: ShowType as Boolean

Description

Set this property to true, to show the type of type definitions in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.23.40 ShowUsedBy

Property: ShowUsedBy as Boolean

Description
Set this property to true, to show the used-by relation for type definitions in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation[1829]. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.41 SPSFile

Property: SPSFile as String

Description
Full path and name of the SPS file that will be used to generate the documentation.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.42 UseFixedDesign

Property: UseFixedDesign as Boolean

Description
Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.24 SpyProject

Methods
CloseProject[1892]
SaveProject[1892]
SaveProjectAs[1893]

Properties

RootItems 1892
ProjectFile 1892

Description
SpyProject Class

29.3.2.24.1 CloseProject

Declaration: CloseProject(*bDiscardChanges* as Boolean, *bCloseFiles* as Boolean, *bDialog* as Boolean)

Parameters
bDiscardChanges
Set bDiscardChanges to FALSE if you want to save the changes of the open project files and the project.

bCloseFiles
Set bCloseFiles to TRUE to close all open project files.

bDialog
Show dialogs for user input.

Description
CloseProject closes the current project.

29.3.2.24.2 ProjectFile

Declaration: ProjectFile as String

Description
Path and filename of the project.

29.3.2.24.3 RootItems

Declaration: RootItems as SpyProjectItems 1895

Description
Root level of collection of project items.

29.3.2.24.4 SaveProject

Declaration: SaveProject

Description
SaveProject saves the current project.

© 2016-2022 Altova GmbH

29.3.2.24.5 SaveProjectAs

Declaration: SaveProjectAs (strPath as String, bDialog as Boolean)

Parameters
strPath
Full path with file name of new project file.

bDialog
If bDialog is TRUE, a file-dialog will be displayed.

Description
SaveProjectAs stores the project data into a new location.

29.3.2.25 SpyProjectItem

Methods
Open 1894

Properties
ChildItems 1893
ParentItem 1894
FileExtensions 1893
ItemType 1894
Name 1894
Path 1894
ValidateWith 1894
XMLForXSLTransformation 1895
XSLForXMLTransformation 1895
XSLTransformationFileExtension 1895
XSLTransformationFolder 1895

Description
SpyProjectItem Class

29.3.2.25.1 ChildItems

Declaration: ChildItems as SpyProjectItems 1895

Description
If the item is a folder, ChildItems is the collection of the folder content.

29.3.2.25.2 FileExtensions

Declaration: FileExtensions as String

Description

Used to set the file extensions if the project item is a folder.

29.3.2.25.3 ItemType

Declaration: ItemType as SPYProjectItemTypes 1994

Description
This property is read-only.

29.3.2.25.4 Name

Declaration: Name as String

Description
Name of the project item. This property is read-only.

29.3.2.25.5 Open

Declaration: Open as Document 1813

Return Value
The project item opened as document.

Description
Opens the project item.

29.3.2.25.6 ParentItem

Declaration: ParentItem as SpyProjectItem 1893

Description
Parent item of the current project item. Can be NULL (Nothing) if the project item is a top-level item.

29.3.2.25.7 Path

Declaration: Path as String

Description
Path of project item. This property is read-only.

29.3.2.25.8 ValidateWith

Declaration: ValidateWith as String

© 2016-2022 Altova GmbH

Description
Used to set the schema/DTD for validation.

29.3.2.25.9 XMLForXSLTransformation

Declaration: XMLForXSLTransformation as String

Description
Used to set the XML for XSL transformation.

29.3.2.25.10 XSLForXMLTransformation

Declaration: XSLForXMLTransformation as String

Description
Used to set the XSL for XML transformation.

29.3.2.25.11 XSLTransformationFileExtension

Declaration: XSLTransformationFileExtension as String

Description
Used to set the file extension for XSL transformation output files.

29.3.2.25.12 XSLTransformationFolder

Declaration: XSLTransformationFolder as String

Description
Used to set the destination folder for XSL transformation output files.

29.3.2.26 SpyProjectItems

Methods
AddFile 1896
AddFolder 1896
AddURL 1896
RemoveItem 1897

Properties
Count 1897
Item 1897

Description
SpyProjectItems Class

29.3.2.26.1 AddFile

Declaration: AddFile (*strPath* as String)

Parameters

strPath
Full path with file name of new project item

Description
The method adds a new file to the collection of project items.

29.3.2.26.2 AddFolder

Declaration: AddFolder (*strName* as String)

Parameters

strName
Name of the new folder.

Description
The method AddFolder adds a folder with the name strName to the collection of project items.

29.3.2.26.3 AddURL

Declaration: AddURL (*strURL* as String, *nURLType* as SPYURLTypes[1998], *strUser* as String, *strPassword* as String, *bSave* as Boolean)

Description

strURL
URL to open as document.

nURLType
Type of document to open. Set to -1 for auto detection.

strUser
Name of the user if required. Can be empty.

strPassword
Password for authentification. Can be empty.

bSave
Save user and password information.

Description
The method adds an URL item to the project collection.

© 2016-2022 Altova GmbH

29.3.2.26.4 Count

Declaration: Count as long

Description
This property gets the count of project items in the collection. The property is read-only.

29.3.2.26.5 Item

Declaration: Item (*n* as long) as SpyProjectItem 1893

Description
Retrieves the n-th element of the collection of project items. The first item has index 1.

29.3.2.26.6 RemoveItem

Declaration: RemoveItem (*pItem* as SpyProjectItem 1893)

Description
RemoveItem deletes the item pItem from the collection of project items.

29.3.2.27 TextImportExportSettings

Properties for import only
ImportFile 1899

Properties for export only
DestinationFolder 1898
FileExtension 1899
CommentIncluded 1898
RemoveDelimiter 1899
RemoveNewline 1899

Properties for import and export
HeaderRow 1899
FieldDelimiter 1898
EnclosingCharacter 1898
Encoding 1898
EncodingByteOrder 1898

Description
TextImportExportSettings contains options common to text import and export functions.

29.3.2.27.1 CommentIncluded

Property: CommentIncluded as Boolean

Description
This property tells whether additional comments are added to the generated text file. Default is true. This property is used only when exporting to text files.

29.3.2.27.2 DestinationFolder

Property: DestinationFolder as String

Description
The property DestinationFolder sets the folder where the created files are saved during text export.

29.3.2.27.3 EnclosingCharacter

Property: EnclosingCharacter as SPYTextEnclosing[1998]

Description
This property defines the character that encloses all field values for import and export. Default is spyNoEnclosing[1998].

29.3.2.27.4 Encoding

Property: Encoding as String

Description
The property Encoding sets the character encoding for the text files for importing and exporting.

29.3.2.27.5 EncodingByteOrder

Property: EncodingByteOrder as SPYEncodingByteOrder[1991]

Description
The property EncodingByteOrder sets the byte order for Unicode characters. Default is spyNONE[1991].

29.3.2.27.6 FieldDelimiter

Property: FieldDelimiter as SPYTextDelimiters[1998]

Description

© 2016-2022 Altova GmbH

The property FieldDelimiter defines the delimiter between the fields during import and export. Default is spyTabulator[1998].

29.3.2.27.7 FileExtension

Property: FileExtension as String

Description
This property sets the file extension for files created on text export.

29.3.2.27.8 HeaderRow

Property: HeaderRow as Boolean

Description
The property HeaderRow is used during import and export. Set HeaderRow true on import, if the first line of the text file contains the names of the columns. Set HeaderRow true on export, if the first line in the created text files should contain the name of the columns. Default value is true.

29.3.2.27.9 ImportFile

Property: ImportFile as String

Description
This property is used to set the text file for import. The string has to be a full qualified path. See also Import and Export[1694].

29.3.2.27.10 RemoveDelimiter

Property: RemoveDelimiter as Boolean

Description
The property RemoveDelimiter defines whether characters in the text that are equal to the delimiter character are removed. Default is false. This property is used only when exporting to text files.

29.3.2.27.11 RemoveNewline

Property: RemoveNewline as Boolean

Description
The property RemoveNewline defines whether newline characters in the text are removed. Default is false. This property is used only when exporting to text files.

29.3.2.28 TextView

Properties and Methods

Application[1901]
Parent[1903]

LineFromPosition[1903]
PositionFromLine[1904]
LineLength[1903]
SelText[1905]
GetRangeText[1902]
ReplaceText[1904]
MoveCaret[1903]
GoToLineChar[1902]
SelectText[1905]
SelectionStart[1904]
SelectionEnd[1904]
Text[1905]
LineCount[1902]
Length[1902]

Description

29.3.2.28.1 Events

29.3.2.28.1.1 OnBeforeShowSuggestions

Event: OnBeforeShowSuggestions() as Boolean

Description
This event gets fired before a suggestion window is shown. The Document[1813] property Suggestions[1840] contains a string array that is recommended to the user. It is possible to modify the displayed recommendations during this event. Before doing so you have to assign an empty array to the Suggestions[1840] property. The best location for this is the OnDocumentOpened[1727] event. To prevent the suggestion window to show up return false and true to continue its display.

Examples
Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:
```
Function On_BeforeShowSuggestions()
End Function
```

XMLSpy scripting environment - JScript:
```
function On_BeforeShowSuggestions()
{
}
```


© 2016-2022 Altova GmbH

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (33, ...) // nEventId = 33

29.3.2.28.1.2 OnChar

Event: OnChar(nChar as Long, bExistSuggestion as Boolean) as Boolean

Description
This event gets fired on each key stroke. The parameter nChar is the key that was pressed and bExistSuggestions tells whether a XMLSpy generated suggestions window is displayed after this key. The Document[1813] property Suggestions[1840] contains a string array that is recommended to the user. It is possible to modify the displayed recommendations during this event. Before doing so you have to assign an empty array to the Suggestions[1840] property. The best location for this is the OnDocumentOpened[1727] event. To prevent the suggestion window to show up return false and true to continue its display.
It is also possible to create a new suggestions window when none is provided by XMLSpy. Set the Document[1813] property Suggestions[1840] to a string array with your recommendations and return true.
This event is fired before the OnBeforeShowSuggestions[1900] event. If you prevent to show the suggestion window by returning false then OnBeforeShowSuggestions[1900] is not fired.

Examples
Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:
```
Function On_Char(nChar, bExistSuggestions)
End Function
```

XMLSpy scripting environment - JScript:
```
function On_Char(nChar, bExistSuggestions)
{
}
```

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (35, ...) // nEventId = 35

29.3.2.28.2 Application

Property: Application as Application[1724] (read-only)

Description
Access the XMLSpy application object.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.3 GetRangeText

Method: GetRangeText(nStart as Long, nEnd as Long) as String

Description
Returns the text in the specified range.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.4 GoToLineChar

Method: GoToLineChar(nLine as Long, nChar as Long)

Description
Moves the caret to the specified line and character position.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.5 Length

Property: Length as Long

Description
Returns the character count of the document.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.6 LineCount

Property: LineCount as Long

Description
Returns the number of lines in the document.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.28.7 LineFromPosition

Method: LineFromPosition(nCharPos as Long) as Long

Description
Returns the line number of the character position.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.8 LineLength

Method: LineLength(nLine as Long) as Long

Description
Returns the length of the line.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.9 MoveCaret

Method: MoveCaret(nDiff as Long)

Description
Moves the caret nDiff characters.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.10 Parent

Property: Parent as Document 1813 (read-only)

Description
Access the parent of the object.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.11 PositionFromLine

Method: PositionFromLine(nLine as Long) as Long

Description
Returns the start position of the line.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.12 ReplaceText

Method: ReplaceText(nPosFrom as Long, nPosTill as Long, sText as String)

Description
Replaces the text in the specified range.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.13 SelectionEnd

Property: SelectionEnd as Long

Description
Returns/sets the text selection end position.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.14 SelectionStart

Property: SelectionStart as Long

Description
Returns/sets the text selection start position.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.28.15 SelectText

Method: SelectText(nPosFrom as Long, nPosTill as Long)

Description
Selects the text in the specified range.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.16 SelText

Property: SelText as String

Description
Returns/sets the selected text.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.17 Text

Property: Text as String

Description
Returns/sets the document text.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29 WSDLDocumentationDlg

Properties and Methods

Standard automation properties
Application 1907
Parent 1912

Interaction and visibility properties
GlobalElementsAndTypesOnly 1908
OptionsDialogAction 1911
OutputFile 1911
OutputFileDialogAction 1912
SeparateSchemaDocument 1912

ShowProgressBar 1914
ShowResult 1915

Document generation options and methods
OutputFormat 1912
UseFixedDesign 1916
SPSFile 1916
EmbedDiagrams 1908
DiagramFormat 1907
MultipleOutputFiles 1911
EmbedCSSInHTML 1908
CreateDiagramsFolder 1907

IncludeAll 1908
IncludeBinding 1909
IncludeImportedWSDLFiles 1909
IncludeMessages 1909
IncludeOverview 1910
IncludePortType 1910
IncludeService 1910
IncludeTypes 1910

AllDetails 1906
ShowBindingDiagram 1913
ShowExtensibility 1913
ShowMessageParts 1913
ShowPort 1914
ShowPortTypeDiagram 1914
ShowPortTypeOperations 1914
ShowServiceDiagram 1915
ShowSourceCode 1915
ShowTypesDiagram 1916
ShowUsedBy 1916

Description
This object combines all options for WSDL document generation as they are available through user interface dialog boxes in XMLSpy. The document generation options are initialized with the values used during the last generation of WSDL documentation. However, before using the object you have to set the OutputFile 1911 property to a valid file path. Use OptionsDialogAction 1911, OutputFileDialogAction 1911 and ShowProgressBar 1914 to specify the level of user interaction desired. You can use IncludeAll 1908 and AllDetails 1906 to set whole option groups at once or the individual properties to operate on a finer granularity.

29.3.2.29.1 AllDetails

Method: AllDetails (i_bDetailsOn as Boolean)

Description
Use this method to turn all details options on or off.

© 2016-2022 Altova GmbH

Errors
4300 The object is no longer valid.

29.3.2.29.2 Application

Property: Application as Application[1724] (read-only)

Description
Access the XMLSpy application object.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.3 CreateDiagramsFolder

Property: CreateDiagramsFolder as Boolean

Description
Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is false.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.4 DiagramFormat

Property: DiagramFormat as SPYImageKind[1992]

Description
This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is PNG.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.5 EmbedCSSInHTML

Property: EmbedCSSInHTML as Boolean

Description
Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.6 EmbedDiagrams

Property: EmbedDiagrams as Boolean

Description
Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.7 GlobalElementsAndTypesOnly

Property: GlobalElementsAndTypesOnly as Boolean

Description
Returns/sets a value indicating whether a full Schema documentation is done or only Global Elements and Types are documented.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.8 IncludeAll

Method: IncludeAll (i_bInclude as Boolean)

Description

© 2016-2022 Altova GmbH

Use this method to mark or unmark all include options.

Errors
4300 The object is no longer valid.

29.3.2.29.9 IncludeBinding

Property: IncludeBinding as Boolean

Description
Set this property to true, to include bindings in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.10 IncludeImportedWSDLFiles

Property: IncludeImportedWSDLFiles as Boolean

Description
Set this property to true, to include imported WSDL files in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.11 IncludeMessages

Property: IncludeMessages as Boolean

Description
Set this property to true, to include messages in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.12 IncludeOverview

Property: IncludeOverview as Boolean

Description
Set this property to true, to include an overview in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.13 IncludePortType

Property: IncludePortType as Boolean

Description
Set this property to true, to include port types in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.14 IncludeService

Property: IncludeService as Boolean

Description
Set this property to true, to include services in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.15 IncludeTypes

Property: IncludeTypes as Boolean

Description
Set this property to true, to include types in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

© 2016-2022 Altova GmbH

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.16 MultipleOutputFiles

Property: MultipleOutputFiles as Boolean

Description

Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is false.

Errors

3900 The object is no longer valid.
3901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.29.17 OptionsDialogAction

Property: OptionsDialogAction as SPYDialogAction[1990]

Description

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

3900 The object is no longer valid.
3901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.29.18 OutputFile

Property: OutputFile as String

Description

Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to Document.GenerateWSDLDocumentation[1830].

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.19 OutputFileDialogAction

Property: OutputFileDialogAction as SPYDialogAction[1990]

Description
To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in OutputFile[1911] should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
3900 The object is no longer valid.
3901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.29.20 OutputFormat

Property: OutputFormat as SPYSchemaDocumentationFormat[1997]

Description
Defines the kind of documentation that will be generated: HTML (value=`0`), MS-Word (value=`1`), or RTF (value=`2`). The property gets initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is HTML.

Errors
3900 The object is no longer valid.
3901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.29.21 Parent

Property: Parent as Dialogs[1809] (read-only)

Description
Access the parent of the object.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.22 SeparateSchemaDocument

Property: SeparateSchemaDocument as Boolean

Description
Returns/sets a value indicating whether the Schema documentation should be placed in a separate document.

© 2016-2022 Altova GmbH

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.23 ShowBindingDiagram

Property: ShowBindingDiagram as Boolean

Description
Set this property to true, to show binding diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.24 ShowExtensibility

Property: ShowExtensibility as Boolean

Description
Set this property to true, to show service and binding extensibilities in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.25 ShowMessageParts

Property: ShowMessageParts as Boolean

Description
Set this property to true, to show message parts of messges in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.26 ShowPort

Property: ShowPort as Boolean

Description
Set this property to true, to show service ports in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.27 ShowPortTypeDiagram

Property: ShowPortTypeDiagram as Boolean

Description
Set this property to true, to show port type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.28 ShowPortTypeOperations

Property: ShowPortTypeOperations as Boolean

Description
Set this property to true, to show port type operations in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.29 ShowProgressBar

Property: ShowProgressBar as Boolean

Description

© 2016-2022 Altova GmbH

Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.30 ShowResult

Property: ShowResult as Boolean

Description

Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in XMLSpy. To show Word documentation, MS-Word will be started. The property gets initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.31 ShowServiceDiagram

Property: ShowServiceDiagram as Boolean

Description

Set this property to true, to show service diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.32 ShowSourceCode

Property: ShowSourceCode as Boolean

Description

Set this property to true, to show source code for the includes in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.33 ShowTypesDiagram

Property: ShowTypesDiagram as Boolean

Description
Set this property to true, to show type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.34 ShowUsedBy

Property: ShowUsedBy as Boolean

Description
Set this property to true, to show the used-by relation for types, bindings and messages definitions in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation[1830]. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.35 UseFixedDesign

Property: UseFixedDesign as Boolean

Description
Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.36 SPSFile

Property: SPSFile as String

Description
Full path and name of the SPS file that will be used to generate the documentation.

© 2016-2022 Altova GmbH

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.30 WSDL20DocumentationDlg

Properties and Methods

Standard automation properties
Application 1918
Parent 1923

Interaction and visibility properties
GlobalElementsAndTypesOnly 1919
OptionsDialogAction 1922
OutputFile 1922
OutputFileDialogAction 1923
SeparateSchemaDocument 1923
ShowProgressBar 1925
ShowResult 1926

Document generation options and methods
OutputFormat 1923
UseFixedDesign 1927
SPSFile 1927
EmbedDiagrams 1919
DiagramFormat 1919
MultipleOutputFiles 1922
EmbedCSSInHTML 1919
CreateDiagramsFolder 1918

IncludeAll 1920
IncludeBinding 1920
IncludeImportedWSDLFiles 1920
IncludeInterface 1921
IncludeOverview 1921
IncludeService 1921
IncludeTypes 1921

AllDetails 1918
ShowBindingDiagram 1924
ShowExtensibility 1924
ShowEndpoint 1924
ShowFault 1925
ShowInterfaceDiagram 1925
ShowOperation 1925
ShowServiceDiagram 1926
ShowSourceCode 1926
ShowTypesDiagram 1927
ShowUsedBy 1927

Description

This object combines all options for WSDL document generation as they are available through user interface dialog boxes in XMLSpy. The document generation options are initialized with the values used during the last generation of WSDL documentation. However, before using the object you have to set the `OutputFile`[1922] property to a valid file path. Use `OptionsDialogAction`[1922], `OutputFileDialogAction`[1923] and `ShowProgressBar`[1925] to specify the level of user interaction desired. You can use `IncludeAll`[1920] and `AllDetails`[1918] to set whole option groups at once or the individual properties to operate on a finer granularity.

29.3.2.30.1 AllDetails

Method: AllDetails (i_bDetailsOn as Boolean)

Description
Use this method to turn all details options on or off.

Errors
4300 The object is no longer valid.

29.3.2.30.2 Application

Property: Application as Application[1724] (read-only)

Description
Access the XMLSpy application object.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.3 CreateDiagramsFolder

Property: CreateDiagramsFolder as Boolean

Description
Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is false.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.30.4 DiagramFormat

Property: DiagramFormat as SPYImageKind[1992]

Description
This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is PNG.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.5 EmbedCSSInHTML

Property: EmbedCSSInHTML as Boolean

Description
Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.6 EmbedDiagrams

Property: EmbedDiagrams as Boolean

Description
Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.7 GlobalElementsAndTypesOnly

Property: GlobalElementsAndTypesOnly as Boolean

Description
Returns/sets a value indicating whether a full Schema documentation is done or only Global Elements and Types are documented.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.30.8 IncludeAll

Method: IncludeAll (i_bInclude as Boolean)

Description
Use this method to mark or unmark all include options.

Errors
4300 The object is no longer valid.

29.3.2.30.9 IncludeBinding

Property: IncludeBinding as Boolean

Description
Set this property to true, to include bindings in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.10 IncludeImportedWSDLFiles

Property: IncludeImportedWSDLFiles as Boolean

Description
Set this property to true, to include imported WSDL files in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.30.11 IncludeInterface

Property: IncludeInterface as Boolean

Description
Set this property to true, to include interfaces in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.12 IncludeOverview

Property: IncludeOverview as Boolean

Description
Set this property to true, to include an overview in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.13 IncludeService

Property: IncludeService as Boolean

Description
Set this property to true, to include services in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.14 IncludeTypes

Property: IncludeTypes as Boolean

Description
Set this property to true, to include types in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.15 MultipleOutputFiles

Property: MultipleOutputFiles as Boolean

Description
Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is false.

Errors
4300 The object is no longer valid.
4301 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.30.16 OptionsDialogAction

Property: OptionsDialogAction as SPYDialogAction[1990]

Description
To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
4300 The object is no longer valid.
4301 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.30.17 OutputFile

Property: OutputFile as String

Description
Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to Document.GenerateWSDL20Documentation[1830].

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.30.18 OutputFileDialogAction

Property: OutputFileDialogAction as SPYDialogAction[1990]

Description
To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in OutputFile[1922] should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
4300 The object is no longer valid.
4301 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.30.19 OutputFormat

Property: OutputFormat as SPYSchemaDocumentationFormat[1997]

Description
Defines the kind of documentation that will be generated: HTML (value=`0`), MS-Word (value=`1`), or RTF (value=`2`). The property gets initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is HTML.

Errors
4300 The object is no longer valid.
4301 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.30.20 Parent

Property: Parent as Dialogs[1809] (read-only)

Description
Access the parent of the object.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.21 SeparateSchemaDocument

Property: SeparateSchemaDocument as Boolean

Description
Returns/sets a value indicating whether the Schema documentation should be placed in a separate document.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.30.22 ShowBindingDiagram

Property: ShowBindingDiagram as Boolean

Description
Set this property to true, to show binding diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.23 ShowEndpoint

Property: ShowEndpoint as Boolean

Description
Set this property to true, to show service endpoints in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.24 ShowExtensibility

Property: ShowExtensibility as Boolean

Description
Set this property to true, to show service and binding extensibilities in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.30.25 ShowFault

Property: ShowFault as Boolean

Description
Set this property to true, to show faults in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.26 ShowInterfaceDiagram

Property: ShowInterfaceDiagram as Boolean

Description
Set this property to true, to show interface diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.27 ShowOperation

Property: ShowOperation as Boolean

Description
Set this property to true, to show interface and binding operations in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.28 ShowProgressBar

Property: ShowProgressBar as Boolean

Description
Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.29 ShowResult

Property: ShowResult as Boolean

Description
Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in XMLSpy. To show Word documentation, MS-Word will be started. The property gets initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.30 ShowServiceDiagram

Property: ShowServiceDiagram as Boolean

Description
Set this property to true, to show service diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.31 ShowSourceCode

Property: ShowSourceCode as Boolean

Description
Set this property to true, to show source code for the includes in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.30.32 ShowTypesDiagram

Property: ShowTypesDiagram as Boolean

Description
Set this property to true, to show type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.33 ShowUsedBy

Property: ShowUsedBy as Boolean

Description
Set this property to true, to show the used-by relation for types, bindings and messages definitions in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation[1830]. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.34 SPSFile

Property: SPSFile as String

Description
Full path and name of the SPS file that will be used to generate the documentation.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.35 UseFixedDesign

Property: UseFixedDesign as Boolean

Description
Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

Errors

4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.31 XBRLDocumentationDlg

Properties and Methods

Standard automation properties
Application 1929
Parent 1934

Interaction and visibility properties
OptionsDialogAction 1932
OutputFile 1933
OutputFileDialogAction 1933
ShowProgressBar 1937
ShowResult 1937

Document generation options and methods
OutputFormat 1933
UseFixedDesign 1938
SPSFile 1938
EmbedDiagrams 1930
DiagramFormat 1930
EmbedCSSInHTML 1930
CreateDiagramsFolder 1929

IncludeAll 1931
IncludeOverview 1932
IncludeNamespacePrefixes 1932
IncludeGlobalElements 1931
IncludeDefinitionLinkroles 1931
IncludePresentationLinkroles 1932
IncludeCalculationLinkroles 1931

AllDetails 1929
ShowDiagram 1935
ShowSubstitutiongroup 1938
ShowItemtype 1935
ShowBalance 1935
ShowPeriod 1937
ShowAbstract 1934
ShowNillable 1936
ShowLabels 1936
ShowReferences 1937
ShowLinkbaseReferences 1936

ShortQualifiedName 1934

© 2016-2022 Altova GmbH

ShowImportedElements 1935

Description
This object combines all options for XBRL document generation as they are available through user interface dialog boxes in XMLSpy. The document generation options are initialized with the values used during the last generation of XBRL documentation. However, before using the object you have to set the OutputFile 1933 property to a valid file path. Use OptionsDialogAction 1932, OutputFileDialogAction 1933 and ShowProgressBar 1937 to specify the level of user interaction desired. You can use IncludeAll 1931 and AllDetails 1929 to set whole option groups at once or the individual properties to operate on a finer granularity.

29.3.2.31.1 AllDetails

Method: AllDetails (i_bDetailsOn as Boolean)

Description
Use this method to turn all details options on or off.

Errors
4400 The object is no longer valid.

29.3.2.31.2 Application

Property: Application as Application 1724 (read-only)

Description
Access the XMLSpy application object.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.3 CreateDiagramsFolder

Property: CreateDiagramsFolder as Boolean

Description
Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation 1831. The default for the first run is false.

Errors
4400 The object is no longer valid.

4401 Invalid address for the return parameter was specified.

29.3.2.31.4 DiagramFormat

Property: DiagramFormat as SPYImageKind[1992]

Description
This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is PNG.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.5 EmbedCSSInHTML

Property: EmbedCSSInHTML as Boolean

Description
Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.6 EmbedDiagrams

Property: EmbedDiagrams as Boolean

Description
Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.31.7 IncludeAll

Method: IncludeAll (i_bInclude as Boolean)

Description
Use this method to mark or unmark all include options.

Errors
4400 The object is no longer valid.

29.3.2.31.8 IncludeCalculationLinkroles

Property: IncludeCalculationLinkroles as Boolean

Description
Set this property to true, to include calculation linkroles in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.9 IncludeDefinitionLinkroles

Property: IncludeDefinitionLinkroles as Boolean

Description
Set this property to true, to include definition linkroles in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.10 IncludeGlobalElements

Property: IncludeGlobalElements as Boolean

Description
Set this property to true, to include global elements in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.11 IncludeNamespacePrefixes

Property: IncludeNamespacePrefixes as Boolean

Description
Set this property to true, to include namespace prefixes in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.12 IncludeOverview

Property: IncludeOverview as Boolean

Description
Set this property to true, to include an overview in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.13 IncludePresentationLinkroles

Property: IncludePresentationLinkroles as Boolean

Description
Set this property to true, to include presentation linkroles in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.14 OptionsDialogAction

Property: OptionsDialogAction as SPYDialogAction[1990]

© 2016-2022 Altova GmbH

Description
To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
4400 The object is no longer valid.
4401 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.31.15 OutputFile

Property: OutputFile as String

Description
Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to Document.GenerateXBRLDocumentation[1831].

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.16 OutputFileDialogAction

Property: OutputFileDialogAction as SPYDialogAction[1990]

Description
To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in OutputFile[1933] should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
4400 The object is no longer valid.
4401 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.31.17 OutputFormat

Property: OutputFormat as SPYSchemaDocumentationFormat[1997]

Description
Defines the kind of documentation that will be generated: HTML (value=`0`), MS-Word (value=`1`), or RTF (value=`2`). The property gets initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is HTML.

Errors

4400 The object is no longer valid.
4401 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.31.18 Parent

Property: Parent as Dialogs[1809] (read-only)

Description
Access the parent of the object.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.19 ShortQualifiedName

Property: ShortQualifiedName as Boolean

Description
Set this property to true, to use short qualified names in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.20 ShowAbstract

Property: ShowAbstract as Boolean

Description
Set this property to true, to show abstracts in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.31.21 ShowBalance

Property: ShowBalance as Boolean

Description
Set this property to true, to show balances in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.22 ShowDiagram

Property: ShowDiagram as Boolean

Description
Set this property to true, to show diagrams in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.23 ShowImportedElements

Property: ShowImportedElements as Boolean

Description
Set this property to true, to show imported elements in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.24 ShowItemtype

Property: ShowItemtype as Boolean

Description

Set this property to true, to show item types in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.25 ShowLabels

Property: ShowLabels as Boolean

Description
Set this property to true, to show labels in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.26 ShowLinkbaseReferences

Property: ShowLinkbaseReferences as Boolean

Description
Set this property to true, to show linkbase references in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.27 ShowNillable

Property: ShowNillable as Boolean

Description
Set this property to true, to show nillable properties in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.31.28 ShowPeriod

Property: ShowPeriod as Boolean

Description
Set this property to true, to show periods in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.29 ShowProgressBar

Property: ShowProgressBar as Boolean

Description
Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.30 ShowReferences

Property: ShowReferences as Boolean

Description
Set this property to true, to show references in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.31 ShowResult

Property: ShowResult as Boolean

Description

Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in XMLSpy. To show Word documentation, MS-Word will be started. The property gets initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.32 ShowSubstitutiongroup

Property: ShowSubstitutiongroup as Boolean

Description
Set this property to true, to show substitution groups in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation[1831]. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.33 SPSFile

Property: SPSFile as String

Description
Full path and name of the SPS file that will be used to generate the documentation.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.34 UseFixedDesign

Property: UseFixedDesign as Boolean

Description
Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.32 XMLData

Properties
Kind 1949
Name 1949
TextValue 1950

HasChildren 1947
MayHaveChildren 1949
Parent 1949

Methods
GetFirstChild 1944
GetNextChild 1945
GetCurrentChild 1944

InsertChild 1947
InsertChildAfter 1948
InsertChildBefore 1948
AppendChild 1940

EraseAllChildren 1941
EraseChild 1941
EraseCurrentChild 1942

IsSameNode 1948

CountChildren 1940
CountChildrenKind 1941

GetChild 1943
GetChildAttribute 1943
GetChildElement 1943
GetChildKind 1944
GetNamespacePrefixForURI 1945

HasChildrenKind 1947
SetTextValueXMLEncoded 1950

Description
The XMLData interface provides direct XML-level access to a document. You can read and directly modify the XML representation of the document. However, please, note the following restrictions:

- The XMLData representation is only valid when the document is shown in grid view or authentic view.
- When in authentic view, additional XMLData elements are automatically inserted as parents of each visible document element. Typically this is an XMLData of kind spyXMLDataElement with the Name 1949 property set to 'Text'.
- When you use the XMLData interface while in a different view mode you will not receive errors, but changes are not reflected to the view and might get lost during the next view switch.

Note also:

- Setting a new text value for an XML element is possible if the element does not have non-text children. A text value can be set even if the element has attributes.
- When setting a new text value for an XML element which has more than one text child, the latter will be deleted and replaced by one new text child.
- When reading the text value of an XML element which has more than one text child, only the value of the first text child will be returned.

Objects of this class represent the different atomic parts of an XML document. See the enumeration type SPYXMLDataKind[2000] for the available part types. Each part knows its children, thus forming a XMLData tree with Document.RootElement[1836] at its top. To get the top element of the document content - ignoring the XML header - use Document.DataRoot[1823]. For an examples on how to traverse the XMLData tree, see GetNextChild[1945].

29.3.2.32.1 AppendChild

Declaration: AppendChild (*pNewData* as XMLData[1939])

Description
AppendChild appends pNewData as last child to the XMLData object.

Errors

1500	The XMLData object is no longer valid.
1505	Invalid XMLData kind was specified.
1506	Invalid address for the return parameter was specified.
1507	Element cannot have Children
1512	Cyclic insertion - new data element is already part of document
1514	Invalid XMLData kind was specified for this position.
1900	Document must not be modified

Example

```
Dim objCurrentParent As XMLData
Dim objNewChild As XMLData

Set objNewChild = objSpy.ActiveDocument.CreateChild(spyXMLDataElement)
Set objCurrentParent = objSpy.ActiveDocument.RootElement

objCurrentParent.AppendChild objNewChild

Set objNewChild = Nothing
```

29.3.2.32.2 CountChildren

Declaration: CountChildren as long

Description
CountChildren gets the number of children.

© 2016-2022 Altova GmbH

Available with TypeLibrary version 1.5

Errors
1500 The XMLData object is no longer valid.

29.3.2.32.3 CountChildrenKind

Declaration: CountChildrenKind (*nKind* as SPYXMLDataKind[2000]) as long

Description
CountChildrenKind gets the number of children of the specific kind.

Available with TypeLibrary version 1.5

Errors
1500 The XMLData object is no longer valid.

29.3.2.32.4 EraseAllChildren

Declaration: EraseAllChildren

Description
EraseAllChildren deletes all associated children of the XMLData object.

Errors
1500 The XMLData object is no longer valid.
1900 Document must not be modified

Example
The sample erases all elements of the active document.

```
Dim objCurrentParent As XMLData

Set objCurrentParent = objSpy.ActiveDocument.RootElement
objCurrentParent.EraseAllChildren
```

29.3.2.32.5 EraseChild

Method: EraseChild (Child as XMLData[1939])

Description
Deletes the given child node.

Errors

1500 Invalid object.
1506 Invalid input xml
1510 Invalid parameter.

29.3.2.32.6 EraseCurrentChild

Declaration: EraseCurrentChild

Description
EraseCurrentChild deletes the current XMLData child object. Before you call EraseCurrentChild you must initialize an internal iterator with XMLData.GetFirstChild 1944. After deleting the current child, EraseCurrentChild increments the internal iterator of the XMLData element. No error is returned when the last child gets erased and the iterator is moved past the end of the child list. The next call to EraseCurrentChild however, will return error 1503.

Errors
1500 The XMLData object is no longer valid.
1503 No iterator is initialized for this XMLData object, or the iterator points past the last child.
1900 Document must not be modified

Examples

```
// ------------------------------------
// XMLSpy scripting environment - JScript
// erase all children of XMLData
// ------------------------------------
// let's get an XMLData element, we assume that the
// cursor selects the parent of a list in grid view
var objList = Application.ActiveDocument.GridView.CurrentFocus;

// the following line would be shorter, of course
//        objList.EraseAllChildren ();

// but we want to demonstrate the usage of EraseCurrentChild
if ((objList != null) && (objList.HasChildren))
{
        try
        {
                objEle = objList.GetFirstChild(-1);
                while (objEle != null)
                        objList.EraseCurrentChild();
                        // no need to call GetNextChild
        }
        catch (err)
                // 1503 - we reached end of child list
                { if ((err.number & 0xffff) != 1503) throw (err); }
}
```


© 2016-2022 Altova GmbH

29.3.2.32.7 GetChild

Declaration: GetChild (*position* as long) as XMLData[1939]

Return Value
Returns an XML element as XMLData object.

Description
GetChild() returns a reference to the child at the given index (zero-based).

Available with TypeLibrary version 1.5

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.8 GetChildAttribute

Method: GetChildAttribute (strName as string) child as XMLData object (`NULL` on error)

Description
Retrieves the attribute having the given name.

Errors
1500 Invalid object.
1510 Invalid parameter.

29.3.2.32.9 GetChildElement

Method: GetChildElement (strName as string, nIndex as long) child as XMLData object (`NULL` on error)

Description
Retrieves the Nth child element with the given name.

Errors
1500 Invalid object.
1510 Invalid parameter.

29.3.2.32.10 GetChildKind

Declaration: GetChildKind (*position* as long, *nKind* as SPYXMLDataKind[2000]) as XMLData[1939]

Return Value
Returns an XML element as XMLData object.

Description
GetChildKind() returns a reference to a child of this kind at the given index (zero-based). The position parameter is relative to the number of children of the specified kind and not to all children of the object.

Available with TypeLibrary version 1.5

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.11 GetCurrentChild

Declaration: GetCurrentChild as XMLData[1939]

Return Value
Returns an XML element as XMLData object.

Description
GetCurrentChild gets the current child. Before you call GetCurrentChild you must initialize an internal iterator with XMLData.GetFirstChild[1944].

Errors
1500 The XMLData object is no longer valid.
1503 No iterator is initialized for this XMLData object.
1510 Invalid address for the return parameter was specified.

29.3.2.32.12 GetFirstChild

Declaration: GetFirstChild (*nKind* as SPYXMLDataKind[2000]) as XMLData[1939]

Return Value
Returns an XML element as XMLData object.

Description
GetFirstChild initializes a new iterator and returns the first child. Set nKind = -1 to get an iterator for all kinds of children.
REMARK: The iterator is stored inside the XMLData object and gets destroyed when the XMLData object gets destroyed. Be sure to keep a reference to this object as long as you want to use GetCurrentChild[1944], GetNextChild[1945] or EraseCurrentChild[1942].

© 2016-2022 Altova GmbH

Errors
1500 The XMLData object is no longer valid.
1501 Invalid XMLData kind was specified.
1504 Element has no children of specified kind.
1510 Invalid address for the return parameter was specified.

Example
See the example at XMLData.GetNextChild[1945].

29.3.2.32.13 GetNamespacePrefixForURI

Method: GetNamespacePrefixForURI (strURI as string) strNS as string

Description
Returns the namespace prefix of the supplied URI.

Errors
1500 Invalid object.
1510 Invalid parameter.

29.3.2.32.14 GetNextChild

Declaration: GetNextChild as XMLData[1939]

Return Value
Returns an XML element as XMLData object.

Description
GetNextChild steps to the next child of this element. Before you call GetNextChild you must initialize an internal iterator with XMLData.GetFirstChild[1944].

Check for the last child of the element as shown in the sample below.

Errors
1500 The XMLData object is no longer valid.
1503 No iterator is initialized for this XMLData object.
1510 Invalid address for the return parameter was specified.

Examples
```
' ----------------------------------------------
' VBA code snippet - iterate XMLData children
' ----------------------------------------------
On Error Resume Next
Set objParent = objSpy.ActiveDocument.RootElement

'get elements of all kinds
```

```
Set objCurrentChild = objParent.GetFirstChild(-1)

Do
        'do something useful with the child

        'step to next child
        Set objCurrentChild = objParent.GetNextChild
Loop Until (Err.Number - vbObjectError = 1503)
```

```
// ---------------------------------------
// XMLSpy scripting environment - JScript
// iterate through children of XMLData
// ---------------------------------------
try
{
        var objXMLData = ... // initialize somehow
        var objChild = objXMLData.GetFirstChild(-1);

        while (true)
        {
                // do something usefull with objChild

                objChild = objXMLData.GetNextChild();
        }
}
catch (err)
{
        if ((err.number & 0xffff) == 1504)
                ; // element has no children
        else if ((err.number & 0xffff) == 1503)
                ;       // last child reached
        else
                throw (err);
}
```

29.3.2.32.15 GetTextValueXMLDecoded

Method: GetTextValueXMLDecoded ()as string

Description
Gets the decoded text value of the XML.

Errors
1500 Invalid object.
1510 Invalid parameter.

© 2016-2022 Altova GmbH

29.3.2.32.16 HasChildren

Declaration: HasChildren as Boolean

Description
The property is true if the object is the parent of other XMLData objects. This property is read-only.

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.17 HasChildrenKind

Declaration: HasChildrenKind (*nKind* as SPYXMLDataKind[2000]) as Boolean

Description
The method returns true if the object is the parent of other XMLData objects of the specific kind.

Available with TypeLibrary version 1.5

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.18 InsertChild

Declaration: InsertChild (*pNewData* as XMLData[1939])

Description
InsertChild inserts the new child before the current child (see also XMLData.GetFirstChild[1944], XMLData.GetNextChild[1945] to set the current child).

Errors
1500 The XMLData object is no longer valid.
1503 No iterator is initialized for this XMLData object.
1505 Invalid XMLData kind was specified.
1506 Invalid address for the return parameter was specified.
1507 Element cannot have Children
1512 Cyclic insertion - new data element is already part of document
1514 Invalid XMLData kind was specified for this position.
1900 Document must not be modified

29.3.2.32.19 InsertChildAfter

Method: InsertChildAfter (Node as XMLData, NewData as XMLData)

Description
Inserts a new XML node (supplied with the second parameter) after the specified node (first parameter).

Errors
1500 Invalid object.
1506 Invalid input xml
1507 No children allowed
1510 Invalid parameter.
1512 Child is already added
1514 Invalid kind at position

29.3.2.32.20 InsertChildBefore

Method: InsertChildBefore (Node as XMLData, NewData as XMLData)

Description
Inserts a new XML node (supplied with the second parameter) before the specified node (first parameter).

Errors
1500 Invalid object.
1506 Invalid input xml
1507 No children allowed
1510 Invalid parameter.
1512 Child is already added
1514 Invalid kind at position

29.3.2.32.21 IsSameNode

Declaration: IsSameNode (*pNodeToCompare* as XMLData[1939]) as Boolean

Description

Returns true if pNodeToCompare references the same node as the object itself.

Errors
1500 The XMLData object is no longer valid.
1506 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.32.22 Kind

Declaration: Kind as SPYXMLDataKind 2000

Description
Kind of this XMLData object. This property is read-only.

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.23 MayHaveChildren

Declaration: MayHaveChildren as Boolean

Description
Indicates whether it is allowed to add children to this XMLData object.
This property is read-only.

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.24 Name

Declaration: Name as String

Description
Used to modify and to get the name of the XMLData object.

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.25 Parent

Declaration: Parent as XMLData 1939

Return value
Parent as XMLData object. Nothing (or NULL) if there is no parent element.

Description
Parent of this element. This property is read-only.

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.26 SetTextValueXMLEncoded

Method: SetTextValueXMLEncoded (*strVal* as String[2000])

Description
Sets the encoded text value of the XML.

Errors
1500 Invalid object.
1513 Modification not allowed.

29.3.2.32.27 TextValue

Declaration: TextValue as String

Description
Used to modify and to get the text value of this XMLData object.

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.3 Interfaces (obsolete)

Interfaces contained in this book are obsolete. It is recommended to migrate your applications to the new interfaces. See the different properties and methods in this book for migration hints.

29.3.3.1 AuthenticEvent (obsolete)

Superseded by AuthenticView[1778] and AuthenticRange[1750]

The DocEditView object is renamed to OldAuthenticView.
DocEditSelection is renamed to AuthenticSelection.
DocEditEvent is renamed to AuthenticEvent.
DocEditDataTransfer is renamed to AuthenticDataTransfer.

Their usage - except for AuthenticDataTransfer - is no longer recommended. We will continue to support existing functionality for a yet undefined period of time but no new features will be added to these interface. All functionality available up to now in DocEditView[1965], DocEditSelection[1963], DocEditEvent[1950] and DocEditDataTransfer[1746] is now available via AuthenticView[1778], AuthenticRange[1750] and AuthenticDataTransfer[1745]. Many new features have been added.

© 2016-2022 Altova GmbH

For examples on migrating from DocEdit to Authentic see the description of the different methods and properties of the different DocEdit objects.

Properties

altKey 1951
altLeft 1952
ctrlKey 1956
ctrlLeft 1957
shiftKey 1960
shiftLeft 1961

keyCode 1959
repeat 1960

button 1953

clientX 1955
clientY 1955

dataTransfer 1957

srcElement 1962
fromElement 1958

propertyName 1959

cancelBubble 1954
returnValue 1960

type 1962

Description
DocEditEvent interface.

29.3.3.1.1 altKey (obsolete)

Superseded by parameters to
AuthenticView.OnKeyboardEvent 1784 (On_AuthenticView_KeyPressed)
AuthenticView.OnMouseEvent 1785 (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver 1783 (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period

of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditKeyPressed ()
// {
//     if (Application.ActiveDocument.DocEditView.event.altKey ||
//         Application.ActiveDocument.DocEditView.event.altLeft)
//         MsgBox ("alt key is down");
// }
// use now:
function On_AuthenticView_KeyPressed (SPYKeyEvent i_eKeyEvent, long i_nKeyCode,
SPYVirtualKeyMask i_nVirtualKeyStatus)
{
    if (i_nVirtualKeyStatus & spyAltKeyMask)
        MsgBox ("alt key is down");
}
```

Declaration: altKey as Boolean

Description
True if the right ALT key is pressed.

29.3.3.1.2 altLeft (obsolete)

Superseded by parameters to
AuthenticView.OnKeyboardEvent[1784] (On_AuthenticView_KeyPressed)
AuthenticView.OnMouseEvent[1785] (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver[1783] (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditKeyDown ()
// {
//     if (Application.ActiveDocument.DocEditView.event.altKey ||
//         Application.ActiveDocument.DocEditView.event.altLeft)
//         MsgBox ("alt key is down");
// }
// use now:
function On_AuthenticView_KeyDown (SPYKeyEvent i_eKeyEvent, long i_nKeyCode, SPYVirtualKeyMask
i_nVirtualKeyStatus)
```


© 2016-2022 Altova GmbH

```
{
   if (i_nVirtualKeyStatus & spyAltKeyMask)
      MsgBox ("alt key is down");
}
```

Declaration: altLeft as Boolean

Description
True if the left ALT key is pressed.

29.3.3.1.3 button (obsolete)

Superseded by parameters to
AuthenticView.OnMouseEvent[1785] (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver[1783] (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditButtonDown ()
// {
//    if (Application.ActiveDocument.DocEditView.event.button == 1)
//       MsgBox ("left mouse button down detected");
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos, SPYMouseEvent i_eMouseEvent,
IAuthenticRange *i_ipRange)
{
   if (i_eMouseEvent & spyLeftButtonDownMask)
      MsgBox ("left mouse button down detected");
}
```

Declaration: button as long

Description
Specifies which mouse button is pressed:

0	No button is pressed.
1	Left button is pressed.
2	Right button is pressed.
3	Left and right buttons are both pressed.
4	Middle button is pressed.
5	Left and middle buttons both are pressed.

6 Right and middle buttons are both pressed.
7 All three buttons are pressed.

29.3.3.1.4 cancelBubble (obsolete)

Superseded by the boolean return value of following event handler functions
AuthenticView.OnKeyboardEvent[1784] (On_AuthenticView_KeyPressed)
AuthenticView.OnMouseEvent[1785] (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver[1783] (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

Returning *true* from an event handler function signals that the event has beend handled and normal event handling should be aborted.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditKeyPressed ()
// {
//     if (Application.ActiveDocument.DocEditView.event.keyCode == 0x20)
//     {
//         // cancel key processing, swallow spaces :-)
//         Application.ActiveDocument.DocEditView.event.cancelBubble = true;
//     }
// }
// use now:
function On_AuthenticView_KeyPressed (SPYKeyEvent i_eKeyEvent, long i_nKeyCode,
SPYVirtualKeyMask i_nVirtualKeyStatus)
{
    if (i_nKeyCode == 0x20)
        return true;  // cancel key processing, swallow spaces :-)
}
```

Declaration: cancelBubble as Boolean

Description
Set cancelBubble to TRUE if the default event handler should not be called.

© 2016-2022 Altova GmbH

29.3.3.1.5 clientX (obsolete)

Superseded by parameters to
AuthenticView.OnMouseEvent[1785] (On_AuthenticView_MouseEvent)
AuthenticView.OnBeforeDrop[1781] (On_AuthenticView_BeforeDrop)
AuthenticView.OnDragOver[1783] (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditMouseMove ()
// {
//     MsgBox ("moving over " + Application.ActiveDocument.DocEditView.event.clientX +
//             "/" + Application.ActiveDocument.DocEditView.event.clientY);
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos, SPYMouseEvent i_eMouseEvent,
IAuthenticRange *i_ipRange)
{
    if (i_eMouseEvent & spyMouseMoveMask)
        MsgBox ("moving over " + i_nXPos + "/" + i_nYPos);
}
```

Declaration: clientX as long

Description
X value of the current mouse position in client coordinates.

29.3.3.1.6 clientY (obsolete)

Superseded by parameters to
AuthenticView.OnMouseEvent[1785] (On_AuthenticView_MouseEvent)
AuthenticView.OnBeforeDrop[1781] (On_AuthenticView_BeforeDrop)
AuthenticView.OnDragOver[1783] (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
```

```
// instead of:
// function On_DocEditMouseMove ()
// {
//     MsgBox ("moving over " + Application.ActiveDocument.DocEditView.event.clientX +
//             "/" + Application.ActiveDocument.DocEditView.event.clientY);
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos, SPYMouseEvent i_eMouseEvent,
IAuthenticRange *i_ipRange)
{
    if (i_eMouseEvent & spyMouseMoveMask)
        MsgBox ("moving over " + i_nXPos + "/" + i_nYPos);
}
```

Declaration: clientY as long

Description
Y value of the current mouse position in client coordinates.

29.3.3.1.7 ctrlKey (obsolete)

Superseded by parameters to
AuthenticView.OnKeyboardEvent[1784] (On_AuthenticView_KeyPressed)
AuthenticView.OnMouseEvent[1785] (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver[1783] (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditMouseMove ()
// {
//     if (Application.ActiveDocument.DocEditView.event.ctrlKey ||
//         Application.ActiveDocument.DocEditView.event.altLeft)
//       MsgBox ("control key is down");
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos, SPYMouseEvent i_eMouseEvent,
IAuthenticRange *i_ipRange)
{
    if (i_eMouseEvent & spyCtrlKeyMask)
        MsgBox ("control key is down");
}
```


© 2016-2022 Altova GmbH

Declaration: ctrlKey as Boolean

Description
True if the right CTRL key is pressed.

29.3.3.1.8 ctrlLeft (obsolete)

Superseded by parameters to
AuthenticView.OnKeyboardEvent[1784] (On_AuthenticView_KeyPressed)
AuthenticView.OnMouseEvent[1785] (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver[1783] (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditMouseMove ()
// {
//     if (Application.ActiveDocument.DocEditView.event.ctrlKey ||
//         Application.ActiveDocument.DocEditView.event.altLeft)
//       MsgBox ("control key is down");
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos, SPYMouseEvent i_eMouseEvent,
IAuthenticRange *i_ipRange)
{
   if (i_eMouseEvent & spyCtrlKeyMask)
      MsgBox ("control key is down");
}
```

Declaration: ctrlLeft as Boolean

Description
True if the left CTRL key is pressed.

29.3.3.1.9 dataTransfer (obsolete)

Superseded by parameters to
AuthenticView.OnBeforeDrop[1781] (On_AuthenticView_BeforeDrop)

AuthenticView.OnDragOver[1783] (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditDrop ()
// {
//     if (Application.ActiveDocument.DocEditView.event.dataTransfer != null)
//         if (! Application.ActiveDocument.DocEditView.event.dataTransfer.ownDrag)
//         {
//            // cancel key processing, don't drop foreign objects :-)
//            Application.ActiveDocument.DocEditView.event.cancelBubble = true;
//         }
// }
// use now:
function On_AuthenticView_BeforeDrop (long i_nXPos, long i_nYPos,
                                      IAuthenticRange *i_ipRange,
                                      IAuthenticDataTransfer *i_ipData)
{
    if (i_ipRange != null)
        if (! i_ipRange.ownDrag)
            return true;  // cancel key processing, don't drop foreign objects :-)

    return false;
}
```

Declaration: dataTransfer as Variant

Description
Property dataTransfer.

29.3.3.1.10 fromElement (obsolete)

Not supported

Declaration: fromElement as Variant (not supported)

Description
Currently no event sets this property.

© 2016-2022 Altova GmbH

29.3.3.1.11 keyCode (obsolete)

Superseded by a parameter to AuthenticView.OnKeyboardEvent[1784] (On_AuthenticView_KeyPressed)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditKeyPressed ()
// {
//     if (Application.ActiveDocument.DocEditView.event.keyCode == 0x20)
//     {
//         // cancel key processing, swallow spaces :-)
//         Application.ActiveDocument.DocEditView.event.cancelBubble = true;
//     }
// }
// use now:
function On_AuthenticView_KeyPressed (SPYKeyEvent i_eKeyEvent, long i_nKeyCode,
SPYVirtualKeyMask i_nVirtualKeyStatus)
{
    if (i_nKeyCode == 0x20)
        return true;  // cancel key processing, swallow spaces :-)
}
```

Declaration: keyCode as long

Description
Keycode of the currently pressed key. This property is read-write.

29.3.3.1.12 propertyName (obsolete)

Not supported

Declaration: propertyName as String (not supported)

Description
Currently no event sets this property.

29.3.3.1.13 repeat (obsolete)

Not supported

Declaration: repeat as Boolean (not supported)

Description
True if the onkeydown event is repeated.

29.3.3.1.14 returnValue (obsolete)

No longer supported

Declaration: returnValue as Variant

Description
Use returnValue to set a return value for your event handler.

29.3.3.1.15 shiftKey (obsolete)

Superseded by parameters to
AuthenticView.OnKeyboardEvent[1784] (On_AuthenticView_KeyPressed)
AuthenticView.OnMouseEvent[1785] (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver[1783] (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditDragOver ()
// {
//     if (Application.ActiveDocument.DocEditView.event.shiftKey ||
//         Application.ActiveDocument.DocEditView.event.shiftLeft)
//       MsgBox ("shift key is down");
// }
// use now:
function On_AuthenticView_DragOver (long i_nXPos, long i_nYPos,
```


© 2016-2022 Altova GmbH

```
                                    SPYMouseEvent i_eMouseEvent,
                                    IAuthenticRange *i_ipRange,
                                    IAuthenticDataTransfer *i_ipData)
{
   if (i_eMouseEvent & spyShiftKeyMask)
      MsgBox ("shift key is down");
}
```

Declaration: shiftKey as Boolean

Description
True if the right SHIFT key is pressed.

29.3.3.1.16 shiftLeft (obsolete)

Superseded by parameters to
AuthenticView.OnKeyboardEvent[1784] (On_AuthenticView_KeyPressed)
AuthenticView.OnMouseEvent[1785] (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver[1783] (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditDragOver ()
// {
//    if (Application.ActiveDocument.DocEditView.event.shiftKey ||
//        Application.ActiveDocument.DocEditView.event.shiftLeft)
//       MsgBox ("shift key is down");
// }
// use now:
function On_AuthenticView_DragOver (long i_nXPos, long i_nYPos,
                                    SPYMouseEvent i_eMouseEvent,
                                    IAuthenticRange *i_ipRange,
                                    IAuthenticDataTransfer *i_ipData)
{
   if (i_eMouseEvent & spyShiftKeyMask)
      MsgBox ("shift key is down");
}
```

Declaration: shiftLeft as Boolean

Description
True if the left SHIFT key is pressed.

29.3.3.1.17 srcElement (obsolete)

Superseded by parameters to
AuthenticView.OnMouseEvent[1785] (On_AuthenticView_MouseEvent)
AuthenticView.OnBeforeDrop[1781] (On_AuthenticView_BeforeDrop)
AuthenticView.OnDragOver[1783] (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

With the new event handler function, a range object selecting this element is provided instead of the XMLData element currently below the mouse cursor.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditMouseMove ()
// {
//     var objEvent = Application.ActiveDocument.DocEditView.event;
//     if (objEvent.srcElement != null)
//         MsgBox ("moving over " + objEvent.srcElement.Parent.Name);
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos, SPYMouseEvent i_eMouseEvent,
IAuthenticRange *i_ipRange)
{
   if ((i_eMouseEvent & spyMouseMoveMask) &&
       (i_ipRange != null))
       MsgBox ("moving over " + i_ipRange.FirstXMLData.Parent.Name);
}
```

Declaration: srcElement as Variant

Description
Element which fires the current event. This is usually an XMLData[1939] object.

29.3.3.1.18 type (obsolete)

Not supported

© 2016-2022 Altova GmbH

Declaration: type as String (not supported)

Description
Currently no event sets this property.

29.3.3.2 AuthenticSelection (obsolete)

Superseded by AuthenticRange 1750

The DocEditView object is renamed to OldAuthenticView.
DocEditSelection is renamed to AuthenticSelection.
DocEditEvent is renamed to AuthenticEvent.
DocEditDataTransfer is renamed to AuthenticDataTransfer.

Their usage - except for AuthenticDataTransfer - is no longer recommended. We will continue to support existing functionality for a yet undefined period of time but no new features will be added to these interface. All functionality available up to now in DocEditView 1965, DocEditSelection 1963, DocEditEvent 1950 and DocEditDataTransfer 1745 is now available via AuthenticView 1778, AuthenticRange 1750 and AuthenticDataTransfer 1745. Many new features have been added.

For examples on migrating from DocEdit to Authentic see the description of the different methods and properties of the different DocEdit objects.

Properties
Start 1964
StartTextPosition 1964
End 1963
EndTextPosition 1964

29.3.3.2.1 End (obsolete)

Superseded by AuthenticRange.LastXMLData 1770

```
// ----- javascript sample -----
// instead of:
// var objXMLData = Application.ActiveDocument.DocEditView.CurrentSelection.End;
// use now:
var objXMLData = Application.ActiveDocument.AuthenticView.Selection.LastXMLData;
```

Declaration: End as XMLData 1939

Description
XML element where the current selection ends.

29.3.3.2.2 EndTextPosition (obsolete)

Superseded by AuthenticRange.LastXMLDataOffset 1770

```
// ----- javascript sample -----
// instead of:
// var nOffset = Application.ActiveDocument.DocEditView.CurrentSelection.EndTextPosition;
// use now:
var nOffset = Application.ActiveDocument.AuthenticView.Selection.LastXMLDataOffset;
```

Declaration: EndTextPosition as long

Description
Position in DocEditSelection.End.TextValue 1963 where the selection ends.

29.3.3.2.3 Start (obsolete)

Superseded by AuthenticRange.FirstXMLData 1758

```
// ----- javascript sample -----
// instead of:
// var objXMLData = Application.ActiveDocument.DocEditView.CurrentSelection.Start;
// use now:
var objXMLData = Application.ActiveDocument.AuthenticView.Selection.FirstXMLData;
```

Declaration: Start as XMLData 1939

Description
XML element where the current selection starts.

29.3.3.2.4 StartTextPosition (obsolete)

Superseded by AuthenticRange.FirstXMLDataOffset 1759

```
// ----- javascript sample -----
// instead of:
// var nOffset = Application.ActiveDocument.DocEditView.CurrentSelection.StartTextPosition;
```


© 2016-2022 Altova GmbH

```
// use now:
var nOffset = Application.ActiveDocument.AuthenticView.Selection.FirstXMLDataOffset;
```

Declaration: StartTextPosition as long

Description
Position in DocEditSelection.Start.TextValue[1964] where the selection starts.

29.3.3.3 OldAuthentictView (obsolete)

Superseded by AuthenticView[1778] and AuthenticRange[1750]

The DocEditView object is renamed to OldAuthenticView.
DocEditSelection is renamed to AuthenticSelection.
DocEditEvent is renamed to AuthenticEvent.
DocEditDataTransfer is renamed to AuthenticDataTransfer.

Their usage - except for AuthenticDataTransfer - is no longer recommended. We will continue to support existing functionality for a yet undefined period of time but no new features will be added to these interfaces. All functionality available up to now in DocEditView[1965], DocEditSelection[1963], DocEditEvent[1950] and DocEditDataTransfer[1746] is now available via AuthenticView[1778], AuthenticRange[1750] and AuthenticDataTransfer[1745]. Many new features have been added.

For examples on migrating from DocEdit to Authentic see the description of the different methods and properties of the different DocEdit objects.

Methods

LoadXML[1981]
SaveXML[1985]

EditClear[1968]
EditCopy[1968]
EditCut[1969]
EditPaste[1969]
EditRedo[1970]
EditSelectAll[1970]
EditUndo[1970]

RowAppend[1982]
RowDelete[1983]
RowDuplicate[1983]
RowInsert[1984]
RowMoveDown[1984]
RowMoveUp[1985]

ApplyTextState[1966]

IsTextStateApplied 1980
IsTextStateEnabled 1981

MarkUpView 1982

SelectionSet 1986
SelectionMoveTabOrder 1986

GetNextVisible 1973
GetPreviousVisible 1974

GetAllowedElements 1971

Properties
CurrentSelection 1967

event 1971

XMLRoot 1987

IsEditClearEnabled 1974
IsEditCopyEnabled 1975
IsEditCutEnabled 1975
IsEditPasteEnabled 1976
IsEditRedoEnabled 1976
IsEditUndoEnabled 1977

IsRowAppendEnabled 1977
IsRowDeleteEnabled 1978
IsRowDuplicateEnabled 1978
IsRowInsertEnabled 1979
IsRowMoveDownEnabled 1979
IsRowMoveUpEnabled 1980

Description
Interface for vista Authentic.

29.3.3.3.1 ApplyTextState (obsolete)

Superseded by AuthenticRange.PerformAction 1773

Use spyAuthenticApply for the eAction parameter. The PerformAction method allows to apply text state attributes to any range of the document, not only the current UI selection.

// ----- javascript sample -----
// instead of:

© 2016-2022 Altova GmbH

```
// Application.ActiveDocument.DocEditView.ApplyTextState ("bold");
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.PerformAction (spyAuthenticApply, "bold"))
    MsgBox ("Error: can't set current selection to bold");
```

Declaration: ApplyTextState (*elementName* as String)

Description
Applies or removes the text state defined by the parameter elementName. Common examples for the parameter elementName would be strong and italic.

In an XML document there are segments of data, which may contain sub-elements. For example consider the following HTML:

```
<b>fragment</b>
```

The HTML tag <b> will cause the word fragment to be bold. However, this only happens because the HTML parser knows that the tag <b> is bold. With XML there is much more flexibility. It is possible to define any XML tag to do anything you desire. The point is that it is possible to apply a Text state using XML. But the Text state that is applied must be part of the schema. For example in the OrgChart.xml, OrgChart.sps, OrgChart.xsd example the tag is the same as bold. And to apply bold the method ApplyTextState() is called. But like the row and edit operations it is necessary to test if it is possible to apply the text state.

See also IsTextStateEnabled[1981] and IsTextStateApplied[1980].

29.3.3.3.2 CurrentSelection (obsolete)

Superseded by AuthenticView.Selection[1795]

The returned AuthenticRange[1750] object supports navigation via XMLData elements as well as navigation by document elements (e.g. characters, words, tags) or text cursor positions.

```
// ----- javascript sample -----
// instead of:
// var objDocEditSel = Application.ActiveDocument.DocEditView.CurrentSelection;
// use now:
    var objRange = Application.ActiveDocument.AuthenticView.Selection;
```

Declaration: CurrentSelection as DocEditSelection[1963]

Description
The property provides access to the current selection in the vista Authentic.

29.3.3.3.3 EditClear (obsolete)

Superseded by AuthenticRange.Delete 1755

The Delete method of AuthenticRange allows to delete any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.EditClear();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.Delete())
    MsgBox ("Error: can't delete current selection");
```

Declaration: EditClear

Description
Deletes the current selection.

29.3.3.3.4 EditCopy (obsolete)

Superseded by AuthenticRange.Copy 1754

The Copy method of AuthenticRange allows to delete any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.EditCopy();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.Copy())
    MsgBox ("Error: can't copy current selection");
```

Declaration: EditCopy

Description
Copies the current selection to the clipboard.

© 2016-2022 Altova GmbH

29.3.3.3.5 EditCut (obsolete)

Superseded by AuthenticRange.Cut 1755

The Cut method of AuthenticRange allows to delete any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.EditCut();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.Cut())
    MsgBox ("Error: can't cut out current selection");
```

Declaration: EditCut

Description
Cuts the current selection from the document and copies it to the clipboard.

29.3.3.3.6 EditPaste (obsolete)

Superseded by AuthenticRange.Paste 1773

The Paste method of AuthenticRange allows to delete any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.EditPaste();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.Paste())
    MsgBox ("Error: can't paste to current selection");
```

Declaration: EditPaste

Description
Pastes the content from the clipboard into the document.

29.3.3.3.7 EditRedo (obsolete)

Superseded by AuthenticView.Redo 1795

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.EditRedo();
// use now:
if (! Application.ActiveDocument.AuthenticView.Redo())
    MsgBox ("Error: no redo step available");
```

Declaration: EditRedo

Description
Redo the last undo step.

29.3.3.3.8 EditSelectAll (obsolete)

Superseded by AuthenticView.WholeDocument 1797 and **AuthenticRange.Select** 1774

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.EditSelectAll();
// use now:
Application.ActiveDocument.AuthenticView.WholeDocument.Select();
```

Declaration: EditSelectAll

Description
The method selects the complete document.

29.3.3.3.9 EditUndo (obsolete)

Superseded by AuthenticView.Undo 1796

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.EditUndo();
// use now:
```


© 2016-2022 Altova GmbH

```
if (! Application.ActiveDocument.AuthenticView.Undo())
    MsgBox ("Error: no undo step available");
```

Declaration: EditUndo

Description
Undo the last action.

29.3.3.3.10 event (obsolete)

Superseded by parameters to AuthenticView events[1778].

Declaration: event as DocEditEvent[1950]

Description
The event property holds a DocEditEvent object which contains information about the current event.

29.3.3.3.11 GetAllowedElements (obsolete)

Superseded by AuthenticRange.CanPerformActionWith[1753]

AuthenticRange now supports all functionality of the 'elements' entry helper. Besides querying the elements that can be inserted, appended, etc., you can invoke the action as well. See AuthenticRange.PerformAction[1773] for more information.

```
// ----- javascript sample -----
// instead of:
// var arrElements = New Array();
// var objDocEditView = Application.ActiveDocument.DocEditView;
// var objStartElement = objDocEditView.CurrentSelection.Start;
// var objEndElement = objDocEditView.CurrentSelection.End;
// objDocEditView.GetAllowedElements(k_ActionInsertBefore, objStartElement, objEndElement,
arrElements);
// use now:
var arrElements = New Array();
Application.ActiveDocument.AuthenticView.Selection.CanPerformActionWith (spyAuthenticInsertBefore,
arrElements);
```

Declaration: GetAllowedElements (*nAction* as SpyAuthenticElementActions[1989], *pStartElement* as XMLData[1939], *pEndElement* as XMLData[1939], *pElements* as Variant)

Description
GetAllowedElements() returns the allowed elements for the various actions specified by nAction.

JavaScript example:

```
Function GetAllowed()
{
	var objView = Application.ActiveDocument.DocEditView;

	var arrElements = New Array(1);

	var objStart = objView.CurrentSelection.Start;
	var objEnd = objView.CurrentSelection.End;

	var strText;
	strText = "valid elements at current selection:\n\n";

	For(var i = 1;i <= 4;i++)  {
		objPlugIn.GetAllowedElements(i,objStart,objEnd,arrElements);
		strText = strText + ListArray(arrElements) + "----------\n";
	}

	Return strText;
}

Function ListArray(arrIn)
{
	var strText = "";

	If(TypeOf(arrIn) == "object")	{
		For(var i = 0;i <= (arrIn.length - 1);i++)
			strText = strText + arrIn[i] + "\n";
	}

	Return strText;
}
```

VBScript example:

```
Sub DisplayAllowed
	Dim objView
	Set objView = Application.ActiveDocument.DocEditView

	Dim arrElements()

	Dim objStart
	Dim objEnd
	Set objStart = objView.CurrentSelection.Start
	Set objEnd = objView.CurrentSelection.End

	Dim strText
	strText = "valid elements at current selection:" & chr(13) & chr(13)
```

 © 2016-2022 Altova GmbH

```
        Dim i

        For i = 1 To 4
                objView.GetAllowedElements i,objStart,objEnd,arrElements
                strText = strText & ListArray(arrElements) & "----------" & chr(13)
        Next

        msgbox strText
End Sub

Function ListArray(arrln)
        Dim strText

        If IsArray(arrln) Then
                Dim i

                For i = 0 To UBound(arrln)
                        strText = strText & arrln(i) & chr(13)
                Next
        End If

        ListArray = strText
End Function
```

29.3.3.3.12 GetNextVisible (obsolete)

Superseded by AuthenticRange.SelectNext[1775]

AuthenticRange now supports a wide range of element navigation methods based on document elements like characters, words, tags and many more. Selecting the text passage that represents the content of the next XML element is just one of them.

```
// ----- javascript sample -----
// instead of:
// var objCurrXMLData = ...
// var objXMLData = Application.ActiveDocument.DocEditView.GetNextVisible(objCurrXMLData);
// Application.ActiveDocument.DocEditView.SelectionSet (objXMLData, 0, objXMLData, -1);
// use now:
var objRange = ...
try
   { objRange.SelectNext (spyAuthenticTag).Select(); }
catch (err)
{
   if ((err.number & 0xffff) == 2003)
      MsgBox ("end of document reached");
   else
      throw (err);
```

```
}
```

Declaration: GetNextVisible (*pElement* as XMLData[1939]) as XMLData[1939]

Description
The method gets the next visible XML element in the document.

29.3.3.3.13 GetPreviousVisible (obsolete)

Superseded by AuthenticRange.SelectPrevious[1775]

AuthenticRange now supports a wide range of element navigation methods based on document elements like characters, words, tags and many more. Selecting the text passage that represents the content of the previous XML element is just one of them.

```
// ----- javascript sample -----
// instead of:
// var objCurrXMLData = ...
// var objXMLData = Application.ActiveDocument.DocEditView.GetPreviousVisible(objCurrXMLData);
// Application.ActiveDocument.DocEditView.SelectionSet (objXMLData, 0, objXMLData, -1);
// use now:
var objRange = ...
try
   { objRange.SelectPrevious (spyAuthenticTag).Select(); }
catch (err)
{
   if ((err.number & 0xffff) == 2004)
      MsgBox ("begin of document reached");
   else
      throw (err);
}
```

Declaration: GetPreviousVisible (*pElement* as XMLData[1939]) as XMLData[1939]

Description
The method gets the previous visible XML element in the document.

29.3.3.3.14 IsEditClearEnabled (obsolete)

Superseded by AuthenticRange.IsDeleteEnabled[1767]

© 2016-2022 Altova GmbH

The IsDeleteEnabled property is now supported for any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsEditClearEnabled)
//    Application.ActiveDocument.DocEditView.EditClear();
// use now:
var objCurrSelection = Application.ActiveDocument.AuthenticView.Selection;
if (objCurrSelection.IsDeleteEnabled)
   objCurrSelection.Delete();
```

Declaration: IsEditClearEnabled as Boolean

Description
True if EditClear[1968] is possible. See also Editing operations[1722].

29.3.3.3.15 IsEditCopyEnabled (obsolete)

Superseded by AuthenticRange.IsCopyEnabled[1766]

The IsCopyEnabled property is now supported for any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsEditCopyEnabled)
//    Application.ActiveDocument.DocEditView.EditCopy();
// use now:
var objCurrSelection = Application.ActiveDocument.AuthenticView.Selection;
if (objCurrSelection.IsCopyEnabled)
   objCurrSelection.Copy();
```

Declaration: IsEditCopyEnabled as Boolean

Description
True if copy to clipboard is possible. See also EditCopy[1968] and Editing operations[1722].

29.3.3.3.16 IsEditCutEnabled (obsolete)

Superseded by AuthenticRange.IsCutEnabled[1766]

The IsCutEnabled property is now supported for any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsEditCutEnabled)
//    Application.ActiveDocument.DocEditView.EditCut();
// use now:
var objCurrSelection = Application.ActiveDocument.AuthenticView.Selection;
if (objCurrSelection.IsCutEnabled)
   objCurrSelection.Cut();
```

Declaration: IsEditCutEnabled as Boolean

Description
True if EditCut[1969] is currently possible. See also Editing operations[1722].

29.3.3.3.17 IsEditPasteEnabled (obsolete)

Superseded by AuthenticRange.IsPasteEnabled[1768]

The IsPasteEnabled property is now supported for any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsEditPasteEnabled)
//    Application.ActiveDocument.DocEditView.EditPaste();
// use now:
var objCurrSelection = Application.ActiveDocument.AuthenticView.Selection;
if (objCurrSelection.IsPasteEnabled)
   objCurrSelection.Paste();
```

Declaration: IsEditPasteEnabled as Boolean

Description
True if EditPaste[1969] is possible. See also Editing operations[1722].

29.3.3.3.18 IsEditRedoEnabled (obsolete)

Superseded by AuthenticView.IsRedoEnabled[1793]

© 2016-2022 Altova GmbH

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsEditRedoEnabled)
//    Application.ActiveDocument.DocEditView.EditRedo();
// use now:
if (Application.ActiveDocument.AuthenticView.IsRedoEnabled)
   Application.ActiveDocument.AuthenticView.Redo();
```

Declaration: IsEditRedoEnabled as Boolean

Description
True if EditRedo[1970] is currently possible. See also Editing operations[1722].

29.3.3.3.19 IsEditUndoEnabled (obsolete)

Superseded by AuthenticView.IsUndoEnabled[1794]

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsEditUndoEnabled)
//    Application.ActiveDocument.DocEditView.EditUndo();
// use now:
if (Application.ActiveDocument.AuthenticView.IsUndoEnabled)
   Application.ActiveDocument.AuthenticView.Undo();
```

Declaration: IsEditUndoEnabled as Boolean

Description
True if EditUndo[1970] is possible. See also Editing operations[1722].

29.3.3.3.20 IsRowAppendEnabled (obsolete)

Superseded by AuthenticRange.IsInDynamicTable[1768]

The operations 'insert', 'append', 'delete' and 'duplicate' row are available whenever the selection is inside a dynamic table.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsRowAppendEnabled)
```

```
//    Application.ActiveDocument.DocEditView.RowAppend();
// use now:
if (Application.ActiveDocument.AuthenticView.Selection.IsInDynamicTable())
    Application.ActiveDocument.AuthenticView.Selection.AppendRow();
```

Declaration: IsRowAppendEnabled as Boolean

Description
True if RowAppend[1982] is possible. See also Row operations[1722].

29.3.3.3.21 IsRowDeleteEnabled (obsolete)

Superseded by AuthenticRange.IsInDynamicTable[1768]

The operations 'insert', 'append', 'delete' and 'duplicate' row are available whenever the selection is inside a dynamic table.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsRowDeleteEnabled)
//    Application.ActiveDocument.DocEditView.Rowdelete();
// use now:
if (Application.ActiveDocument.AuthenticView.Selection.IsInDynamicTable())
    Application.ActiveDocument.AuthenticView.Selection.DeleteRow();
```

Declaration: IsRowDeleteEnabled as Boolean

Description
True if RowDelete[1983] is possible. See also Row operations[1722].

29.3.3.3.22 IsRowDuplicateEnabled (obsolete)

Superseded by AuthenticRange.IsInDynamicTable[1768]

The operations 'insert', 'append', 'delete' and 'duplicate' row are available whenever the selection is inside a dynamic table.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsRowDuplicateEnabled)
//    Application.ActiveDocument.DocEditView.RowDuplicate();
// use now:
```


© 2016-2022 Altova GmbH

```
if (Application.ActiveDocument.AuthenticView.Selection.IsInDynamicTable())
    Application.ActiveDocument.AuthenticView.Selection.DuplicateRow();
```

Declaration: IsRowDuplicateEnabled as Boolean

Description
True if RowDuplicate[1983] is currently possible. See also Row operations[1722].

29.3.3.3.23 IsRowInsertEnabled (obsolete)

Superseded by AuthenticRange.IsInDynamicTable[1768]

The operations 'insert', 'append', 'delete' and 'duplicate' row are available whenever the selection is inside a dynamic table.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsRowInsertEnabled)
//    Application.ActiveDocument.DocEditView.RowInsert();
// use now:
if (Application.ActiveDocument.AuthenticView.Selection.IsInDynamicTable())
    Application.ActiveDocument.AuthenticView.Selection.InsertRow();
```

Declaration: IsRowInsertEnabled as Boolean

Description
True if RowInsert[1984] is possible. See also Row operations[1722].

29.3.3.3.24 IsRowMoveDownEnabled (obsolete)

Superseded by AuthenticRange.IsLastRow[1768]

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.OldAuthenticView.IsRowMoveDownEnabled)
//    Application.ActiveDocument.DocEditView.RowMoveDown();
// use now:
if (!Application.ActiveDocument.AuthenticView.Selection.IsLastRow)
    Application.ActiveDocument.AuthenticView.Selection.MoveRowDown();
```

Declaration: IsRowMoveDownEnabled as Boolean

Description
True if RowMoveDown[1984] is currently possible. See also Row operations[1722].

29.3.3.3.25 IsRowMoveUpEnabled (obsolete)

Superseded by AuthenticRange.IsFirstRow[1767]

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsRowMoveUpEnabled)
//    Application.ActiveDocument.DocEditView.RowMoveUp();
// use now:
if (!Application.ActiveDocument.AuthenticView.Selection.IsFirstRow)
   Application.ActiveDocument.AuthenticView.Selection.MoveRowUp();
```

Declaration: IsRowMoveUpEnabled as Boolean

Description
True if RowMoveUp[1985] is possible. See also Row operations[1722].

29.3.3.3.26 IsTextStateApplied (obsolete)

Superseded by AuthenticRange.IsTextStateApplied[1769]

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.IsTextStateApplied ("bold");
// use now:
if (Application.ActiveDocument.AuthenticView.Selection.IsTextStateApplied ("bold"))
   MsgBox ("bold on");
else
   MsgBox ("bold off");
```

Declaration: IsTextStateApplied (*elementName* as String) as Boolean

Description
Checks to see if the it the text state has already been applied. Common examples for the parameter elementName would be strong and italic.

© 2016-2022 Altova GmbH

29.3.3.3.27 IsTextStateEnabled (obsolete)

Superseded by AuthenticRange.CanPerformAction 1753

Use spyAuthenticApply for the eAction parameter. The CanPerformAction method allows to operate on any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.IsTextStateEnabled ("bold");
// use now:
if (Application.ActiveDocument.AuthenticView.Selection.CanPerformAction (spyAuthenticApply, "bold"))
    ... // e.g. enable 'bold' button
```

Declaration: IsTextStateEnabled (*i_strElementName* as String) as Boolean

Description
Checks to see if it is possible to apply a text state. Common examples for the parameter elementName would be strong and italic.

29.3.3.3.28 LoadXML (obsolete)

Superseded by AuthenticView.AsXMLString 1789

AuthenticView now supports the property AsXMLString that can be used to directly access and replace the document content as an XMLString.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.LoadXML (strDocAsXMLString);
// use now:
try
    { Application.ActiveDocument.AuthenticView.AsXMLString = strDocAsXMLString; }
catch (err)
    { MsgBox ("Error: invalid XML string"); }
```

Declaration: LoadXML (*xmlString* as String)

Description
Loads the current XML document with the XML string applied. The new content is displayed immediately. The *xmlString* parameter must begin with the XML declaration, e.g.,
objPlugIn.LoadXML("<?xml version='1.0' encoding='UTF-8'?><root></root>");

29.3.3.3.29 MarkUpView (obsolete)

Superseded by AuthenticView.MarkupVisibility 1794

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.MarkuUpView = 2;
// use now:
Application.ActiveDocument.AuthenticView.MarkupVisibility = spyAuthenticMarkupLarge;
```

Declaration: MarkUpView (*kind* as long)

Description
By default the document displayed is using HTML techniques. But sometimes it is desirable to show the editing tags. Using this method it is possible to display three different types of markup tags:

0	hide the markup tags
2	show the large markup tags
3	show the mixed markup tags.

29.3.3.3.30 RowAppend (obsolete)

Superseded by AuthenticRange.AppendRow 1752

The table operations of AuthenticRange now allow to manipulate any table in the current document independent of the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.RowAppend();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.AppendRow())
    MsgBox ("Error: can't append row");
```

Declaration: RowAppend

Description
Appends a row at the current position.

See also Row operations 1722.

© 2016-2022 Altova GmbH

29.3.3.3.31 RowDelete (obsolete)

Superseded by AuthenticRange.DeleteRow 1755

The table operations of AuthenticRange now allow to manipulate any table in the current document independent of the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.RowDelete();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.DeleteRow())
    MsgBox ("Error: can't delete row");
```

Declaration: RowDelete

Description
Deletes the currently selected row(s).

See also Row operations 1722.

29.3.3.3.32 RowDuplicate (obsolete)

Superseded by AuthenticRange.DuplicateRow 1756

The table operations of AuthenticRange now allow to manipulate any table in the current document independent of the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.RowDuplicate();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.DuplicateRow())
    MsgBox ("Error: can't duplicate row");
```

Declaration: RowDuplicate

Description
The method duplicates the currently selected rows.

See also Row operations 1722.

29.3.3.3.33 RowInsert (obsolete)

Superseded by AuthenticRange.InsertRow 1766

The table operations of AuthenticRange now allow to manipulate any table in the current document independent of the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.RowInsert();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.InsertRow())
    MsgBox ("Error: can't insert row");
```

Declaration: RowInsert

Description
Inserts a new row immediately above the current selection.

See also Row operations 1722.

29.3.3.3.34 RowMoveDown (obsolete)

Superseded by AuthenticRange.MoveRowDown 1772

The table operations of AuthenticRange now allow to manipulate any table in the current document independent of the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.RowMoveDown();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.MoveRowDown())
    MsgBox ("Error: can't move row down");
```

Declaration: RowMoveDown

Description
Moves the current row one position down.

See also Row operations 1722.

 © 2016-2022 Altova GmbH

29.3.3.3.35 RowMoveUp (obsolete)

Superseded by AuthenticRange.MoveRowUp 1773

The table operations of AuthenticRange now allow to manipulate any table in the current document independent of the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.RowAppend();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.MoveRowUp())
    MsgBox ("Error: can't move row up");
```

Declaration: RowMoveUp

Description
Moves the current row one position up.

See also Row operations 1722.

29.3.3.3.36 SaveXML (obsolete)

Superseded by AuthenticView.AsXMLString 1789

AuthenticView now supports the property XMLString that can be used to directly access and replace the document content as an XMLString.

```
// ----- javascript sample -----
// instead of:
// var strDocAsXMLString = Application.ActiveDocument.DocEditView.SaveXML();
// use now:
try
{
    var strDocAsXMLString = Application.ActiveDocument.AuthenticView.AsXMLString;
    ... // do something here
}
catch (err)
    { MsgBox ("Error: invalid XML string"); }
```

Declaration: SaveXML as String

Return Value

XML structure as string

Description
Saves the current XML data to a string that is returned to the caller.

29.3.3.3.37 SelectionMoveTabOrder (obsolete)

Superseded by AuthenticRange.SelectNext[1775]

AuthenticRange now supports a wide range of element navigation methods based on document elements like characters, words, tags and many more. Selecting the next paragraph is just one of them, and navigation is not necessarily bound to the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.SelectionMoveTabOrder(true, true);
// use now:
Application.ActiveDocument.AuthenticView.Selection.SelectNext (spyAuthenticParagraph).Select();
// to append a row to a table use AuthenticRange.AppendRow
```

Declaration: SelectionMoveTabOrder (*bForward* as Boolean, *bTag* as Boolean)

Description
SelectionMoveTabOrder() moves the current selection forwards or backwards.

If bTag is false and the current selection is at the last cell of a table a new line will be added.

29.3.3.3.38 SelectionSet (obsolete)

Superseded by AuthenticRange.FirstXMLData[1758] and related properties

AuthenticRange supports navigation via XMLData elements as well as navigation by document elements (e.g. characters, words, tags) or text cursor positions.

```
// ----- javascript sample -----
// instead of:
// if (! Application.ActiveDocument.DocEditView.SelectionSet(varXMLData1, 0, varXMLData2, -1))
//    MsgBox ("Error: invalid data position");
// use now:
try
{
   var objSelection = Application.ActiveDocument.AuthenticView.Selection;
```


© 2016-2022 Altova GmbH

```
    objSelection.FirstXMLData = varXMLData1;
    objSelection.FirstXMLdataOffset = 0;
    objSelection.LastXMLData = varXMLData2;
    objSelection.LastXMLDataOffset = -1;
    objSelection.Select();
}
catch (err)
    { MsgBox ("Error: invalid data position"); }
// to select all text between varXMLData1 and varXMLdata2, inclusive
```

Declaration: SelectionSet (*pStartElement* as XMLData[1939], *nStartPos* as long, *pEndElement* as XMLData[1939], *nEndPos* as long) as Boolean

Description
Use SelectionSet() to set a new selection in the vista Authentic. Its possible to set pEndElement to null (nothing) if the selection should be just over one (pStartElement) XML element.

29.3.3.3.39 XMLRoot (obsolete)

Superseded by AuthenticView.XMLDataRoot[1797]

```
// ----- javascript sample -----
// instead of:
// var objXMLData = Application.ActiveDocument.DocEditView.XMLRoot;
// use now:
var objXMLData = Application.ActiveDocument.AuthenticView.XMLDataRoot;
```

Declaration: XMLRoot as XMLData[1939]

Description
XMLRoot is the parent element of the currently displayed XML structure. Using the XMLData[1939] interface you have full access to the complete content of the file.

29.3.4 Enumerations

This is a list of all enumerations used by the XMLSpy API. If your scripting environment does not support enumerations use the number-values instead.

29.3.4.1 ENUMApplicationStatus

Description
Enumeration to specify the current Application status.

Possible values:

eApplicationRunning	= 0
eApplicationAfterLicenseCheck	= 1
eApplicationBeforeLicenseCheck	= 2
eApplicationConcurrentLicenseCheckFailed	= 3
eApplicationProcessingCommandLine	= 4

29.3.4.2 SPYAttributeTypeDefinition

Description
Attribute type definition that can be selected for generation of Sample XML.
This type is used with the method GenerateDTDOrSchema[1828] and GenerateDTDOrSchemaEx[1828].

Possible values:

spyMergedGlobal	= 0
spyDistinctGlobal	= 1
spyLocal	= 2

29.3.4.3 SPYAuthenticActions

Description
Actions that can be performed on AuthenticRange[1750] objects.

Possible values:

spyAuthenticInsertAt	= 0
spyAuthenticApply	= 1
spyAuthenticClearSurr	= 2
spyAuthenticAppend	= 3
spyAuthenticInsertBefore	= 4
spyAuthenticRemove	= 5

29.3.4.4 SPYAuthenticDocumentPosition

Description
Relative and absolute positions used for navigating with AuthenticRange[1750] objects.

Possible values:

spyAuthenticDocumentBegin	= 0
spyAuthenticDocumentEnd	= 1
spyAuthenticRangeBegin	= 2
spyAuthenticRangeEnd	= 3

© 2016-2022 Altova GmbH

29.3.4.5 SPYAuthenticElementActions

Description
Actions that can be used with the obsolete object GetAllowedElements (superseded by AuthenticRange.CanPerformActionWith[1753]).

Possible values:

k_ActionInsertAt	= 0
k_ActionApply	= 1
k_ActionClearSurr	= 2
k_ActionAppend	= 3
k_ActionInsertBefore	= 4
k_ActionRemove	= 5

29.3.4.6 SPYAuthenticElementKind

Description
Enumeration of the different kinds of elements used for navigation and selection within the AuthenticRange[1750] and AuthenticView[1778] objects.

Possible values:

spyAuthenticChar	= 0
spyAuthenticWord	= 1
spyAuthenticLine	= 3
spyAuthenticParagraph	= 4
spyAuthenticTag	= 6
spyAuthenticDocument	= 8
spyAuthenticTable	= 9
spyAuthenticTableRow	= 10
spyAuthenticTableColumn	= 11

29.3.4.7 SPYAuthenticMarkupVisibility

Description
Enumeration values to customize the visibility of markup with MarkupVisibility[1794].

Possible values:

spyAuthenticMarkupHidden	= 0
spyAuthenticMarkupSmall	= 1
spyAuthenticMarkupLarge	= 2
spyAuthenticMarkupMixed	= 3

29.3.4.8 SPYAuthenticToolbarButtonState

Description
Authentic toolbar button states are given by the following enumeration:

Possible values:

authenticToolbarButtonDefault	= 0
authenticToolbarButtonEnabled	= 1
authenticToolbarButtonDisabled	= 2

29.3.4.9 SPYDatabaseKind

Description
Values to select different kinds of databases for import. See DatabaseConnection.DatabaseKind[1805] for its use.

Possible values:

spyDB_Access	= 0
spyDB_SQLServer	= 1
spyDB_Oracle	= 2
spyDB_Sybase	= 3
spyDB_MySQL	= 4
spyDB_DB2	= 5
spyDB_Other	= 6
spyDB_Unspecified	= 7
spyDB_PostgreSQL	= 8
spyDB_iSeries	= 9

29.3.4.10 SPYDialogAction

Description
Values to simulate different interactions on dialogs. See Dialogs[1809] for all dialogs available.

Possible values:

spyDialogOK	= 0	// simulate click on OK button
spyDialogCancel	= 1	// simulate click on Cancel button
spyDialogUserInput	= 2	// show dialog and allow user interaction

29.3.4.11 SPYDOMType

Description
Enumeration values to parameterize generation of C++ code from schema definitions.

Possible values:

© 2016-2022 Altova GmbH

spyDOMType_msxml4	= 0	Obsolete
spyDOMType_xerces	= 1	
spyDOMType_xerces3	= 2	
spyDOMType_msxml6	= 3	

`spyDOMType_xerces` indicates Xerces 2.x usage; `spyDOMType_xerces3` indicates Xerces 3.x usage.

29.3.4.12 SPYDTDSchemaFormat

Description
Enumeration to identify the different schema formats.

Possible values:

spyDTD	= 0
spyW3C	= 1

29.3.4.13 SPYEncodingByteOrder

Description
Enumeration values to specify encoding byte ordering for text import and export.

Possible values:

spyNONE	= 0
spyLITTLE_ENDIAN	= 1
spyBIG_ENDIAN	= 2

29.3.4.14 SPYExportNamespace

Description
Enumeration type to configure handling of namespace identifiers during export.

Possible values:

spyNoNamespace	= 0
spyReplaceColonWithUnderscore	= 1

29.3.4.15 SPYFindInFilesSearchLocation

Description
The different locations where a search can be performed. This type is used with the FindInFilesDlg 1857 dialog.

Possible values:

spyFindInFiles_Documents	= 0
spyFindInFiles_Project	= 1
spyFindInFiles_Folder	= 2

29.3.4.16 SPYFrequentElements

Description
Enumeration value to parameterize schema generation.

Possible values:

spyGlobalElements	= 0
spyGlobalComplexType	= 1

29.3.4.17 SPYImageKind

Description
Enumeration values to parameterize image type of the generated documentation. These values are used in SchemaDocumentationDialog.DiagramFormat[1880] and WSDLDocumentationDlg.DiagramFormat[1907].

Possible values:

spyImageType_PNG	= 0
spyImageType_EMF	= 1

29.3.4.18 SPYImportColumnsType

Description
Enumeration to specify different Import columns types.

Possible values:

spyImportColumns_Element	= 0
spyImportColumns_Attribute	= 1

29.3.4.19 SPYKeyEvent

Description
Enumeration type to identify the different key events. These events correspond with the equally named windows messages.

Possible values:

spyKeyDown	= 0
spyKeyUp	= 1
spyKeyPressed	= 2

29.3.4.20 SPYKeyStatus

Description
Enumeration type to identify the key status.

© 2016-2022 Altova GmbH

Possible values:

spyLeftShiftKeyMask	= 1
spyRightShiftKeyMask	= 2
spyLeftCtrlKeyMask	= 4
spyRightCtrlKeyMask	= 8
spyLeftAltKeyMask	= 16
spyRightAltKeyMask	= 32

29.3.4.21 SPYLibType

Description
Enumeration values to parameterize generation of C++ code from schema definitions.

Possible values:

spyLibType_static	= 0
spyLibType_dll	= 1

29.3.4.22 SPYLoading

Description
Enumeration values to define loading behaviour of URL files.

Possible values:

spyUseCacheProxy	= 0
spyReload	= 1

29.3.4.23 SPYMouseEvent

Description
Enumeration type that defines the mouse status during a mouse event. Use the enumeration values as bitmasks rather then directly comparing with them.

Examples

```
' to check for ctrl-leftbutton-down in VB
If (i_eMouseEvent = (XMLSpyLib.spyLeftButtonDownMask Or XMLSpyLib.spyCtrlKeyDownMask)) Then
        ' react on ctrl-leftbutton-down
End If

' to check for double-click with any button in VBScript
If (((i_eMouseEvent And spyDoubleClickMask) <> 0) Then
        ' react on double-click
End If
```

Possible values:

spyNoButtonMask	= 0

```
spyMouseMoveMask          = 1
spyLeftButtonMask         = 2
spyMiddleButtonMask       = 4
spyRightButtonMask        = 8
spyButtonUpMask           = 16
spyButtonDownMask         = 32
spyDoubleClickMask        = 64
spyShiftKeyDownMask       = 128
spyCtrlKeyDownMask        = 256
spyLeftButtonDownMask     = 34   // spyLeftButtonMask | spyButtonDownMask
spyMiddleButtonDownMask   = 36   // spyMiddleButtonMask | spyButtonDownMask
spyRightButtonDownMask    = 40   // spyRightButtonMask | spyButtonDownMask
spyLeftButtonUpMask       = 18   // spyLeftButtonMask | spyButtonUpMask
spyMiddleButtonUpMask     = 20   // spyMiddleButtonMask | spyButtonUpMask
spyRightButtonUpMask      = 24   // spyRightButtonMask | spyButtonUpMask
spyLeftDoubleClickMask    = 66   // spyRightButtonMask | spyButtonUpMask
spyMiddleDoubleClickMask  = 68   // spyMiddleButtonMask | spyDoubleClickMask
spyRightDoubleClickMask   = 72   // spyRightButtonMask | spyDoubleClickMask
```

29.3.4.24 SPYNumberDateTimeFormat

Description
Enumeration value to configure database connections.

Possible values:

```
spySystemLocale      = 0
spySchemaCompatible  = 1
```

29.3.4.25 SPYProgrammingLanguage

Description
Enumeration values to select the programming language for code generation from schema definitions.

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Possible values:

```
spyUndefinedLanguage  = -1
spyJava               = 0
spyCpp                = 1
spyCSharp             = 2
```

29.3.4.26 SPYProjectItemTypes

Description
Enumeration values to identify the different elements in project item lists. See SpyProjectItem.ItemType 1894.

Possible values:

© 2016-2022 Altova GmbH

spyUnknownItem = 0
spyFileItem = 1
spyFolderItem = 2
spyURLItem = 3

29.3.4.27 SPYProjectType

Description
Enumeration values to parameterize generation of C# from schema definitions.

Possible values:

spyVisualStudioProject	= 0	Obsolete
spyVisualStudio2003Project	= 1	Obsolete
spyBorlandProject	= 2	Obsolete
spyMonoMakefile	= 3	Obsolete
spyVisualStudio2005Project	= 4	Obsolete
spyVisualStudio2008Project	= 5	Obsolete
spyVisualStudio2010Project	= 6	For C++ code also
spyVisualStudio2013Project	= 7	For C++ code also
spyVisualStudio2015Project	= 8	For C++ code also
spyVisualStudio2017Project	= 9	For C++ code also
spyVisualStudio2019Project	=10	For C++ code also

29.3.4.28 SpySampleXMLGenerationChoiceMode

Description
This enumeration is used in `GenerateSampleXMLDlg.ChoiceMode`[1870]:

spySampleXMLGen_FirstBranch = 0
spySampleXMLGen_AllBranches = 1
spySampleXMLGen_ShortestBranch = 2

29.3.4.29 SPYSampleXMLGenerationOptimization (Obsolete)

This enumeration is OBSOLETE since v2014.

Description
Specify the elements that will be generated in the Sample XML.
This enumeration is used in GenerateSampleXMLDlg[1869].

Possible values:

spySampleXMLGen_Optimized	= 0

spySampleXMLGen_NonMandatoryElements = 1
spySampleXMLGen_Everything = 2

29.3.4.30 SpySampleXMLGenerationSampleValueHints

Description
This enumeration is used in GenerateSampleXMLDlg.SampleValueHints[1874]

spySampleXMLGen_FirstFit = 0
spySampleXMLGen_RandomFit = 1
spySampleXMLGen_CycleThrough = 2

29.3.4.31 SPYSampleXMLGenerationSchemaOrDTDAssignment

Description
Specifies what kind of reference to the schema/DTD should be added to the generated Sample XML.
This enumeration is used in GenerateSampleXMLDlg[1869].

Possible values:
spySampleXMLGen_AssignRelatively = 0
spySampleXMLGen_AssignAbsolutely = 1
spySampleXMLGen_DoNotAssign = 2

29.3.4.32 SPYSchemaDefKind

Description
Enumeration type to select schema diagram types.

Possible values:
spyKindElement = 0
spyKindComplexType = 1
spyKindSimpleType = 2
spyKindGroup = 3
spyKindModel = 4
spyKindAny = 5
spyKindAttr = 6
spyKindAttrGroup = 7
spyKindAttrAny = 8

© 2016-2022 Altova GmbH

spyKindIdentityUnique = 9
spyKindIdentityKey = 10
spyKindIdentityKeyRef = 11
spyKindIdentitySelector = 12
spyKindIdentityField = 13
spyKindNotation = 14
spyKindInclude = 15
spyKindImport = 16
spyKindRedefine = 17
spyKindFacet = 18
spyKindSchema = 19
spyKindCount = 20

29.3.4.33 SPYSchemaDocumentationFormat

Description
Enumeration values to parameterize generation of schema documentation. These values are used in SchemaDocumentationDialog.OutputFormat[1886] and WSDLDocumentationDlg.OutputFormat[1912].

Possible values:
spySchemaDoc_HTML = 0
spySchemaDoc_MSWord = 1
spySchemaDoc_RTF = 2
spySchemaDoc_PDF = 3

29.3.4.34 SPYSchemaExtensionType

Description
Enumeration to specify different Schema Extension types.

Possible values:

spySchemaExtension_None = 0
spySchemaExtension_SQL_XML = 1
spySchemaExtension_MS_SQL_Server = 2
spySchemaExtension_Oracle = 3

29.3.4.35 SPYSchemaFormat

Description
Enumeration to specify different Schema Format types.

Possible values:

spySchemaFormat_Hierarchical = 0
spySchemaFormat_Flat = 1

29.3.4.36 SPYTextDelimiters

Description
Enumeration values to specify text delimiters for text export.

Possible values:

spyTabulator	= 0
spySemicolon	= 1
spyComma	= 2
spySpace	= 3

29.3.4.37 SPYTextEnclosing

Description
Enumeration value to specify text enclosing characters for text import and export.

Possible values:

spyNoEnclosing	= 0
spySingleQuote	= 1
spyDoubleQuote	= 2

29.3.4.38 SPYTypeDetection

Description
Enumeration to select how type detection works during GenerateDTDOrSchema[1828] and GenerateDTDOrSchemaEx[1828].

Possible values:

spyBestPossible	= 0
spyNumbersOnly	= 1
spyNoDetection	= 2

29.3.4.39 SPYURLTypes

Description
Enumeration to specify different URL types.

Possible values:

spyURLTypeAuto	= -1
spyURLTypeXML	= 0
spyURLTypeDTD	= 1

© 2016-2022 Altova GmbH

29.3.4.40 SPYValidateXSDVersion

Description
Enumeration values that select what XSD version to use. The XSD version that is selected depends on both (i) the presence/absence—and, if present, the value—of the `/xs:schema/@vc:minVersion` attribute of the XSD document, and (ii) the value of this enumeration.

`spyValidateXSDVersion_1_0` selects `XSD 1.0` if `vc:minVersion` is absent, or is present with any value.
`spyValidateXSDVersion_1_1` selects `XSD 1.1` if `vc:minVersion` is absent, or is present with any value.
`spyValidateXSDVersion_AutoDetect` selects `XSD 1.1` if `vc:minVersion=1.1`. If the `vc:minVersion` attribute is absent, or is present with a value other than `1.1`, then `XSD 1.0` is selected.

Possible values

`spyValidateXSDVersion_AutoDetect`	`= 0`
`spyValidateXSDVersion_1_1`	`= 1`
`spyValidateXSDVersion_1_0`	`= 2`

29.3.4.41 SPYValidateErrorFormat

Description
Enumeration values that select the format of the error message.

Possible values

`spyValidateErrorFormat_Text`	`= 0`
`spyValidateErrorFormat_ShortXML`	`= 1`
`spyValidateErrorFormat_LongXML`	`= 2`

29.3.4.42 SPYViewModes

Description
Enumeration values that define the different view modes for XML documents. The mode *spyViewAuthentic(4)* identifies the mode that was intermediately called DocEdit mode and is now called Authentic mode. The mode *spyViewJsonSchema* identifies a mode which is mapped to the Schema Design View on the GUI but is distinguished internally.

Possible values:

spyViewGrid	= 0
spyViewText	= 1
spyViewBrowser	= 2
spyViewSchema	= 3

```
spyViewContent      = 4     // obsolete
spyViewAuthentic    = 4
spyViewWSDL         = 5
spyViewZIP          = 6
spyViewEditionInfo  = 7
spyViewXBRL         = 8
spyViewJsonSchema   = 9
```

29.3.4.43 SPYVirtualKeyMask

Description
Enumeration type for the most frequently used key masks that identify the status of the virtual keys. Use these values as bitmasks rather then directly comparing with them. When necessary, you can create further masks by using the 'logical or' operator.

Examples

```
'  VBScript sample: check if ctrl-key is pressed
If ((i_nVirtualKeyStatus And spyCtrlKeyMask) <> 0)) Then
        ' ctrl-key is pressed
End If

'  VBScript sample: check if ONLY ctrl-key is pressed
If (i_nVirtualKeyStatus == spyCtrlKeyMask) Then
        ' exactly ctrl-key is pressed
End If

// JScript sample: check if any of the right virtual keys is pressed
if ((i_nVirtualKeyStatus & (spyRightShiftKeyMask | spyRightCtrlKeyMask | spyRightAltKeyMask)) != 0)
{
        ; ' right virtual key is pressed
}
```

Possible values:

```
spyNoVirtualKeyMask     = 0
spyLeftShiftKeyMask     = 1
spyRightShiftKeyMask    = 2
spyLeftCtrlKeyMask      = 4
spyRightCtrlKeyMask     = 8
spyLeftAltKeyMask       = 16
spyRightAltKeyMask      = 32
spyShiftKeyMask         = 3     // spyLeftShiftKeyMask | spyRightShiftKeyMask
spyCtrlKeyMask          = 12    // spyLeftCtrlKeyMask | spyRightCtrlKeyMask
spyAltKeyMask           = 48    // spyLeftAltKeyMask | spyRightAltKeyMask
```

29.3.4.44 SPYXMLDataKind

Description
The different types of XMLData elements available for XML documents.

© 2016-2022 Altova GmbH

Possible values:

spyXMLDataXMLDocStruct	= 0
spyXMLDataXMLEntityDocStruct	= 1
spyXMLDataDTDDocStruct	= 2
spyXMLDataXML	= 3
spyXMLDataElement	= 4
spyXMLDataAttr	= 5
spyXMLDataText	= 6
spyXMLDataCData	= 7
spyXMLDataComment	= 8
spyXMLDataPI	= 9
spyXMLDataDefDoctype	= 10
spyXMLDataDefExternalID	= 11
spyXMLDataDefElement	= 12
spyXMLDataDefAttlist	= 13
spyXMLDataDefEntity	= 14
spyXMLDataDefNotation	= 15
spyXMLDataKindsCount	= 16

29.3.5 Application API for Java (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

The Application API in Java has an interface built up of Java classes, each of which corresponds to an object in the Application API[1686]. Developers can use these Java classes to interact with the COM API. These classes are listed below and described in subsequent sections. For a description of the Application API[1686] objects themselves, see the Application API documentation[1686]. Bear in mind that some API features are only available in scripting environments; these have therefore not been ported to Java.

Java classes

SpyApplication[2004]
- SpyProject[2018]
  - SpyProjectItems[2019]
    - SpyProjectItem[2019]
- SpyDocuments[2011]
  - SpyDoc[2009]
    - SpyAuthenticView[2032]
      - SpyAuthenticRange[2031]
    - SpyDocEditView[2034]
      - SpyDocEditSelection[2033]
    - SpyGridView[2018]

SpyTextView 2023
SpyXMLData 2030
SpyDialogs 2008
SpyCodeGeneratorDlg 2006
SpyDTDSchemaGeneratorDlg 2011
SpyFileSelectionDlg 2014
SpyFindInFilesDlg 2014
SpyGenerateSampleXMLDlg 2017
SpySchemaDocumentationDlg 2020
SpyWSDL20DocumentationDlg 2023
SpyWSDLDocumentationDlg 2025
SpyXBRLDocumentationDlg 2028
SpyDatabaseConnection 2007
SpyElementList 2012
SpyElementListItem 2012
SpyExportSettings 2013
SpyFindInFilesResults 2016
SpyFindInFilesResult 2016
SpyFindInFilesMatch 2015
SpyTextImportExportSettings 2022

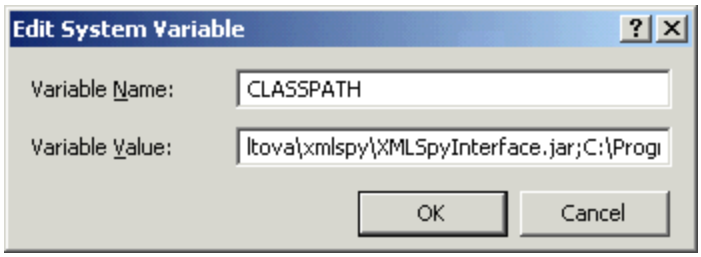
Implementation of COM properties in Java
Properties in Java have been defined to include both a **`set`** and **`get`** method (`set` if it is allowed by the COM implementation). For example, the COM class `Document` contains the **`GridView`** property. In Java the method is called `SpyDoc` and the property is defined as a **`GetGridView`** method.

If you encounter compiling problems, please check the following points:

- The `xmlspylib.dll` must be available in `..\windows\system32`.
- The `XMLSpyInterface.jar` file must be inserted in the ClassPath environment variable.

Setting the `ClassPath` variable in Windows XP

1. Click **Start | Settings | Control panel | System | Advanced | Environment Variables**. This opens the Environment Variables dialog box.
2. If a `ClassPath` entry already exists in the **System variables** group, select the `ClassPath` entry, and click the **Edit** button. Edit the path to: `"C:\Program Files\Altova\xmlspy\XMLSpyInterface.jar"`.



If a `ClassPath` entry does not exist in the System variables group, click the **New** button. The New System Variable dialog pops up. Enter `CLASSPATH` as the variable name, and `"C:\Program Files\Altova\xmlspy\XMLSpyInterface.jar"` as the `ClassPath` variable (alter the path to match your installation, if necessary).

 © 2016-2022 Altova GmbH

29.3.5.1 Sample source code (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

The "`SpyDoc doc = app.GetDocuments().OpenFile(...)`" command parameter must be altered to suit your environment.

What the sample does:

- Starts a new XMLSpy instance
- Opens the Datasheet.xml file (alter the path here...)
- Switches to the Enhanced Grid view
- Appends a new child element called "NewChild" with the text value "NewValuE" element to the root element
- Checks if the document is valid and outputs a message to the Java console
- Quits and releases the XMLSpy application

```
import XMLSpyInterface.*;

public class TestSpyInterface
{
    public TestSpyInterface() {}

    public static void main(String[] args)
    {
        SpyApplication app = null;
        SpyDoc oDoc = null;
        SpyXMLData oData = null;
        SpyXMLData oNewChild = null;

        try
        {
            app = new SpyApplication();
            app.ShowApplication( true );

            oDoc = app.GetDocuments().OpenFile("C:\\FilePath\\OrgChart.xml", true );

            // OrgChart.xml is in the folder C:\Documents and Settings\<usuario>\Mis
            Documentos\Altova\XMLSpy2023. The filepath should be in
            // the form: C:\\Documents and Settings\\Username\\Folder\\Filename.xml

            if ( oDoc != null )
            {
                oDoc.SwitchViewMode(SPYViewModes.spyViewGrid);
```

```
            oData = oDoc.GetRootElement();
            oNewChild = oDoc.CreateChild(SPYXMLDataKind.spyXMLDataElement);

            oNewChild.SetName( "NewChild" );
            oNewChild.SetTextValue("newVaLuE");
            oData.AppendChild(oNewChild);

            if ( oDoc.IsValid() == false )
            {
                // is to be expected after above insertion
                System.out.println( "!!!!!!validation error: " + oDoc.GetErrorString() );
                System.out.println( "!!!!!!validation error: " + oDoc.GetErrorPos() );
                System.out.println( "!!!!!!validation error: " + oDoc.GetBadData() );

            }
        }

        app.Quit();
    }
    finally
    {
        // Free any allocated resources by calling ReleaseInstance().
        if ( oNewChild != null )
            oNewChild.ReleaseInstance();

        if ( oData != null )
            oData.ReleaseInstance();

        if ( oDoc != null )
            oDoc.ReleaseInstance();

        if ( app != null )
            app.ReleaseInstance();
    }
  }
}
```

If you have difficulties compiling this sample, please try the following commands on the (**Start | Run | cmd**) command line. Please make sure you are currently in the folder that contains the sample java file.

compilation
javac -classpath c:*yourpathhere*\XMLSpyInterface.jar *testspyinterface.java*

Execution
java -classpath c:*yourpathhere*\XMLSpyInterface.jar *testspyinterface*

29.3.5.2 SpyApplication (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

© 2016-2022 Altova GmbH

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyApplication
{
  public void ReleaseInstance[2003]();
  public void ShowApplication[1741]( boolean bShow );
  public void Quit[1739]();
  public void AddMacroMenuItem[1728]( String sMacro, String sDisplayText );
  public void ClearMacroMenu[1729]();
  public SpyDoc[2009] GetActiveDocument[1728]();
  public SpyProject[2018] GetCurrentProject[1729]();
  public SpyDocuments[2011] GetDocuments[1730]();
  public SpyElementList[2012] GetDatabaseImportElementList[1731]( SpyDatabaseConnection[2007]
  oImportSettings );
  public SpyDatabaseConnection[2007] GetDatabaseSettings[1731]();
  public SpyElementList[2012] GetDatabaseTables[1732]( SpyDatabaseConnection[2007] oImportSettings );
  public SpyExportSettings[2013] GetExportSettings[1732]();
  public SpyElementList[2012] GetTextImportElementList[1733]( SpyTextImportExportSettings[2022]
  oImportSettings );
  public SpyTextImportExportSettings[2022] GetTextImportExportSettings[1733]();
  public SpyDoc[2009] ImportFromDatabase[1735]( SpyDatabaseConnection[2007] oImportSettings,
  SpyElementList[2012] oElementList );
  public SpyDoc[2009] ImportFromSchema[1736]( SpyDatabaseConnection[2007] oImportSettings, String
  strTable, SpyDoc[2009] oSchemaDoc );
  public SpyDoc[2009] ImportFromText[1736]( SpyTextImportExportSettings[2022] oImportSettings,
  SpyElementList[2012] oElementList );
  public SpyDoc[2009] ImportFromWord[1737]( String sFile );
  public void NewProject[1738]( String sPath, boolean bDiscardCurrent );
  public void OpenProject[1738](String sPath , boolean bDiscardCurrent, boolean bDialog );
  public long ShowForm[1742]( String sName );
  public void URLDelete[1742]( String sURL, String sUser, String sPassword );
  public void URLMakeDirectory[1742]( String sURL, String sUser, String sPassword );
  public int GetWarningNumber[1743]();
  public String GetWarningText[1743]();

  // since Version 2004R4
  public SpyApplication[2004] GetApplication[1728]();
  public SpyApplication[2004] GetParent[1739]();
  public SpyDialogs[2008] GetDialogs[1730]();
  public boolean GetVisible[1743]();
  public void SetVisible[1743]( boolean i_bVisibility );
  public long GetWindowHandle();

  public void ReloadSettings[1740]();
  public SpyFindInFilesResults[2016] FindInFiles[1730]( SpyFindInFilesDlg[2014] dlgSettings );
  public boolean ShowFindInFiles[1741]( SpyFindInFilesDlg[2014] dlgSettings );
  public void Selection( String sVal );

  public long[2035] Status[1742]();
```

```
  public int MajorVersion1738();
  public int MinorVersion1738();
  public String Edition1730();
  public boolean IsAPISupported1737();
  public long ServicePackVersion1741();
  public void CreateXMLSchemaFromDBStructure1729( SpyDatabaseConnection2007 oConnection,
SpyElementList2012 oTables );
}
```

29.3.5.3 SpyCodeGeneratorDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java1711.

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

```
// since version 2004R4
public class SpyCodeGeneratorDlg
{
  public void ReleaseInstance2003();
  public SpyApplication2004 GetApplication1798();
  public SpyDialogs2008 GetParent1802();
  public long2043 GetProgrammingLanguage1802();
  public void SetProgrammingLanguage1802( long2043 i_eVal );
  public String GetTemplateFileName1803();
  public void SetTemplateFileName1803( String i_strVal );
  public String GetOutputPath1801();
  public void SetOutputPath1801( String i_strVal );
  public long2039 GetOutputPathDialogAction1801();
  public void SetOutputPathDialogAction1801( long2039 i_eVal );
  public long2039 GetPropertySheetDialogAction1803();
  public void SetPropertySheetDialogAction1803( long2039 i_eVal );
  public long2039 GetOutputResultDialogAction1802();
  public void SetOutputResultDialogAction1802( long2039 i_eVal );
  public long2039 GetCPPSettings_DOMType1799();
  public void SetCPPSettings_DOMType1799( long2039 i_eVal );
  public long2042 GetCPPSettings_LibraryType1800();
  public void SetCPPSettings_LibraryType1800( long2042 i_eVal );
  public boolean GetCPPSettings_UseMFC1800();
  public void SetCPPSettings_UseMFC1800( boolean i_bVal );
  public long2044 GetCSharpSettings_ProjectType1801();
  public void SetCSharpSettings_ProjectType1801( long2044 i_eVal );
}
```


© 2016-2022 Altova GmbH

29.3.5.4 SpyDatabaseConnection (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyDatabaseConnection
{
  public void ReleaseInstance[2003]();
  public String GetADOConnection[1804]();
  public void SetADOConnection[1804]( String sValue );
  public boolean GetAsAttributes[1804]();
  public void SetAsAttributes[1804]( boolean bValue );
  public boolean GetCreateMissingTables[1805]();
  public void SetCreateMissingTables[1805]( boolean bValue );
  public boolean GetCreateNew[1805]();
  public void SetCreateNew[1805]( boolean bValue );
  public boolean GetExcludeKeys[1806]();
  public void SetExcludeKeys[1806]( boolean bValue );
  public String GetFile[1806]();
  public void SetFile[1806]( String sValue );
  public boolean GetIncludeEmptyElements[1807]();
  public void SetIncludeEmptyElements[1807]( boolean bValue );
  public long[2043] GetNumberDateTimeFormat[1807]();
  public void SetNumberDateTimeFormat[1807]( long[2043] nValue );
  public String GetODBCConnection[1808]();
  public void SetODBCConnection[1808]( String sValue );
  public String GetSQLSelect[1809]();
  public void SetSQLSelect[1809]( String sValue );
  public long GetTextFieldLen[1809]();
  public void SetTextFieldLen[1809]( long nValue );

  // since version 2004R4
  public long[2038] GetDatabaseKind[1805]();
  public void SetDatabaseKind[1805]( long[2038] nValue );

  // since version 2008R2
  public boolean GetCommentIncluded[1805]();
  public void SetCommentIncluded[1805]( boolean bValue );
  public String GetNullReplacement[1807]();
  public void SetNullReplacement[1807]( String sValue );
  public String GetDatabaseSchema[1806]();
  public void SetDatabaseSchema[1806]( String sValue );
```

```
  // since version 2010r3
  public boolean GetPrimaryKeys()
  public void SetPrimaryKeys( boolean bValue )
  public boolean GetForeignKeys()
  public void SetForeignKeys( boolean bValue )
  public boolean GetUniqueKeys()
  public void SetUniqueKeys( boolean bValue )
  public long GetSchemaExtensionType()
  public void SetSchemaExtensionType( long nValue )
  public long GetSchemaFormat()
  public void SetSchemaFormat( long nValue )
  public long GetImportColumnsType()
  public void SetImportColumnsType( long nValue )
}
```

29.3.5.5 SpyDialogs (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
// Since version 2004R4
public class SpyDialogs
{
  public SpyApplication GetApplication();
  public SpyApplication GetParent();
  public SpyCodeGeneratorDlg GetCodeGeneratorDlg();
  public SpyFileSelectionDlg GetFileSelectionDlg();
  public SpySchemaDocumentationDlg GetSchemaDocumentationDlg();
  public SpyGenerateSampleXMLDlg GetGenerateSampleXMLDlg();
  public SpyDTDSchemaGeneratorDlg GetDTDSchemaGeneratorDlg();
  public SpyFindInFilesDlg GetFindInFilesDlg();
  public SpyWSDLDocumentationDlg GetWSDLDocumentationDlg();

  // Since version 2010
  public SpyWSDL20DocumentationDlg GetWSDL20DocumentationDlg();
  public SpyXBRLDocumentationDlg GetXBRLDocumentationDlg();
}
```


© 2016-2022 Altova GmbH

29.3.5.6 SpyDoc (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyDoc
{
  public void ReleaseInstance[2003]();
  public void SetEncoding[1824]( String strEncoding );
  public void SetPathName[1824]( String strPath );
  public String GetPathName[1832]();
  public String GetTitle[1841]();
  public boolean IsModified[1833]();
  public void Save[1837]();
  public void Close[1820]( boolean bDiscardChanges );
  public void UpdateViews[1842]();
  public long[2050] GetCurrentViewMode[1823]();
  public boolean SwitchViewMode[1840]( long[2050] nMode );
  public SpyGridView[2018] GetGridView[1832]();
  public void SetActiveDocument[1838]();
  public void StartChanges[1840]();
  public void EndChanges[1824]();
  public void TransformXSL[1841]();
  public void AssignDTD[1818]( String sDTDFile, boolean bDialog );
  public void AssignSchema[1818]( String sSchemaFile, boolean bDialog );
  public void AssignXSL[1818]( String sXSLFile, boolean bDialog );
  public void ConvertDTDOrSchema[1820]( long[2039] nFormat, long[2041] nFrequentElements );
  public SpyXMLData[2030] CreateChild[1822]( long[2051] nKind );
  public void CreateSchemaDiagram[1823]( long[2046] nKind, String sName, String sFile );
  public SpyDocEditView[2034] GetDocEditView[1824]();
  public void ExportToDatabase[1825]( SpyXMLData[2030] oFromChild, SpyExportSettings[2013]
  oExportSettings, SpyDatabaseConnection[2007] oDatabaseConnection );
  public void ExportToText[1826]( SpyXMLData[2030] oFromChild, SpyExportSettings[2013]
  oExportSettings, SpyTextImportExportSettings[2022] oTextSettings );
  public void GenerateDTDOrSchema[1828]( long[2039] nFormat, int nValuesList, long[2049] nDetection,
  long[2041] nFrequentElements );
  public SpyElementList[2012] GetExportElementList[1831]( SpyXMLData[2030] oFromChild,
  SpyExportSettings[2013] oExportSettings );
  public SpyXMLData[2030] GetRootElement[1836]();
  public String SaveInString[1837]( SpyXMLData[2030] oData, boolean bMarked );
  public void SaveToURL[1838]( String sUrl, String sUser, String sPassword );
  public String GetErrorString(); // See IsValid[1833]() or IsWellFormed[1835]()
  public int GetErrorPos(); // See IsValid[1833]() or IsWellFormed[1835]()
  public SpyXMLData[2030] GetBadData(); // See IsValid[1833]() or IsWellFormed[1835]()
  public boolean IsValid[1833]();
```

```
  public boolean IsWellFormed( SpyXMLData oData, boolean bWithChildren );

  // Since version 2004R3
  public SpyAuthenticView GetAuthenticView()

  // Since version 2004R4
  public SpyApplication GetApplication();
  public SpyDocuments GetParent();
  public String GetFullName();
  public void SetFullName( String i_strName );
  public String GetName();
  public String GetPath();
  public boolean GetSaved();
  public void SaveAs( String i_strFileNameOrPath );
  public String GetEncoding();
  public SpyXMLData GetDataRoot();
  public void GenerateProgramCode( SpyCodeGeneratorDlg i_dlg );
  public void AssignXSLFO( String i_strFile, boolean i_bUseDialog );
  public void TransformXSLFO();
  public void GenerateSchemaDocumentation( SpySchemaDocumentationDlg i_dlg );

  public void ExecuteXQuery( String i_strXMLSourceFile );
  public void SetExternalIsValid( boolean bIsValid );
  public SpyDoc GenerateSampleXML( SpyGenerateSampleXMLDlg ipGenerateXMLDlg );
  public boolean UpdateXMLData();
  public String GetAsXMLString();
  public void SetAsXMLString( String newVal );
  public SpyDoc GenerateDTDOrSchemaEx( SpyDTDSchemaGeneratorDlg
  ipDTDSchemaGeneratorDlg );
  public SpyDoc ConvertDTDOrSchemaEx( long nFormat, long nFrequentElements,
  String sOutputPath, long nOutputPathDialogAction );
  public SpyTextView GetTextView();
  public String[] GetSuggestions();
  public void SetSuggestions( String[] aList );
  public void SetSelection( String sVal );

  // Since version 2009
  public void GenerateWSDLDocumentation( SpyWSDLDocumentationDlg ipWSDLDocumenationDlg
  );
  public void TransformXSLEx( long nDialogAction );

  // Since version 2010
  public void GenerateWSDL20Documentation( SpyWSDL20DocumentationDlg
  ipWSD20DocumenationDlg );
  public void GenerateXBRLDocumentation( SpyXBRLDocumentationDlg
  ipXBRLDocumentationDlg );
  public SpyDoc ConvertToWSDL20( String sFilePath, boolean bShowDialogs );

  // Since version 2010r3
  public String CreateDBStructureFromXMLSchema( SpyDatabaseConnection oConnection,
  SpyElementList oTables, boolean bDropTableWithExistingName );
  public SpyElementList GetDBStructureList( SpyDatabaseConnection oConnection );
}
```


© 2016-2022 Altova GmbH

29.3.5.7 SpyDocuments (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyDocuments
{
  public void ReleaseInstance[2003]();
  public long Count[1844]();
  public SpyDoc[2009] GetItem[1844]( long nNo );
  public SpyDoc[2009] NewFile[1845]( String strFile, String strType );
  public SpyDoc[2009] NewFileFromText[1845]( String nSource, String strType );
  public SpyDoc[2009] OpenFile[1846]( String sPath, boolean bDialog );
  public SpyDoc[2009] OpenURL[1846]( String sUrl, long[2049] nURLType, long[2043] nLoading, String
  sUser, String sPassword );
  public SpyDoc[2009] OpenURLDialog[1847](String sURL, long[2049] nURLType, long[2043] nLoading, String
  sUser, String sPassword );
  // Since version 2011r2
  public SpyDoc[2009] NewAuthenticFile[1844]( String strSPSPath, String strXMLPath );
  public SpyDoc[2009] OpenAuthenticFile[1845]( String strSPSPath, String strXMLPath );
}
```

29.3.5.8 SpyDTDSchemaGeneratorDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyDTDSchemaGeneratorDlg
{
  public void ReleaseInstance[2003]();
  public SpyApplication[2004] GetApplication[1848]();
  public long[2039] GetDTDSchemaFormat[1848]();
  public void SetDTDSchemaFormat[1848]( long[2039] newVal );
  public short GetValueList[1851]();
  public void SetValueList[1851]( short newVal );
  public long[2049] GetTypeDetection[1851]();
```

```
  public void SetTypeDetection( long newVal );
  public long GetFrequentElements();
  public void SetFrequentElements( long newVal );
  public boolean GetMergeAllEqualNamed();
  public void SetMergeAllEqualNamed( boolean newVal );
  public boolean GetResolveEntities();
  public void SetResolveEntities( boolean newVal );
  public long GetAttributeTypeDefinition();
  public void SetAttributeTypeDefinition( long newVal );
  public boolean GetGlobalAttributes();
  public void SetGlobalAttributes( boolean newVal );
  public boolean GetOnlyStringEnums();
  public void SetOnlyStringEnums( boolean newVal );
  public long GetMaxEnumLength();
  public void SetMaxEnumLength( long newVal );
  public String GetOutputPath();
  public void SetOutputPath( String newVal );
  public long GetOutputPathDialogAction();
  public void SetOutputPathDialogAction( long newVal );
}
```

29.3.5.9 SpyElementList (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyElementList
{
  public void ReleaseInstance();
  public long GetCount();
  public SpyElementListItem GetItem( long nIndex );
  public void RemoveElement( long nIndex );
}
```

29.3.5.10 SpyElementListItem (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

© 2016-2022 Altova GmbH

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyElementListItem
{
  public void ReleaseInstance[2003]();
  public long[2051] GetElementKind[1853]();
  public void SetElementKind[1853]( long[2051] nKind );
  public long GetFieldCount[1853]();
  public String GetName[1853]();
  public long GetRecordCount[1853]();
}
```

29.3.5.11 SpyExportSettings (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyExportSettings
{
  public void ReleaseInstance[2003]();
  public boolean GetCreateKeys[1854]();
  public void SetCreateKeys[1854]( boolean bValue );
  public SpyElementList[2012] GetElementList[1854]();
  public void SetElementList[1854]( SpyElementList[2012] obj );
  public boolean GetEntitiesToText[1854] ();
  public void SetEntitiesToText[1854]( boolean bValue );
  public boolean GetExportAllElements[1854]();
  public void SetExportAllElements[1854]( boolean bValue );
  public boolean GetFromAttributes[1855]();
  public void SetFromAttributes[1855]( boolean bValue );
  public boolean GetFromSingleSubElements[1855]();
  public void SetFromSingleSubElements[1855]( boolean bValue );
  public boolean GetFromTextValues[1855]();
  public void SetFromTextValues[1855]( boolean bValue );
  public boolean GetIndependentPrimaryKey[1855]();
  public void SetIndependentPrimaryKey[1855]( boolean bValue );
  public long[2040] GetNamespace[1855]();
  public void SetNamespace[1855]( long[2040] nValue );
  public int GetSubLevelLimit[1856]();
  public void SetSubLevelLimit[1856]( int nValue );
}
```

29.3.5.12 SpyFileSelectionDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
// Since version 2004R4
public class SpyFileSelectionDlg
{
  public void ReleaseInstance[2003]();
  public SpyApplication[2004] GetApplication[1856]();
  public SpyDialogs[2008] GetParent[1857]();
  public String GetFullName[1857]();
  public void SetFullName[1857]( String i_strName );
  public long[2039] GetDialogAction[1857]();
  public void SetDialogAction[1857]( long[2039] i_eAction );
}
```

29.3.5.13 SpyFindInFilesDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyFindInFilesDlg
{
  public void ReleaseInstance[2003]();
  public SpyApplication[2004] GetApplication[1858]();
  public String GetFind[1859]();
  public void SetFind[1859]( String sNewVal );
  public boolean GetRegularExpression[1860]();
  public void SetRegularExpression[1860]( boolean bNewVal );
  public String GetReplace[1861]();
  public void SetReplace[1861]( String sNewVal );
  public boolean GetReplaceOnDisk[1861]();
  public void SetReplaceOnDisk[1861]( boolean bNewVal );
  public boolean GetDoReplace[1859]();
  public void SetDoReplace[1859]( boolean bNewVal );
```


© 2016-2022 Altova GmbH

```
  public boolean GetMatchWholeWord[1860]();
  public void SetMatchWholeWord[1860]( boolean bNewVal );
  public boolean GetMatchCase[1860]();
  public void SetMatchCase[1860]( boolean bNewVal );
  public long[2041] GetSearchLocation[1861]();
  public void SetSearchLocation[1861]( long[2041] nPosition );
  public String GetStartFolder[1862]();
  public void SetStartFolder[1862]( String sNewVal );
  public boolean GetIncludeSubfolders[1859]();
  public void SetIncludeSubfolders[1859]( boolean bNewVal );
  public boolean GetSearchInProjectFilesDoExternal[1861]();
  public void SetSearchInProjectFilesDoExternal[1861]( boolean bNewVal );
  public String GetFileExtension[1859]();
  public void SetFileExtension[1859]( String sNewVal );
  public boolean GetAdvancedXMLSearch[1858]();
  public void SetAdvancedXMLSearch[1858]( boolean bNewVal );
  public boolean GetXMLElementNames[1863]();
  public void SetXMLElementNames[1863]( boolean bNewVal );
  public boolean GetXMLElementContents[1863]();
  public void SetXMLElementContents[1863]( boolean bNewVal );
  public boolean GetXMLAttributeNames[1862]();
  public void SetXMLAttributeNames[1862]( boolean bNewVal );
  public boolean GetXMLAttributeContents[1862]();
  public void SetXMLAttributeContents[1862]( boolean bNewVal );
  public boolean GetXMLComments[1863]();
  public void SetXMLComments[1863]( boolean bNewVal );
  public boolean GetXMLCData[1863]();
  public void SetXMLCData[1863]( boolean bNewVal );
  public boolean GetXMLPI[1864]();
  public void SetXMLPI[1864]( boolean bNewVal );
  public boolean GetXMLRest[1864]();
  public void SetXMLRest[1864]( boolean bNewVal );
  public boolean GetShowResult[1862]();
  public void SetShowResult[1862]( boolean bNewVal );
}
```

29.3.5.14 SpyFindInFilesMatch (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyFindInFilesMatch
{
  public void ReleaseInstance[2003]();
  public long Line[1867]();
```

```
  public long Position();
  public long Length();
  public String LineText();
  public boolean Replaced();
}
```

29.3.5.15 SpyFindInFilesResult (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyFindInFilesResult
{
  public void ReleaseInstance();
  public long Count();
  public SpyFindInFilesMatch GetItem( long nNo );
  public String GetPath();
  public SpyDoc GetDocument();
}
```

29.3.5.16 SpyFindInFilesResults (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyFindInFilesResults
{
  public void ReleaseInstance();
  public long Count();
  public SpyFindInFilesResult GetItem( long nNo );
}
```


© 2016-2022 Altova GmbH

29.3.5.17 SpyGenerateSampleXMLDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyGenerateSampleXMLDlg
{
  public void ReleaseInstance[2003]();
  public SpyApplication[2004] GetApplication[1870]();
  public boolean GetNonMandatoryAttributes[1872]();
  public void SetNonMandatoryAttributes[1872]( boolean newVal );
  public boolean GetNonMandatoryElements[1872]();
  public void SetNonMandatoryElements[1872]( boolean newVal );
  public boolean GetTakeFirstChoice[1874]();
  public void SetTakeFirstChoice[1874]( boolean newVal );
  public long GetRepeatCount[1873]();
  public void SetRepeatCount[1873]( long newVal );
  public boolean GetFillWithSampleData[1871]();
  public void SetFillWithSampleData[1871]( boolean newVal );
  public boolean GetFillElementsWithSampleData[1871]();
  public void SetFillElementsWithSampleData[1871]( boolean newVal );
  public boolean GetFillAttributesWithSampleData[1871]();
  public void SetFillAttributesWithSampleData[1871]( boolean newVal );
  public boolean GetContentOfNillableElementsIsNonMandatory[1871]();
  public void SetContentOfNillableElementsIsNonMandatory[1871]( boolean newVal );
  public boolean GetTryToUseNonAbstractTypes[1874]();
  public void SetTryToUseNonAbstractTypes[1874]( boolean newVal );
  public long[2045] GetOptimization[1873]();
  public void SetOptimization[1873]( long[2045] newVal );
  public long[2045] GetSchemaOrDTDAssignment[1874]();
  public void SetSchemaOrDTDAssignment[1874]( long[2045] newVal );
  public String GetLocalNameOfRootElement[1872]();
  public void SetLocalNameOfRootElement[1872]( String newVal );
  public String GetNamespaceURIOfRootElement[1872]();
  public void SetNamespaceURIOfRootElement[1872]( String newVal );
  public long[2039] GetOptionsDialogAction[1873]();
  public void SetOptionsDialogAction[1873]( long[2039] newVal );
}
```

29.3.5.18 SpyGridView (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyGridView
{
  public void ReleaseInstance[2003]();
  public SpyXMLData[2030] GetCurrentFocus[1877]();
  public void Deselect[1877]( SpyXMLData[2030] oData );
  public boolean GetIsVisible[1878]();
  public void Select[1878]( SpyXMLData[2030] oData );
  public void SetFocus[1878]( SpyXMLData[2030] oData );
}
```

29.3.5.19 SpyProject (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyProject
{
  public void ReleaseInstance[2003]();
  public void CloseProject[1892]( boolean bDiscardChanges, boolean bCloseFiles, boolean bDialog
  );
  public String GetProjectFile[1892]();
  public void SetProjectFile[1892]( String sFile );
  public SpyProjectItems[2019] GetRootItems[1892]();
  public void SaveProject[1892]();
  public void SaveProjectAs[1893]( String sPath, boolean bDialog );
}
```


© 2016-2022 Altova GmbH

29.3.5.20 SpyProjectItem (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyProjectItem
{
  public void ReleaseInstance[2003]();
  public SpyProjectItems[2019] GetChildItems[1893]();
  public String GetFileExtensions[1893]();
  public void SetFileExtensions[1893]( String sExtensions );
  public long[2044] GetItemType[1894]();
  public String GetName[1894]();
  public SpyDoc[2009] Open[1894]();
  public SpyProjectItem[2019] GetParentItem[1894]();
  public String GetPath[1894]();
  public String GetValidateWith[1894]();
  public void SetValidateWith[1894]( String sVal );
  public String GetXMLForXSLTransformation[1895]();
  public void SetXMLForXSLTransformation[1895]( String sVal );
  public String GetXSLForXMLTransformation[1895]();
  public void SetXSLForXMLTransformation[1895]( String sVal );
  public String GetXSLTransformationFileExtension[1895]();
  public void SetXSLTransformationFileExtension[1895]( String sVal );
  public String GetXSLTransformationFolder[1895]();
  public void SetXSLTransformationFolder[1895]( String sVal );
}
```

29.3.5.21 SpyProjectItems (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyProjectItems
{
  public void ReleaseInstance[2003]();
```

```
  public void AddFile( String sPath );
  public void AddFolder( String sName );
  public void AddURL( String sURL, long nURLType, String sUser, String sPassword,
  boolean bSave );
  public long Count();
  public SpyProjectItem GetItem( long nNumber );
  public void RemoveItem( SpyProjectItem oItemToRemove );
}
```

29.3.5.22 SpySchemaDocumentationDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
// Since version 2004R4
public class SpySchemaDocumentationDlg
{
  public void ReleaseInstance();
  public SpyApplication GetApplication();
  public SpyDialogs GetParent();

  public String GetOutputFile();
  public void SetOutputFile( String i_strVal );
  public long GetOutputFormat();
  public void SetOutputFormat( long i_eVal );

  public boolean GetShowResult();
  public void SetShowResult( boolean i_bVal );
  public long GetOptionsDialogAction();
  public void SetOptionsDialogAction( long i_eVal );
  public long GetOutputFileDialogAction();
  public void SetOutputFileDialogAction( long i_eVal );
  public boolean GetShowProgressBar();
  public void SetShowProgressBar( boolean i_bVal );

  public void IncludeAll( boolean i_bInclude );
  public boolean GetIncludeIndex();
  public void SetIncludeIndex( boolean i_bVal );
  public boolean GetIncludeGlobalElements();
  public void SetIncludeGlobalElements( boolean i_bVal );
  public boolean GetIncludeLocalElements();
  public void SetIncludeLocalElements( boolean i_bVal );
  public boolean GetIncludeGroups();
```


© 2016-2022 Altova GmbH

```
public void SetIncludeGroups( boolean i_bVal );
public boolean GetIncludeComplexTypes();
public void SetIncludeComplexTypes( boolean i_bVal );
public boolean GetIncludeSimpleTypes();
public void SetIncludeSimpleTypes( boolean i_bVal );
public boolean GetIncludeAttributeGroups();
public void SetIncludeAttributeGroups( boolean i_bVal );
public boolean GetIncludeRedefines();
public void SetIncludeRedefines( boolean i_bVal );

public void AllDetails( boolean i_bDetailsOn );
public boolean GetShowDiagram();
public void SetShowDiagram( boolean i_bVal );
public boolean GetShowNamespace();
public void SetShowNamespace( boolean i_bVal );
public boolean GetShowType();
public void SetShowType( boolean i_bVal );
public boolean GetShowChildren();
public void SetShowChildren( boolean i_bVal );
public boolean GetShowUsedBy();
public void SetShowUsedBy( boolean i_bVal );
public boolean GetShowProperties();
public void SetShowProperties( boolean i_bVal );
public boolean GetShowSingleFacets();
public void SetShowSingleFacets( boolean i_bVal );
public boolean GetShowPatterns();
public void SetShowPatterns( boolean i_bVal );
public boolean GetShowEnumerations();
public void SetShowEnumerations( boolean i_bVal );
public boolean GetShowAttributes();
public void SetShowAttributes( boolean i_bVal );
public boolean GetShowIdentityConstraints();
public void SetShowIdentityConstraints( boolean i_bVal );
public boolean GetShowAnnotations();
public void SetShowAnnotations( boolean i_bVal );
public boolean GetShowSourceCode();
public void SetShowSourceCode( boolean i_bVal );

// Since version 2009
public boolean GetEmbedDiagrams();
public void SetEmbedDiagrams( boolean i_bVal );
public long GetDiagramFormat();
public void SetDiagramFormat( long i_nVal );
public boolean GetIncludeGlobalAttributes();
public void SetIncludeGlobalAttributes( boolean i_bVal );
public boolean GetIncludeLocalAttributes();
public void SetIncludeLocalAttributes( boolean i_bVal );
public boolean GetIncludeReferencedSchemas();
public void SetIncludeReferencedSchemas( boolean i_bVal );
public boolean GetMultipleOutputFiles();
```

```
  public void SetMultipleOutputFiles[1885]( boolean i_bVal );

  // Since version 2010
  public boolean GetEmbedCSSInHTML[1880]();
  public void SetEmbedCSSInHTML[1880]( boolean i_bVal );
  public boolean GetCreateDiagramsFolder[1880]();
  public void SetCreateDiagramsFolder[1880]( boolean i_bVal );

  // Since version 2010r3
  public boolean GetGenerateRelativeLinks[1881]();
  public void SetGenerateRelativeLinks[1881]( boolean i_bVal );

  // Since version 2011r2
  public boolean GetUseFixedDesign[1891]();
  public void SetUseFixedDesign[1891]( boolean i_bVal );
  public String GetSPSFile[1891]();
  public void SetSPSFile[1891]( String i_strVal );
  }
```

29.3.5.23 SpyTextImportExportSettings (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyTextImportExportSettings
{
  public void ReleaseInstance[2003]();
  public String GetDestinationFolder[1898]();
  public void SetDestinationFolder[1898]( String sVal );
  public long[2048] GetEnclosingCharacter[1898]();
  public void SetEnclosingCharacter[1898]( long[2048] nEnclosing );
  public String GetEncoding[1898]();
  public void SetEncoding[1898]( String sVal );
  public long[2040] GetEncodingByteOrder[1898]();
  public void SetEncodingByteOrder[1898]( long[2040] nByteOrder );
  public long[2048] GetFieldDelimiter[1898]();
  public void SetFieldDelimiter[1898]( long[2048] nDelimiter );
  public String GetFileExtension[1899]();
  public void SetFileExtension[1899]( String sVal );
  public boolean GetHeaderRow[1899]();
  public void SetHeaderRow[1899]( boolean bVal );
  public String GetImportFile[1899]();
```


© 2016-2022 Altova GmbH

```
  public void SetImportFile[1899]( String sVal );
}
```

29.3.5.24 SpyTextView (obsolete)

```
public class SpyTextView
{
  public void ReleaseInstance[2003]();
  public SpyApplication[2004] GetApplication[1901]();
  public SpyDoc[2009] GetParent[1903]();
  public long LineFromPosition[1903]( long nCharPos );
  public long PositionFromLine[1904]( long nLine );
  public long LineLength[1903]( long nLine );
  public String GetSelText[1905]();
  public void SetSelText[1905]( String sText );
  public String GetRangeText[1902]( long nPosFrom, long nPosTill );
  public void ReplaceText[1904]( long nPosFrom, long nPosTill, String sText );
  public void MoveCaret[1903]( long nDiff );
  public void GoToLineChar[1902]( long nLine, long nChar );
  public void SelectText[1905]( long nPosFrom, long nPosTill );
  public long GetSelectionStart[1904]();
  public void SetSelectionStart[1904]( long nNewVal );
  public long GetSelectionEnd[1904]();
  public void SetSelectionEnd[1904]( long nNewVal );
  public String GetText[1905]();
  public void SetText[1905]( String sText );
  public long LineCount[1902]();
  public long Length[1902]();
}
```

29.3.5.25 SpyWSDL20DocumentationDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
// Since version 2010
public class SpyWSDL20DocumentationDlg
{
	public void ReleaseInstance[2003]();
	public SpyApplication[2004] GetApplication[1918]();

	public long[2039] GetOptionsDialogAction[1922]();
```

```
public void SetOptionsDialogAction( long nNewVal );

public long GetOutputFileDialogAction();
public void SetOutputFileDialogAction( long nNewVal );

public boolean GetShowProgressBar();
public void SetShowProgressBar( boolean bNewVal );

public String GetOutputFile();
public void SetOutputFile( String sNewVal );

public long GetOutputFormat();
public void SetOutputFormat( long nNewVal );

public boolean GetMultipleOutputFiles();
public void SetMultipleOutputFiles( boolean bNewVal );

public boolean GetEmbedCSSInHTML();
public void SetEmbedCSSInHTML( boolean bNewVal );

public long GetDiagramFormat();
public void SetDiagramFormat( long nNewVal );

public boolean GetEmbedDiagrams();
public void SetEmbedDiagrams( boolean bNewVal );

public boolean GetCreateDiagramsFolder();
public void SetCreateDiagramsFolder( boolean bNewVal );

public boolean GetShowResult();
public void SetShowResult( boolean bNewVal );

public void IncludeAll( boolean bNewVal );
public void AllDetails( boolean bNewVal );

public boolean GetIncludeOverview();
public void SetIncludeOverview( boolean bNewVal );

public boolean GetIncludeService();
public void SetIncludeService( boolean bNewVal );

public boolean GetIncludeBinding();
public void SetIncludeBinding( boolean bNewVal );

public boolean GetIncludeInterface();
public void SetIncludeInterface( boolean bNewVal );

public boolean GetIncludeTypes();
public void SetIncludeTypes( boolean bNewVal );
```


© 2016-2022 Altova GmbH

```
        public boolean GetIncludeImportedWSDLFiles1920();
        public void SetIncludeImportedWSDLFiles1920( boolean bNewVal );

        public boolean GetShowServiceDiagram1926();
        public void SetShowServiceDiagram1926( boolean bNewVal );

        public boolean GetShowBindingDiagram1924();
        public void SetShowBindingDiagram1924( boolean bNewVal );

        public boolean GetShowInterfaceDiagram1925();
        public void SetShowInterfaceDiagram1925( boolean bNewVal );

        public boolean GetShowTypesDiagram1927();
        public void SetShowTypesDiagram1927( boolean bNewVal );

        public boolean GetShowEndpoint1924();
        public void SetShowEndpoint1924( boolean bNewVal );

        public boolean GetShowSourceCode1926();
        public void SetShowSourceCode1926( boolean bNewVal );

        public boolean GetShowExtensibility1924();
        public void SetShowExtensibility1924( boolean bNewVal );

        public boolean GetShowUsedBy1927();
        public void SetShowUsedBy1927( boolean bNewVal );

        public boolean GetShowOperation1925();
        public void SetShowOperation1925( boolean bNewVal );

        public boolean GetShowFault1925();
        public void SetShowFault1925( boolean bNewVal );

        // Since version 2011r2
        public boolean GetUseFixedDesign1927();
        public void SetUseFixedDesign1927( boolean i_bVal );

        public String GetSPSFile1927();
        public void SetSPSFile1927( String i_strVal );

}
```

29.3.5.26 SpyWSDLDocumentationDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
// Since version 2008r2sp1
public class SpyWSDLDocumentationDlg
{
	public void ReleaseInstance[2003]();
	public SpyApplication[2004] GetApplication[1907]();

	public String GetOutputFile[1911]();
	public void SetOutputFile[1911]( String sNewVal );

	public long[2039] GetOutputFileDialogAction[1912]();
	public void SetOutputFileDialogAction[1912]( long[2039] nNewVal );

	public long[2039] GetOptionsDialogAction[1911]();
	public void SetOptionsDialogAction[1911]( long[2039] nNewVal );

	public boolean GetShowProgressBar[1914]();
	public void SetShowProgressBar[1914]( boolean bNewVal );

	public boolean GetShowResult[1915]();
	public void SetShowResult[1915]( boolean bNewVal );

	public long[2046] GetOutputFormat[1912]();
	public void SetOutputFormat[1912]( long[2046] nNewVal );

	public boolean GetEmbedDiagrams[1908]();
	public void SetEmbedDiagrams[1908]( boolean bNewVal );

	public long[2042] GetDiagramFormat[1907]();
	public void SetDiagramFormat[1907]( long[2042] nNewVal );

	public boolean GetMultipleOutputFiles[1911]();
	public void SetMultipleOutputFiles[1911]( boolean bNewVal );

	public void IncludeAll[1908]( boolean bNewVal );

	public boolean GetIncludeBinding[1909]();
	public void SetIncludeBinding[1909]( boolean bNewVal );

	public boolean GetIncludeImportedWSDLFiles[1909]();
	public void SetIncludeImportedWSDLFiles[1909]( boolean bNewVal );

	public boolean GetIncludeMessages[1909]();
	public void SetIncludeMessages[1909]( boolean bNewVal );
```


© 2016-2022 Altova GmbH

```
public boolean GetIncludeOverview();
public void SetIncludeOverview( boolean bNewVal );

public boolean GetIncludePortType();
public void SetIncludePortType( boolean bNewVal );

public boolean GetIncludeService();
public void SetIncludeService( boolean bNewVal );

public boolean GetIncludeTypes();
public void SetIncludeTypes( boolean bNewVal );

public void AllDetails( boolean bNewVal );

public boolean GetShowBindingDiagram();
public void SetShowBindingDiagram( boolean bNewVal );

public boolean GetShowExtensibility();
public void SetShowExtensibility( boolean bNewVal );

public boolean GetShowMessageParts();
public void SetShowMessageParts( boolean bNewVal );

public boolean GetShowPort();
public void SetShowPort( boolean bNewVal );

public boolean GetShowPortTypeDiagram();
public void SetShowPortTypeDiagram( boolean bNewVal );

public boolean GetShowPortTypeOperations();
public void SetShowPortTypeOperations( boolean bNewVal );

public boolean GetShowServiceDiagram();
public void SetShowServiceDiagram( boolean bNewVal );

public boolean GetShowSourceCode();
public void SetShowSourceCode( boolean bNewVal );

public boolean GetShowTypesDiagram();
public void SetShowTypesDiagram( boolean bNewVal );

public boolean GetShowUsedBy();
public void SetShowUsedBy( boolean bNewVal );

// Since version 2010
public boolean GetEmbedCSSInHTML();
public void SetEmbedCSSInHTML( boolean i_bVal );

public boolean GetCreateDiagramsFolder();
public void SetCreateDiagramsFolder( boolean i_bVal );
```

```
        // Since version 2011r2
        public boolean GetUseFixedDesign();
        public void SetUseFixedDesign( boolean i_bVal );

        public String GetSPSFile();
        public void SetSPSFile( String i_strVal );

    }
```

29.3.5.27 SpyXBRLDocumentationDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
// Since version 2010
public class SpyXBRLDocumentationDlg
{
        public void ReleaseInstance();
        public SpyApplication GetApplication();

        public long GetOptionsDialogAction();
        public void SetOptionsDialogAction( long nNewVal );

        public long GetOutputDialogAction();
        public void SetOutputDialogAction( long nNewVal );

        public boolean GetShowProgressBar();
        public void SetShowProgressBar( boolean bNewVal );

        public String GetOutputFile();
        public void SetOutputFile( String sNewVal );

        public long GetOutputFormat();
        public void SetOutputFormat( long nNewVal );

        public boolean GetEmbedCSSInHTML();
        public void SetEmbedCSSInHTML( boolean bNewVal );

        public long GetDiagramFormat();
        public void SetDiagramFormat( long nNewVal );
```


© 2016-2022 Altova GmbH

```
public boolean GetEmbedDiagrams1930();
public void SetEmbedDiagrams1930( boolean bNewVal );

public boolean GetCreateDiagramsFolder1929();
public void SetCreateDiagramsFolder1929( boolean bNewVal );

public boolean GetShowResult1937();
public void SetShowResult1937( boolean bNewVal );

public void IncludeAll1931( boolean bNewVal );
public void AllDetails1929( boolean bNewVal );

public boolean GetIncludeOverview1932();
public void SetIncludeOverview1932( boolean bNewVal );

public boolean GetIncludeNamespacePrefixes1932();
public void SetIncludeNamespacePrefixes1932( boolean bNewVal );

public boolean GetIncludeGlobalElements1931();
public void SetIncludeGlobalElements1931( boolean bNewVal );

public boolean GetIncludeDefinitionLinkroles1931();
public void SetIncludeDefinitionLinkroles1931( boolean bNewVal );

public boolean GetIncludePresentationLinkroles1932();
public void SetIncludePresentationLinkroles1932( boolean bNewVal );

public boolean GetIncludeCalculationLinkroles1931();
public void SetIncludeCalculationLinkroles1931( boolean bNewVal );

public boolean GetShowDiagram1935();
public void SetShowDiagram1935( boolean bNewVal );

public boolean GetShowSubstitutiongroup1938();
public void SetShowSubstitutiongroup1938( boolean bNewVal );

public boolean GetShowItemtype1935();
public void SetShowItemtype1935( boolean bNewVal );

public boolean GetShowBalance1935();
public void SetShowBalance1935( boolean bNewVal );

public boolean GetShowPeriod1937();
public void SetShowPeriod1937( boolean bNewVal );

public boolean GetShowAbstract1934();
public void SetShowAbstract1934( boolean bNewVal );

public boolean GetShowNillable1936();
public void SetShowNillable1936( boolean bNewVal );
```

```
        public boolean GetShowLabels();
        public void SetShowLabels( boolean bNewVal );

        public boolean GetShowReferences();
        public void SetShowReferences( boolean bNewVal );

        public boolean GetShowLinkbaseReferences();
        public void SetShowLinkbaseReferences( boolean bNewVal );

        public boolean GetShortQualifiedName();
        public void SetShortQualifiedName( boolean bNewVal );

        public boolean GetShowImportedElements();
        public void SetShowImportedElements( boolean bNewVal );

        // Since version 2011r2
        public boolean GetUseFixedDesign();
        public void SetUseFixedDesign( boolean i_bVal );

        public String GetSPSFile();
        public void SetSPSFile( String i_strVal );
};
```

29.3.5.28 SpyXMLData (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyXMLData
{
  public void ReleaseInstance();
  public void AppendChild( SpyXMLData oNewData );
  public void EraseAllChildren();
  public void EraseCurrentChild();
  public SpyXMLData GetCurrentChild();
  public SpyXMLData GetFirstChild( long nKind );
  public SpyXMLData GetNextChild();
  public boolean GetHasChildren();
  public void InsertChild( SpyXMLData oNewData );
  public boolean IsSameNode( SpyXMLData oToComp);
  public long GetKind();
  public boolean GetMayHaveChildren();
  public String GetName();
```


© 2016-2022 Altova GmbH

```
  public void SetName( String sValue );
  public SpyXMLData GetParent();
  public String GetTextValue();
  public void SetTextValue( String sValue );
}
```

29.3.5.29 Authentic (obsolete)

29.3.5.29.1 SpyAuthenticRange (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
// Since version 2004R3
public class SpyAuthenticRange
{
    public void ReleaseInstance();
    public SpyApplication GetApplication();
    public SpyAuthenticView GetParent();
    public SpyAuthenticRange GotoNext( long eKind );
    public SpyAuthenticRange GotoPrevious( long eKind );
    public void Select();
    public long GetFirstTextPosition();
    public void SetFirstTextPosition( long nTextPosition );
    public long GetLastTextPosition();
    public void SetLastTextPosition( long nTextPosition );
    public String GetText();
    public void SetText( String strText );
    public boolean PerformAction( long eAction, String strElementName );
    public boolean CanPerformAction( long eAction, String strElementName );
    public String[] CanPerformActionWith( long eAction );
    public SpyAuthenticRange GoTo( long eKind, long nCount, long nFrom );
    public SpyAuthenticRange SelectNext( long eKind );
    public SpyAuthenticRange SelectPrevious( long eKind );
    public SpyAuthenticRange MoveBegin( long eKind, long nCount );
    public SpyAuthenticRange MoveEnd( long eKind, long nCount );
    public SpyAuthenticRange ExpandTo( long eKind );
    public SpyAuthenticRange CollapsToBegin();
    public SpyAuthenticRange CollapsToEnd();
    public SpyAuthenticRange GotoNextCursorPosition();
    public SpyAuthenticRange GotoPreviousCursorPosition();
    public boolean IsEmpty();
    public boolean IsEqual( SpyAuthenticRange ipCmp );
```

```
    public SpyAuthenticRange[2031] Clone[1754]();
    public SpyAuthenticRange[2031] SetFromRange[1777]( SpyAuthenticRange[2031] ipSrc );
    public boolean Delete[1755]();
    public boolean Cut[1755]();
    public boolean Copy[1754]();
    public boolean Paste[1773]();
    public SpyXMLData[2030] GetFirstXMLData[1758]();
    public void SetFirstXMLData[1758]( SpyXMLData[2030] objXMLDataPtr );
    public long GetFirstXMLDataOffset[1759]();
    public void SetFirstXMLDataOffset[1759]( long nOffset );
    public SpyXMLData[2030] GetLastXMLData[1770]();
    public void SetLastXMLData[1770]( SpyXMLData[2050] objXMLDataPtr );
    public long GetLastXMLDataOffset[1770]();
    public void SetLastXMLDataOffset[1770]( long nOffset );
    public String[] GetElementHierarchy[1760]();
    public String[] GetElementAttributeNames[1760]( String strElementName );
    public boolean HasElementAttribute[1765]( String strElementName, String
strAttributeName );
    public String GetElementAttributeValue[1760]( String strElementName, String
strAttributeName );
    public void SetElementAttributeValue[1776]( String strElementName, String
strAttributeName, String strNewValue );
    public String[] GetEntityNames[1762]();
    public void InsertEntity[1765]( String strEntityName );
    public boolean IsInDynamicTable[1768]();
    public boolean AppendRow[1752]();
    public boolean InsertRow[1766]();
    public boolean DuplicateRow[1756]();
    public boolean DeleteRow[1755]();
    public boolean MoveRowUp[1773]();
    public boolean MoveRowDown[1772]();

    // Since version 2004R4
    public boolean IsCopyEnabled[1766]();
    public boolean IsCutEnabled[1766]();
    public boolean IsPasteEnabled[1768]();
    public boolean IsDeleteEnabled[1767]();
    public boolean IsTextStateApplied[1769]( String i_strElementName );
    public boolean IsFirstRow[1767]();
    public boolean IsLastRow[1768]();
}
```

29.3.5.29.2 SpyAuthenticView (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

© 2016-2022 Altova GmbH

```
// Since version 2004R3
public class SpyAuthenticView
{
    public void ReleaseInstance();
    public SpyApplication GetApplication();
    public SpyDoc GetParent();
    public SpyAuthenticRange GetSelection();
    public void SetSelection( SpyAuthenticRange obj );
    public SpyAuthenticRange GetDocumentBegin();
    public SpyAuthenticRange GetDocumentEnd();
    public SpyAuthenticRange GetWholeDocument();
    public long GetMarkupVisibility();
    public void SetMarkupVisibility( long eSpyAuthenticMarkupVisibility );
    public SpyAuthenticRange GoTo( long eKind, long nCount, long nFrom );
    public void Print( boolean bWithPreview, boolean bPromptUser );
    public boolean Undo();
    public boolean Redo();
    public void UpdateXMLInstanceEntities();

    // Since version 2004R4
    public String GetAsXMLString();
    public void SetAsXMLString( String i_strXML );
    public SpyXMLData GetXMLDataRoot();
    public boolean IsUndoEnabled();
    public boolean IsRedoEnabled();
}
```

29.3.5.29.3 SpyDocEditSelection (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyDocEditSelection
{
  public void ReleaseInstance();
  public SpyXMLData GetEnd();
  public long GetEndTextPosition();
  public SpyXMLData GetStart();
  public long GetStartTextPosition();
}
```

29.3.5.29.4 SpyDocEditView (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SpyDocEditView
{
  public void ReleaseInstance[2003]();
  public void ApplyTextState[1966]( String sElementName );
  public SpyDocEditSelection[2033] GetCurrentSelection[1967]();
  public void EditClear[1968]();
  public void EditCopy[1968]();
  public void EditCut[1969]();
  public void EditPaste[1969]();
  public void EditRedo[1970]();
  public void EditSelectAll[1970]();
  public void EditUndo[1970]();
  public SpyXMLData[2030] GetNextVisible[1973]( SpyXMLData[2030] oElement );
  public SpyXMLData[2030] GetPreviousVisible[1974]( SpyXMLData[2030] oElement );
  public boolean GetIsEditClearEnabled[1974]();
  public boolean GetIsEditCopyEnabled[1975]();
  public boolean GetIsEditCutEnabled[1975]();
  public boolean GetIsEditPasteEnabled[1976]();
  public boolean GetIsEditRedoEnabled[1976]();
  public boolean GetIsEditUndoEnabled[1977]();
  public boolean GetIsRowAppendEnabled[1977]();
  public boolean GetIsRowDeleteEnabled[1978]();
  public boolean GetIsRowDuplicateEnabled[1978]();
  public boolean GetIsRowInsertEnabled[1979]();
  public boolean GetIsRowMoveDownEnabled[1979]();
  public boolean GetIsRowMoveUpEnabled[1980]();
  public boolean IsTextStateApplied[1980]( String sElementName );
  public boolean IsTextStateEnabled[1981]( String sElementName );
  public void LoadXML[1981]( String sXML );
  public void MarkUpView[1982]( long nKind );
  public void RowAppend[1982]();
  public void RowDelete[1983]();
  public void RowDuplicate[1983]();
  public void RowInsert[1984]();
  public void RowMoveDown[1984]();
  public void RowMoveUp[1985]();
  public String SaveXML[1985]();
  public void SelectionMoveTabOrder[1986]( boolean bForward, boolean bTag );
```


© 2016-2022 Altova GmbH

```
  public boolean SelectionSet( SpyXMLData oStart, long nStartPos, SpyXMLData
  oEndElement, long nEndPos );
  public SpyXMLData GetXMLRoot();
  public String[] GetAllowedElements( long nAction, SpyXMLData oStartPtr,
  SpyXMLData oEndPtr );
}
```

29.3.5.30 Predefined constants (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

This section lists all classes that define the predefined constants used by the Java interface.

29.3.5.30.1 SPYApplicationStatus (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYApplicationStatus
  {
      public final static long spyApplicationStatus_Running                = 0;
      public final static long spyApplicationStatus_AfterLicenseCheck      = 1;
      public final static long spyApplicationStatus_BeforeLicenseCheck     = 2;
      public final static long                                             = 3;
      spyApplicationStatus_ConcurrentLicenseCheckFailed
      public final static long spyApplicationStatus_ProcessingCommandLine  = 4;
  }
```

29.3.5.30.2 SPYAttributeTypeDefinition (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYAttributeTypeDefinition
{
   public final static long              = 0;
   spyMergedGlobal
   public final static long              = 1;
   spyDistinctGlobal
   public final static long spyLocal     = 2;
}
```

29.3.5.30.3 SPYAuthenticActions (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYAuthenticActions
{
    public final static long spyAuthenticInsertAt   = 0;
    public final static long spyAuthenticApply      = 1;
    public final static long spyAuthenticClearSurr  = 2;
    public final static long spyAuthenticAppend     = 3;
    public final static long                        = 4;
    spyAuthenticInsertBefore
    public final static long spyAuthenticRemove     = 5;
}
```

29.3.5.30.4 SPYAuthenticDocumentPosition (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

© 2016-2022 Altova GmbH

```
public class SPYAuthenticDocumentPosition
{
    public final static long                          = 0;
    spyAuthenticDocumentBegin
    public final static long                          = 1;
    spyAuthenticDocumentEnd
    public final static long spyAuthenticRangeBegin = 2;
    public final static long spyAuthenticRangeEnd   = 3;
}
```

29.3.5.30.5 SPYAuthenticElementKind (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYAuthenticElementKind
{
   public final static long              = 0;
   spyAuthenticChar
   public final static long              = 1;
   spyAuthenticWord
   public final static long              = 3;
   spyAuthenticLine
   public final static long              = 4;
   spyAuthenticParagraph
   public final static long              = 6;
   spyAuthenticTag
   public final static long              = 8;
   spyAuthenticDocument
   public final static long              = 9;
   spyAuthenticTable
   public final static long              = 10;
   spyAuthenticTableRow
   public final static long              = 11;
   spyAuthenticTableColumn
}
```

29.3.5.30.6 SPYAuthenticMarkupVisibility (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYAuthenticMarkupVisibility
{
    public final static long                          = 0;
    spyAuthenticMarkupHidden
    public final static long spyAuthenticMarkupSmall= 1;
    public final static long spyAuthenticMarkupLarge= 2;
    public final static long spyAuthenticMarkupMixed= 3;
}
```

29.3.5.30.7 SPYDatabaseKind (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYLoading
  {
    public final static long spyDB_Access   = 0;
    public final static long                = 1;
    spyDB_SQLServer
    public final static long spyDB_Oracle   = 2;
    public final static long spyDB_Sybase   = 3;
    public final static long spyDB_MySQL    = 4;
    public final static long spyDB_DB2      = 5;
    public final static long spyDB_Other    = 6;
    public final static long                = 7;
    spyDB_Unspecified
  }
```


© 2016-2022 Altova GmbH

29.3.5.30.8 SPYDialogAction (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYDialogAction
  {
     public final static long spyDialogOK     = 0;
     public final static long                 = 1;
     spyDialogCancel
     public final static long                 = 2;
     spyDialogUserInput
  }
```

29.3.5.30.9 SPYDOMType (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYDOMType
  {
     public final static long               = 0;
     spyDOMType_msxml4
     public final static long               = 1;
     spyDOMType_xerces
  }
```

29.3.5.30.10 SPYDTDSchemaFormat (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYDTDSchemaFormat
{
    public final static long spyDTD     = 0;
    public final static long spyDCD     = 1;
    public final static long spyXMLData = 2;
    public final static long spyBizTalk = 3;
    public final static long spyW3C     = 4;
}
```

29.3.5.30.11 SPYEncodingByteOrder (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYEncodingByteOrder
{
    public final static long spyNONE          = 0;
    public final static long spyLITTLE_ENDIAN = 1;
    public final static long spyBIG_ENDIAN    = 2;
}
```

29.3.5.30.12 SPYExportNamespace (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYExportNamespace
{
    public final static long spyNoNamespace                = 0;
    public final static long spyReplaceColonWithUnderscore = 1;
```


© 2016-2022 Altova GmbH

```
}
```

29.3.5.30.13 SPYFindInFilesSearchLocation (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYFindInFilesSearchLocation
{
   public final static long                    = 0;
   spyFindInFiles_Documents
   public final static long                    = 1;
   spyFindInFiles_Project
   public final static long                    = 2;
   spyFindInFiles_Folder
}
```

29.3.5.30.14 SPYFrequentElements (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYFrequentElements
{
   public final static long              = 0;
   spyGlobalElements
   public final static long              = 1;
   spyGlobalComplexType
}
```

29.3.5.30.15 SPYImageKind (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYImageKind
  {
     public final static long                = 0;
     spyImageType_PNG
     public final static long                = 1;
     spyImageType_EMF
  }
```

29.3.5.30.16 SPYImportColumnsType (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

29.3.5.30.17 SPYLibType (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYLibType
  {
     public final static long                = 0;
     spyLibType_static
     public final static long spyLibType_dll= 1;
```


© 2016-2022 Altova GmbH

```
}
```

29.3.5.30.18 SPYLoading (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYLoading
{
    public final static long               = 0;
    spyUseCacheProxy
    public final static long spyReload     = 1;
}
```

29.3.5.30.19 SPYNumberDateTimeFormat (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class  SPYNumberDateTimeFormat
{
    public final static long spySystemLocale = 0;
    public final static long                 = 1;
    spySchemaCompatible
}
```

29.3.5.30.20 SPYProgrammingLanguage (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYLoading
  {
      public final static long spyUndefinedLanguage = -1;
      public final static long spyJava               = 0;
      public final static long spyCpp                = 1;
      public final static long spyCSharp             = 2;
  }
```

29.3.5.30.21 SPYProjectItemTypes (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYProjectItemTypes
{
    public final static long            = 0;
    spyUnknownItem
    public final static long spyFileItem= 1;
    public final static long            = 2;
    spyFolderItem
    public final static long spyURLItem = 3;
}
```

29.3.5.30.22 SPYProjectType (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYProjectType
  {
```


© 2016-2022 Altova GmbH

```
    public final static long                    = 0;
    spyVisualStudioProject
    public final static long                    = 1;
    spyVisualStudio2003Project
    public final static long spyBorlandProject = 2;
    public final static long spyMonoMakefile   = 3;
 }
```

29.3.5.30.23 SPYSampleXMLGenerationOptimization (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYSampleXMLGenerationOptimization
{
   public final static long spySampleXMLGen_Optimized  = 0;
   public final static long                            = 1;
   spySampleXMLGen_NonMandatoryElements
   public final static long spySampleXMLGen_Everything = 2;
}
```

29.3.5.30.24 SPYSampleXMLGenerationSchemaOrDTDAssignment (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYSampleXMLGenerationOptimization
{
   public final static long                            = 0;
   spySampleXMLGen_AssignRelatively
   public final static long                            = 1;
   spySampleXMLGen_AssignAbsolutely
   public final static long spySampleXMLGen_DoNotAssign= 2;
}
```

29.3.5.30.25 SPYSchemaDefKind (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYSchemaDefKind
{
    public final static long spyKindElement       = 0;
    public final static long spyKindComplexType   = 1;
    public final static long spyKindSimpleType    = 2;
    public final static long spyKindGroup         = 3;
    public final static long spyKindModel         = 4;
    public final static long spyKindAny           = 5;
    public final static long spyKindAttr          = 6;
    public final static long spyKindAttrGroup     = 7;
    public final static long spyKindAttrAny       = 8;
    public final static long spyKindIdentityUnique= 9;
    public final static long spyKindIdentityKey   = 10;
    public final static long spyKindIdentityKeyRef= 11;
    public final static long                      = 12;
    spyKindIdentitySelector
    public final static long spyKindIdentityField = 13;
    public final static long spyKindNotation      = 14;
    public final static long spyKindInclude       = 15;
    public final static long spyKindImport        = 16;
    public final static long spyKindRedefine      = 17;
    public final static long spyKindFacet         = 18;
    public final static long spyKindSchema        = 19;
    public final static long spyKindCount         = 20;
}
```

29.3.5.30.26 SPYSchemaDocumentationFormat (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYSchemaDocumentationFormat
```


© 2016-2022 Altova GmbH

```
  {
     public final static long                 = 0;
     spySchemaDoc_HTML
     public final static long                 = 1;
     spySchemaDoc_MSWord
     public final static long                 = 2;
            spySchemaDoc_RTF
     public final static long                 = 3;
            spySchemaDoc_PDF
  }
```

29.3.5.30.27 SPYSchemaExtensionType (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYSchemaExtensionType
  {
     public final static long spySchemaExtension_None    = 0;
     public final static long spySchemaExtension_SQL_XML = 1;
     public final static long                            = 2;
     spySchemaExtension_MS_SQL_Server
     public final static long spySchemaExtension_Oracle  = 3;
  }
```

29.3.5.30.28 SPYSchemaFormat (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYSchemaFormat
  {
     public final static long                       = 0;
     spySchemaFormat_Hierarchical
     public final static long spySchemaFormat_Flat  = 1;
  }
```

29.3.5.30.29 SPYTextDelimiters (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYTextDelimiters
{
    public final static long spyTabulator = 0;
    public final static long spySemicolon = 1;
    public final static long spyComma     = 2;
    public final static long spySpace     = 3;
}
```

29.3.5.30.30 SPYTextEnclosing (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYTextEnclosing
{
   public final static long           = 0;
   spyNoEnclosing
   public final static long           = 1;
   spySingleQuote
   public final static long           = 2;
   spyDoubleQuote
}
```


© 2016-2022 Altova GmbH

29.3.5.30.31 SPYTypeDetection (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYTypeDetection
{
    public final static long                = 0;
    spyBestPossible
    public final static long                = 1;
    spyNumbersOnly
    public final static long                = 2;
    spyNoDetection
}
```

29.3.5.30.32 SPYURLTypes (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYURLTypes
{
    public final static long              = (-1);
    spyURLTypeAuto
    public final static long              = 0;
    spyURLTypeXML
    public final static long              = 1;
    spyURLTypeDTD
}
```

29.3.5.30.33 SpyViewModes (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYViewModes
{
    public final static long spyViewGrid     = 0;
    public final static long spyViewText     = 1;
    public final static long spyViewBrowser  = 2;
    public final static long spyViewSchema   = 3;
    public final static long spyViewContent  = 4;
    public final static long spyViewAuthentic= 4;
    public final static long spyViewWSDL     = 5;
    public final static long spyViewZIP      = 6;
    public final static long                 = 7;
    spyViewEditionInfo
}
```

29.3.5.30.34 SPYWhitespaceComparison (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYWhitespaceComparison
  {
     public final static long spyCompareAsIs  = 0;
     public final static long                 = 1;
     spyCompareNormalized
     public final static long spyStripAll     = 2;
  }
```


© 2016-2022 Altova GmbH

29.3.5.30.35 SPYXMLDataKind (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java[1711].

```
public class SPYXMLDataKind
{
   public final static long                  = 0;
   spyXMLDataXMLDocStruct
   public final static long                  = 1;
   spyXMLDataXMLEntityDocStruct
   public final static long                  = 2;
   spyXMLDataDTDDocStruct
   public final static long spyXMLDataXML    = 3;
   public final static long                  = 4;
   spyXMLDataElement
   public final static long spyXMLDataAttr   = 5;
   public final static long spyXMLDataText   = 6;
   public final static long spyXMLDataCData  = 7;
   public final static long                  = 8;
   spyXMLDataComment
   public final static long spyXMLDataPI     = 9;
   public final static long                  = 10;
   spyXMLDataDefDoctype
   public final static long                  = 11;
   spyXMLDataDefExternalID
   public final static long                  = 12;
   spyXMLDataDefElement
   public final static long                  = 13;
   spyXMLDataDefAttlist
   public final static long                  = 14;
   spyXMLDataDefEntity
   public final static long                  = 15;
   spyXMLDataDefNotation
   public final static long                  = 16;
   spyXMLDataKindsCount
}
```

29.4 ActiveX Integration

The XMLSpy user interface and the functionality described in this section can be integrated into custom applications that can consume ActiveX controls. ActiveX technology enables a wide variety of languages to be used for integration, such as C++, C#, VB.NET, HTML. (Note that ActiveX components integrated in HTML must be run with Microsoft Internet Explorer versions and platforms that support ActiveX). All components are full OLE Controls. Integration into Java is provided through wrapper classes.

> To integrate the ActiveX controls into your custom code, the XMLSpy Integration Package must be installed (see https://www.altova.com/components/download). Ensure that you install XMLSpy first, and then the XMLSpy Integration Package. Other prerequisites apply, depending on language and platform (see Prerequisites 2052).

You can flexibly choose between two different levels of integration: application level and document level.

Integration at application level means embedding the complete interface of XMLSpy (including its menus, toolbars, panes, etc) as an ActiveX control into your custom application. For example, in the most simple scenario, your custom application could consist of only one form that embeds the XMLSpy graphical user interface. This approach is easier to implement than integration at document level but may not be suitable if you need flexibility to configure the XMLSpy graphical user interface according to your custom requirements.

Integration at document level means embedding XMLSpy into your own application piece-by-piece. This includes implementing not only the main XMLSpy control but also the main document editor window, and, optionally, any additional windows. This approach provides greater flexibility to configure the GUI, but requires advanced interaction with ActiveX controls in your language of choice.

The sections Integration at the Application Level 2055 and Integration at Document Level 2057 describe the key steps at these respective levels. The ActiveX Integration Examples 2060 section provides examples in C#, HTML, and Java. Looking through these examples will help you to make the right decisions quickly. The Object Reference 2091 section describes all COM objects that can be used for integration, together with their properties and methods.

For information about using XMLSpy as a Visual Studio plug-in, see XMLSpy in Visual Studio 1123.

29.4.1 Prerequisites

To integrate the XMLSpy ActiveX control into a custom application, the following must be installed on your computer:

- XMLSpy
- The XMLSpy Integration Package, available for download at https://www.altova.com/components/download

To integrate the 64-bit ActiveX control, install the 64-bit versions of XMLSpy and XMLSpy Integration Package. For applications developed under Microsoft .NET platform with Visual Studio, both the 32-bit and 64-bit versions of XMLSpy and XMLSpy Integration Package must be installed, as explained below.

© 2016-2022 Altova GmbH

Microsoft .NET (C#, VB.NET) with Visual Studio

To integrate the XMLSpy ActiveX control into a 32-bit application developed under Microsoft .NET, the following must be installed on your computer:

- Microsoft .NET Framework 4.0 or later
- Visual Studio 2012/2013/2015/2017/2019/2022
- XMLSpy 32-bit and XMLSpy Integration Package 32-bit
- The ActiveX controls must be added to the Visual Studio toolbox (see Adding the ActiveX Controls to the Toolbox 2053).

If you want to integrate the 64-bit ActiveX control, the following prerequisites apply in addition to the ones above:

- XMLSpy 32-bit and XMLSpy Integration Package 32-bit must still be installed (this is required to provide the 32-bit ActiveX control to the Visual Studio designer, since Visual Studio runs on 32-bit)
- XMLSpy 64-bit and XMLSpy Integration Package 64-bit must be installed (provides the actual 64-bit ActiveX control to your custom application at runtime)
- In Visual Studio, create a 64-bit build configuration and build your application using this configuration. For an example, see Running the Sample C# Solution 2060.

Java

To integrate the XMLSpy ActiveX control into Java application using the Eclipse development environment, the following must be installed on your computer:

- Java Runtime Environment (JRE) or Java Development Kit (JDK) 7 or later
- Eclipse
- XMLSpy and XMLSpy Integration Package

Note: To run the 64-bit version of the XMLSpy ActiveX control, use a 64-bit version of Eclipse, as well as the 64-bit version of XMLSpy and the XMLSpy Integration Package.

XMLSpy integration and deployment on client computers

If you create a .NET application and intend to distribute it to other clients, you will need to install the following on the client computer(s):

- XMLSpy
- The XMLSpy Integration Package
- The custom integration code or application.

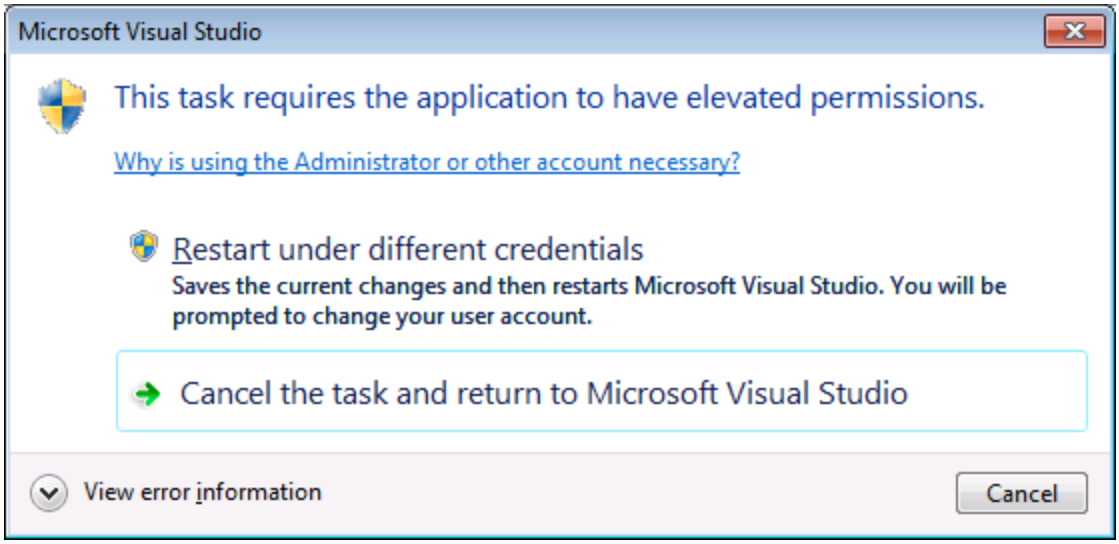
29.4.2 Adding the ActiveX Controls to the Toolbox

To use the XMLSpy ActiveX controls in an application developed with Visual Studio, the controls must first be added to the Visual Studio Toolbox, as follows:

1. On the **Tools** menu of Visual Studio, click **Choose Toolbox Items**.
2. On the **COM Components** tab, select the check boxes next to the XMLSpyControl, XMLSpyControl Document, and XMLSpyControl Placeholder.

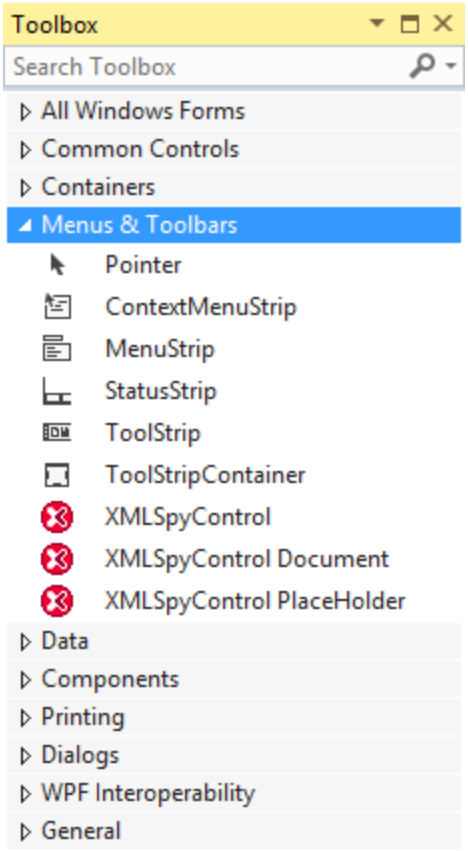
In case the controls above are not available, follow the steps below:

1. On the **COM Components** tab, click **Browse**, and select the **XMLSpyControl.ocx** file from the XMLSpy installation folder. Remember that the XMLSpy Integration Package must be installed; otherwise, this file is not available, see Prerequisites 2052.
2. If prompted to restart Visual Studio with elevated permissions, click **Restart under different credentials**.



If the steps above were successful, the XMLSpy ActiveX controls become available in the Visual Studio Toolbox.

© 2016-2022 Altova GmbH



Note: For an application-level integration, only the **XMLSpyControl** ActiveX control is used (see Integration at Application Level[2055]). The **XMLSpyControl Document** and **XMLSpyControl Placeholder** controls are used for document-level integration (see Integration at Document Level[2057]).

29.4.3 Integration at Application Level

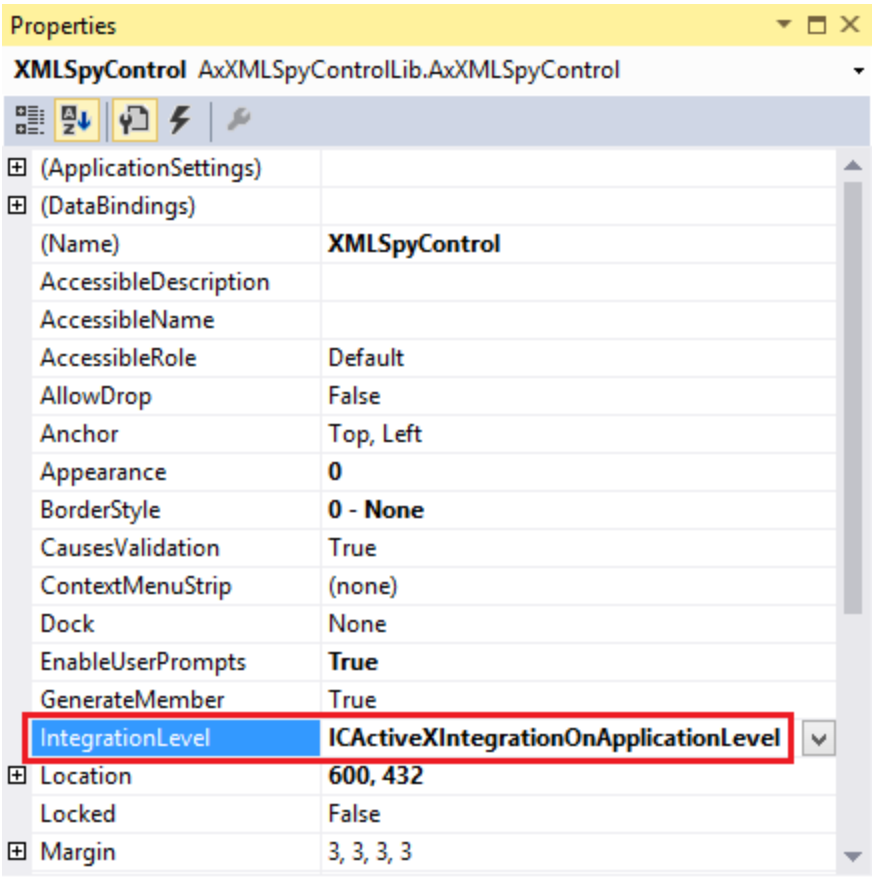
Integration at application level allows you to embed the complete interface of XMLSpy into a window of your application. With this type of integration, you get the whole user interface of XMLSpy, including all menus, toolbars, the status bar, document windows, and helper windows. Customization of the application's user interface is restricted to what XMLSpy provides. This includes rearrangement and resizing of helper windows and customization of menus and toolbars.

The only ActiveX control you need to integrate is `XMLSpyControl`[2095]. Do not instantiate or access `XMLSpyControlDocument`[2102] or `XMLSpyControlPlaceHolder`[2109] ActiveX controls when integrating at application-level.

If you have any initialization to do or if you want to automate some behaviour of XMLSpy, use the properties, methods, and events described for `XMLSpyControl`[2095]. Consider using `XMLSpyControl.Application`[2096] for more complex access to XMLSpy functionality.

In C# or VB.NET with Visual Studio, the steps to create a basic, one-form application which integrates the XMLSpy ActiveX controls at application level are as follows:

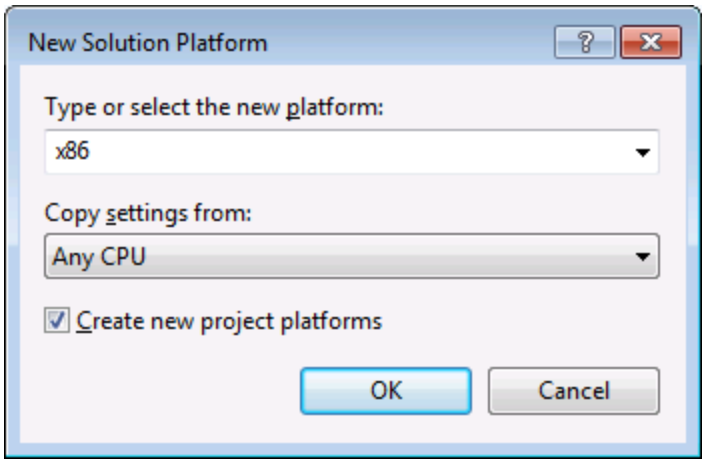
1. Check that all prerequisites are met (see Prerequisites 2052).
2. Create a new Visual Studio Windows Forms project with a new empty form.
3. If you have not done that already, add the ActiveX controls to the toolbox (see Adding the ActiveX Controls to the Toolbox 2053).
4. Drag the **XMLSpyControl** from the toolbox onto your new form.
5. Select the **XMLSpyControl** on the form, and, in the Properties window, set the **IntegrationLevel** property to **ICActiveXIntegrationOnApplicationLevel**.



6. Create a build platform configuration that matches the platform under which you want to build (x86, x64). Here is how you can create the build configuration:

   a. Right-click the solution in Visual Studio, and select **Configuration Manager**.
   b. Under **Active solution platform**, select **New...** and then select the x86 or x64 configuration (in this example, **x86**).

© 2016-2022 Altova GmbH



You are now ready to build and run the solution in Visual Studio. Remember to build using the configuration that matches your target platform (x86, x64).

29.4.4 Integration at Document Level

Compared to integration at application level, integration at document level is a more complex, yet more flexible way to embed XMLSpy functionality into your application by means of ActiveX controls. With this approach, your code can access selectively the following parts of the XMLSpy user interface:

- Document editing window
- Project window
- Entry helper windows
- Validator output window
- XPath profiler window
- XPath dialog window
- XSLT/XQuery debugger windows
- SOAP debugger window

As mentioned in Integration at Application Level[2055], for an ActiveX integration at application level, only one control is required, namely the **XMLSpyControl**. However, for an ActiveX integration at document level, XMLSpy functionality is provided by the following ActiveX controls:

- XMLSpyControl[2095]
- XMLSpyControl Document[2102]
- XMLSpyControl Placeholder[2109]

These controls are supplied by the **XMLSpyControl.ocx** file available in the application installation folder of XMLSpy. When you develop the ActiveX integration with Visual Studio, you will need to add these controls to the Visual Studio toolbox (see Adding the ActiveX Controls to the Toolbox[2053]).

The basic steps to integrate the ActiveX controls at document level into your application are as follows:

1. First, instantiate **XMLSpyControl** in your application. Instantiating this control is mandatory; it enables support for the **XMLSpyControl Document** and **XMLSpyControl Placeholder** controls mentioned above. It is important to set the `IntegrationLevel`[2097] property to

ICActiveXIntegrationOnDocumentLevel (or "1"). To hide the control from the user, set its **Visible** property to **False**.

Note: When integrating at document level, do not use the **Open** method of the **XMLSpyControl**; this might lead to unexpected results. Use the corresponding open methods of **XMLSpyControl Document** and **XMLSpyControl PlaceHolder** instead.

2. Create at least one instance of **XMLSpyControl Document** in your application. This control supplies the document editing window of XMLSpy to your application and can be instantiated multiple times if necessary.

   Use the method **Open** to load any existing file. To access document-related functionality, use the **Path** and **Save** or methods and properties accessible via the property **Document**.

Note: The control does not support a read-only mode. The value of the property **ReadOnly** is ignored.

3. Optionally, add to your application the **XMLSpyControl Placeholder** control for each additional window (other than the document window) that must be available to your application.

   Instances of **XMLSpyControl PlaceHolder** allow you to selectively embed additional windows of XMLSpy into your application. The window kind (for example, Project window) is defined by the property **PlaceholderWindowID**. Therefore, to set the window kind, set the property **PlaceholderWindowID**. For valid window identifiers, see XMLSpyControlPlaceholderWindow 2112.

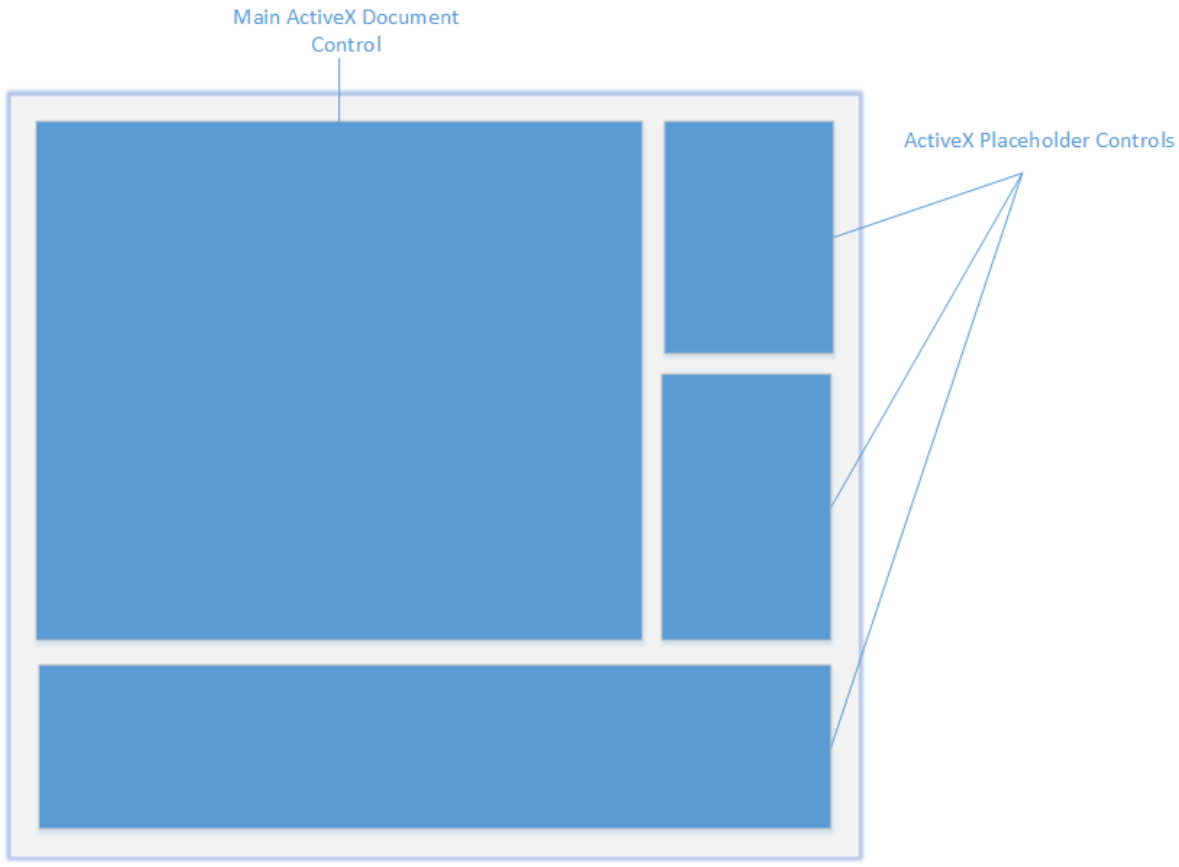
Note: Use only one **XMLSpyControl PlaceHolder** for each window identifier.

   For placeholder controls that select the XMLSpy project window, additional methods are available. Use **OpenProject** to load a XMLSpy project. Use the property Project and the methods and properties from the XMLSpy automation interface to perform any other project related operations.

For example, in C# or VB.NET with Visual Studio, the steps to create a basic, one-form application which integrates the XMLSpy ActiveX controls at document level could be similar to those listed below. Note that your application may be more complex if necessary; however, the instructions below are important to understand the minimum requirements for an ActiveX integration at document level.

1. Create a new Visual Studio Windows Forms project with a new empty form.
2. If you have not done that already, add the ActiveX controls to the toolbox (see Adding the ActiveX Controls to the Toolbox 2053).
3. Drag the **XMLSpyControl** 2095 from the toolbox onto your new form.
4. Set the **IntegrationLevel** property of the **XMLSpyControl** to **ICActiveXIntegrationOnDocumentLevel**, and the **Visible** property to **False**. You can do this either from code or from the **Properties** window.
5. Drag the **XMLSpyControl Document** 2102 from the toolbox onto the form. This control provides the main document window of XMLSpy to your application, so you may need to resize it to a reasonable size for a document.
6. Optionally, add one or more **XMLSpyControl Placeholder** 2109 controls to the form (one for each additional window type that your application needs, for example, the **Project** window). You will typically want to place such additional placeholder controls either below or to the right or left of the main document control, for example:

© 2016-2022 Altova GmbH



7. Set the **PlaceholderWindowID** property of each **XMLSpyControl Placeholder** control to a valid window identifier. For the list of valid values, see XMLSpyControlPlaceholderWindow[2112].
8. Add commands to your application (at minimum, you will need to open, save and close documents), as shown below.

Querying XMLSpy Commands

When you integrate at document level, no XMLSpy menu or toolbar is available to your application. Instead, you can retrieve the required commands, view their status, and execute them programmatically, as follows:

- To retrieve all available commands, use the `CommandsList`[2096] property of the **XMLSpyControl**.
- To retrieve commands organized according to their menu structure, use the `MainMenu`[2097] property.
- To retrieve commands organized by the toolbar in which they appear, use the `Toolbars`[2098] property.
- To send commands to XMLSpy, use the `Exec`[2099] method.
- To query if a command is currently enabled or disabled, use the `QueryStatus`[2099] method.

This enables you to flexibly integrate XMLSpy commands into your application's menus and toolbars.

Your installation of XMLSpy also provides you with command label images used within XMLSpy. See the folder **<ApplicationFolder>\Examples\ActiveX\Images** of your XMLSpy installation for icons in GIF format. The file names correspond to the command names as they are listed in the Command Reference[2072] section.

General considerations

To automate the behaviour of XMLSpy, use the properties, methods, and events described for the **XMLSpyControl**[2095], **XMLSpyControl Document**[2102], and **XMLSpyControl Placeholder**[2109].

For more complex access to XMLSpy functionality, consider using the following properties:

- `XMLSpyControl.Application`[2096]
- `XMLSpyControlDocument.Document`[2104]
- `XMLSpyControlPlaceHolder.Project`[2110]

These properties give you access to the XMLSpy automation interface (XMLSpyAPI)

Note: To open a document, always use `XMLSpyControlDocument.Open`[2105] or `XMLSpyControlDocument.New`[2105] on the appropriate document control. To open a project, always use `XMLSpyControlPlaceHolder.OpenProject`[2110] on a placeholder control embedding a XMLSpy project window.

For examples that show how to instantiate and access the necessary controls in different programming environments, see ActiveX Integration Examples[2060].

29.4.5 ActiveX Integration Examples

This section contains examples of XMLSpy document-level integration using different container environments and programming languages. Source code for all examples is available in the folder `<ApplicationFolder>\Examples\ActiveX` of your XMLSpy installation.

29.4.5.1 C#

A basic ActiveX integration example solution for C# and Visual Studio is available in the folder `<ApplicationFolder>\Examples\ActiveX\C#`. Before you compile the source code and run the sample, make sure that all prerequisites are met (see Running the Sample C# Solution[2060]).

29.4.5.1.1 Running the Sample C# Solution

The sample Visual Studio solution available in the folder **<ApplicationFolder>\Examples\ActiveX\C#** illustrates how to consume the XMLSpy ActiveX controls. Before attempting to build and run this solution, note the following steps:

Step 1: Check the prerequisites

Visual Studio 2010 or later is required to open the sample solution. For the complete list of prerequisites, see Prerequisites[2052].

Step 2: Copy the sample to a directory where you have write permissions

To avoid running Visual Studio as an Administrator, copy the source code to a directory where you have write permissions, instead of running it from the default location.

© 2016-2022 Altova GmbH

Step 3: Check and set all required control properties

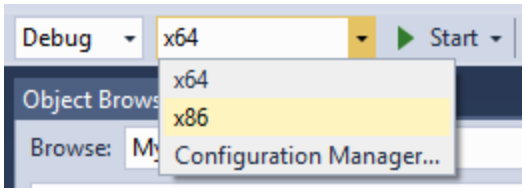
The sample application contains one instance of XMLSpyControlDocument[2102] and one instance of XMLSpyControlPlaceHolder[2109] controls. Double-check that the following properties of these controls are set as shown in the table below:

Control name	Property	Property value
XMLSpyControl	IntegrationLevel	ICActiveXIntegrationOnDocumentLevel
XPathDialog	PlaceholderWindowID	16

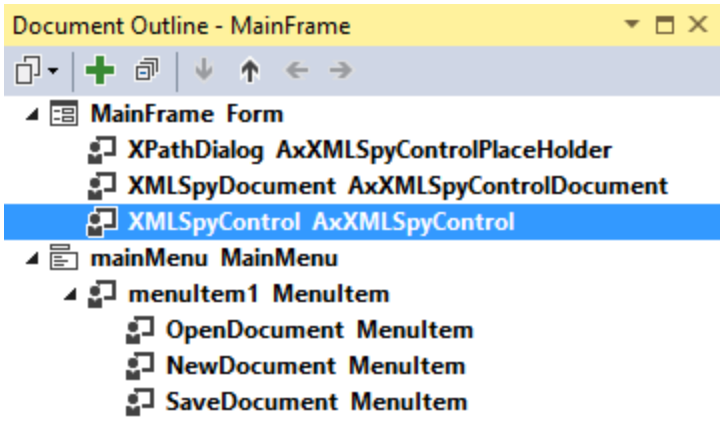
Here is how you can view or set the properties of an ActiveX control:

1. Open the **MDIMain.cs** form in the designer window.

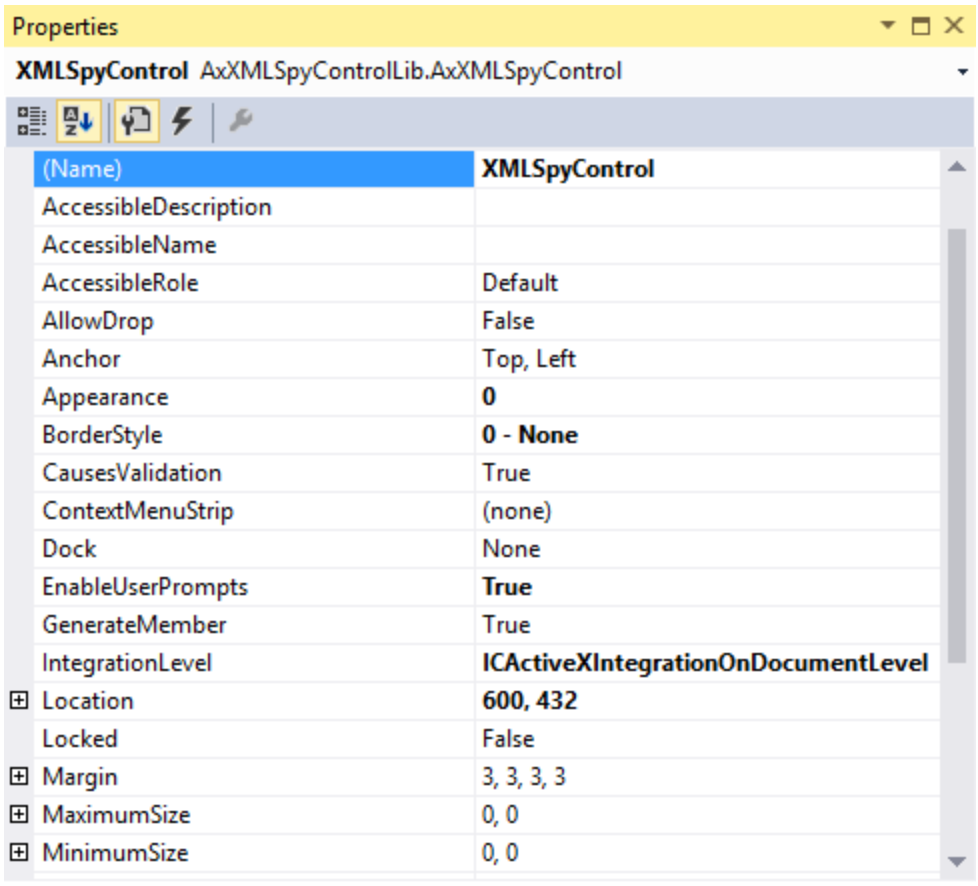
Note: On 64-bit Windows, it may be necessary to change the build configuration of the Visual Studio solution to "x86" **before** opening the designer window. If you need to build the sample as a 64-bit application, see Prerequisites[2052].



2. Open the **Document Outline** window of Visual Studio (On the **View** menu, click **Other Windows | Document Outline**).



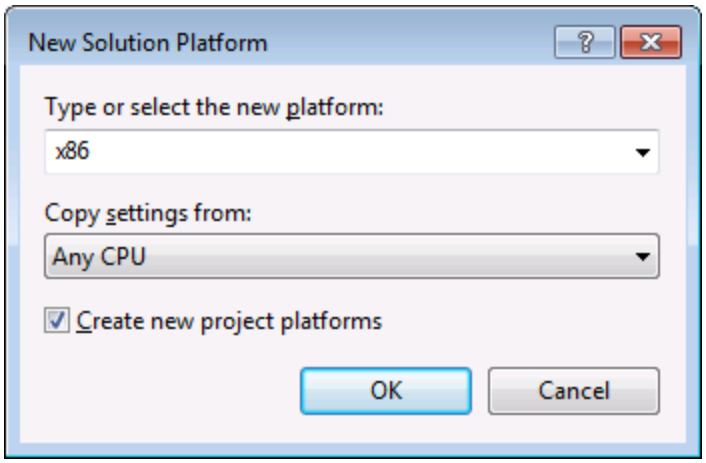
3. Click an ActiveX control in the **Document Outline** window, and edit its required property in the **Properties** window, for example:



Step 4: Set the build platform

- Create a build platform configuration that matches the platform under which you want to build (x86, x64). Here is how you can create the build configuration:

  a. Right-click the solution in Visual Studio, and select **Configuration Manager**.
  b. Under **Active solution platform**, select **New...** and then select the x86 or x64 configuration (in this example, **x86**).



 © 2016-2022 Altova GmbH

You are now ready to build and run the solution in Visual Studio. Remember to build using the configuration that matches your target platform (x86, x64); otherwise, runtime errors might occur.

On running the sample, the main MDI Frame window is created and contains an editing window with an empty XML document and a XPath Dialog window of XMLSpy at the bottom. Use **File | Open** to open any XML file from the XMLSpy examples folder. The file is loaded and displayed. After you load the document, you can start using the XPath dialog. Note that you may need to slightly drag the lower-right corner of the form to cause the dialog to redraw itself and display its contents.

29.4.5.2 Java

XMLSpy ActiveX components can be accessed from Java code. Java integration is provided by the libraries listed below. These libraries are available in the folder `<ApplicationFolder>\Examples\JavaAPI` of your XMLSpy installation, after you have installed both XMLSpy and the XMLSpy Integration Package (see also Prerequisites 2052).

- `AltovaAutomation.dll`: a JNI wrapper for Altova automation servers (in case of the 32-bit installation of XMLSpy)
- `AltovaAutomation_x64.dll`: a JNI wrapper for Altova automation servers (in case of the 64-bit installation of XMLSpy)
- `AltovaAutomation.jar`: Java classes to access Altova automation servers
- `XMLSpyActiveX.jar`: Java classes that wrap the XMLSpy ActiveX interface
- `XMLSpyActiveX_JavaDoc.zip`: a Javadoc file containing help documentation for the Java interface

Note: In order to use the Java ActiveX integration, the .dll and .jar files must be included in the Java class search path.

Example Java project

An example Java project is supplied with your product installation. You can test the Java project and modify and use it as you like. For more details, see Example Java Project 2064.

Rules for mapping the ActiveX Control names to Java

For the documentation of ActiveX controls, see Object Reference 2091. Note that the object naming conventions are slightly different in Java compared to other languages. Namely, the rules for mapping between the ActiveX controls and the Java wrapper are as follows:

- ***Classes and class names***
  For every component of the XMLSpy ActiveX interface a Java class exists with the name of the component.

- ***Method names***
  Method names on the Java interface are the same as used on the COM interfaces but start with a small letter to conform to Java naming conventions. To access COM properties, Java methods that prefix the property name with get and set can be used. If a property does not support write-access, no setter method is available. Example: For the `IntegrationLevel` property of the XMLSpyControl, the Java methods `getIntegrationLevel` and `setIntegrationLevel` are available.

- ***Enumerations***
  For every enumeration defined in the ActiveX interface, a Java enumeration is defined with the same name and values.

- ***Events and event handlers***
  For every interface in the automation interface that supports events, a Java interface with the same name plus 'Event' is available. To simplify the overloading of single events, a Java class with default implementations for all events is provided. The name of this Java class is the name of the event interface plus '`DefaultHandler`'. For example:
  `XMLSpyControl`: Java class to access the application
  `XMLSpyControlEvents`: Events interface for the XMLSpyControl
  `XMLSpyControlEventsDefaultHandler`: Default handler for `XMLSpyControlEvents`

Exceptions to mapping rules

There are some exceptions to the rules listed above. These are listed below:

Interface	Java name
`XMLSpyControlDocument`, method `New`	`newDocument`
`Document`, method `SetEncoding`	`setFileEncoding`
`AuthenticView`, method `Goto`	`gotoElement`
`AuthenticRange`, method `Goto`	`gotoElement`
`AuthenticRange`, method `Clone`	`cloneRange`

This section

This section shows how some basic XMLSpy ActiveX functionality can be accessed from Java code. It is organized into the following sub-sections:

- Example Java Project 2064
- Creating the ActiveX Controls 2066
- Loading Data in the Controls 2067
- Basic Event Handling 2067
- Menus 2068
- UI Update Event Handling 2070
- Creating an XML Tree 2070

29.4.5.2.1 Example Java Project

The XMLSpy installation package contains an example Java project, located in the ActiveX Examples folder of the application folder: `<ApplicationFolder>\Examples\ActiveX\Java\`.

The Java example shows how to integrate the XMLSpyControl in a common desktop application created with Java. You can test it directly from the command line using the batch file `BuildAndRun.bat`, or you can compile and run the example project from within Eclipse. See below for instructions on how to use these procedures.

© 2016-2022 Altova GmbH

File list

The Java examples folder contains all the files required to run the example project. These files are listed below:

`.classpath`	Eclipse project helper file
`.project`	Eclipse project file
`AltovaAutomation.dll`	Java-COM bridge: DLL part (for the 32-bit installation)
`AltovaAutomation_x64.dll`	Java-COM bridge: DLL part (for the 64-bit installation)
`AltovaAutomation.jar`	Java-COM bridge: Java library part
`BuildAndRun.bat`	Batch file to compile and run example code from the command line prompt. Expects folder where Java Virtual Machine resides as parameter.
`XMLSpyActiveX.jar`	Java classes of the XMLSpy ActiveX control
`XMLSpyActiveX_JavaDoc.zip`	Javadoc file containing help documentation for the Java API
`XMLSpyContainer.java`	Java example source code
`XMLSpyContainerEventHandler.java`	Java example source code
`XMLTreeDialog.java`	Java example source code

What the example does

The example places one XMLSpy document editor window, the XMLSpy project window, the XMLSpy XPath window and an XMLSpy entry helper in an AWT frame window. It reads out the File menu defined for XMLSpy and creates an AWT menu with the same structure. You can use this menu or the project window to open and work with files in the document editor.

You can modify the example in any way you like.

The following specific features are described in code listings:

- Creating the ActiveX Controls 2066: Starts XMLSpy, which is registered as an automation server, or activates XMLSpy if it is already running.
- Loading Data in the Controls 2067: Locates one of the example documents installed with XMLSpy and opens it.
- Basic Event Handling 2067: Changes the view of all open documents to Text View. The code also shows how to iterate through open documents.
- Menus 2068: Validates the active document and shows the result in a message box. The code shows how to use output parameters.
- UI Update Event Handling 2070: Shows how to handle XMLSpy events.
- Creating an XML Tree 2070: Shows how to create an XML tree and prepare it for modal activation.

Updating the path to the Examples folder

Before running the provided sample, you may need to edit the **XMLSpyContainer.java** file. Namely, check that the following path refers to the actual folder where the XMLSpy example files are stored on your operating system:

```
// Locate samples installed with the product.
final String strExamplesFolder = System.getenv( "USERPROFILE" ) + "\\Documents\\Altova\
\XMLSpy2023\\XMLSpyExamples\\";
```

Running the example from the command line

To run the example from the command line:

1. Check that all prerequisites are met (see Prerequisites 2052).
2. Open a command prompt window, change the current directory to the sample Java project folder, and type:

   ```
   buildAndRun.bat "<Path-to-the-Java-bin-folder>"
   ```

3. Press **Enter**.

The Java source in `XMLSpyContainer.java` will be compiled and then executed.

Compiling and running the example in Eclipse

To import the sample Java project into Eclipse:

1. Check that all prerequisites are met (see Prerequisites 2052).
2. On the **File** menu, click **Import**.
3. Select **Existing Projects into Workspace**, and browse for the Eclipse project file located at `<ApplicationFolder>\Examples\ActiveX\Java\`. Since you may not have write-access in this folder, it is recommended to select the **Copy projects into workspace** check box on the Import dialog box.

To run the example application, right-click the project in Package Explorer and select the command **Run as | Java Application**.

Help for Java API classes is available through comments in code as well as the Javadoc view of Eclipse. To enable the Javadoc view in Eclipse, select the menu command **Window | Show View | JavaDoc**.

29.4.5.2.2 Creating the ActiveX Controls

The code listing below show how ActiveX controls can be created. The constructors will create the Java wrapper objects. Adding these Canvas-derived objects to a panel or to a frame will trigger the creation of the wrapped ActiveX object.

© 2016-2022 Altova GmbH

```
01  /**
02  * XMLSpy manager control - always needed
03  */
04    public static XMLSpyControl         xmlSpyControl = null;
05
06  /**
07  * XMLSpy document editing control
08  */
09    public static XMLSpyControlDocument     xmlSpyDocument = null;
10
11  /**
12  * Tool windows - XMLSpy place-holder controls
13  */
14    private static XMLSpyControlPlaceHolder   xmlSpyProjectToolWindow = null;
15    private static XMLSpyControlPlaceHolder   xmlSpyXpathToolWindow = null;
16    private static XMLSpyControlPlaceHolder   xmlSpyEHAttributeToolWindow = null;
17
18  // Create the XMLSpy ActiveX control; the parameter determines that we want
19  // to place document controls and place-holder controls individually.
20  // It gives us full control over the menu, as well.
21     xmlSpyControl = new XMLSpyControl(
       ICActiveXIntegrationLevel.ICActiveXIntegrationOnDocumentLevel.getValue() );
22     xmlSpyDocument = new XMLSpyControlDocument();
23     xmlSpyDocument.setPreferredSize( new Dimension ( 640, 480 ) );
24
25  // Create a project window and open the sample project in it
26     xmlSpyProjectToolWindow = new XMLSpyControlPlaceHolder(
27     XMLSpyControlPlaceholderWindow.XMLSpyControlProjectWindowToolWnd.getValue() );
28     xmlSpyProjectToolWindow.setPreferredSize( new Dimension( 200, 200 ) );
29     xmlSpyXpathToolWindow = new XMLSpyControlPlaceHolder(
       XMLSpyControlPlaceholderWindow.XMLSpyControlXPathDialogToolWnd.getValue() );
30     xmlSpyEHAttributeToolWindow = new XMLSpyControlPlaceHolder(
       XMLSpyControlPlaceholderWindow.XMLSpyControlEntryHelperTopToolWnd.getValue() );
31
32     frame.add( xmlSpyControl, BorderLayout.NORTH );
33     frame.add( xmlSpyDocument, BorderLayout.CENTER );
34     southPanel.add( xmlSpyProjectToolWindow );
35     southPanel.add( xmlSpyXpathToolWindow );
36     southPanel.add( xmlSpyEHAttributeToolWindow );
```

29.4.5.2.3 Loading Data in the Controls

The code listing below show how data can be loaded in the ActiveX controls.

```
1    // Locate samples installed with the product.
2       final String strExamplesFolder = System.getenv( "USERPROFILE" ) +
        "\\Documents\\Altova\\XMLSpy2022\\Examples\\";
3       xmlSpyProjectToolWindow.openProject( strExamplesFolder  + "Examples.spp" );
```

29.4.5.2.4 Basic Event Handling

The code listing below shows how basic events can be handled. When calling the XMLSpyControl's `open` method, or when trying to open a file via the menu or Project tree, the `onOpenedOrFocused` event is sent to the

attached event handler. The basic handling for this event is opening the file by calling the XMLSpyDocumentControl's `open` method.

```
01        // Open the PXF file when button is pressed
02        btnOpenPxf.addActionListener( new ActionListener() {
03          public void actionPerformed(ActionEvent e) {
04            try {
05              xmlSpyControl.open( strExamplesFolder + "OrgChart.pxf" );
06            } catch (AutomationException e1) {
07              e1.printStackTrace();
08            }
09          }
10        } );
11        public void onOpenedOrFocused( String i_strFileName, boolean
i_bOpenWithThisControl, boolean i_bFileAlreadyOpened ) throws AutomationException
12    {
13      // Handle the New/Open events coming from the Project tree or from the menus
14      if ( !i_bFileAlreadyOpened )
15      {
16        // This is basically an SDI interface, so open the file in the already existing
document control
17        try {
18          XMLSpyContainer.xmlSpyDocument.open( i_strFileName );
19          XMLSpyContainer.xmlSpyDocument.requestFocusInWindow();
20        } catch (Exception e) {
21          e.printStackTrace();
22        }
23      }
24    }
```

29.4.5.2.5 Menus

The code listing below shows how menu items can be created. Each `XMLSpyCommand` object gets a corresponding `MenuItem` object, with the `ActionCommand` set to the ID of the command. The actions generated by all menu items are handled by the same function, which can perform specific handlings (like reinterpreting the closing mechanism) or can delegate the execution to the `XMLSpyControl` object by calling its `exec` method. The `menuMap` object that is filled during menu creation is used later (see section UI Update Event Handling 2070).

```
01    // Load the file menu when the button is pressed
02        btnMenu.addActionListener( new ActionListener() {
03          public void actionPerformed(ActionEvent e) {
04            try {
05              // Create the menubar that will be attached to the frame
06              MenuBar mb = new MenuBar();
07              // Load the main menu's first item - the File menu
08              XMLSpyCommand xmlSpyMenu =
xmlSpyControl.getMainMenu().getSubCommands().getItem( 0 );
09              // Create Java menu items from the Commands objects
10              Menu fileMenu = new Menu();
11              handlerObject.fillMenu( fileMenu, xmlSpyMenu.getSubCommands() );
12              fileMenu.setLabel( xmlSpyMenu.getLabel().replace( "&", "" ) );
13              mb.add( fileMenu );
14              frame.setMenuBar( mb );
```


© 2016-2022 Altova GmbH

```
15              frame.validate();
16            } catch (AutomationException e1) {
17              e1.printStackTrace();
18            }
19            // Disable the button when the action has been performed
20            ((AbstractButton) e.getSource()).setEnabled( false );
21          }
22        } ) ;
23 /** * Populates a menu with the commands and submenus contained in an XMLSpyCommands
object  */
24        public void fillMenu(Menu newMenu, XMLSpyCommands xmlSpyMenu) throws
AutomationException
25    {
26      // For each command/submenu in the xmlSpyMenu
27      for ( int i = 0 ; i < xmlSpyMenu.getCount() ; ++i  )
28      {
29        XMLSpyCommand xmlSpyCommand = xmlSpyMenu.getItem( i );
30        if ( xmlSpyCommand.getIsSeparator() )
31          newMenu.addSeparator();
32        else
33        {
34          XMLSpyCommands subCommands = xmlSpyCommand.getSubCommands();
35          // Is it a command (leaf), or a submenu?
36          if ( subCommands.isNull() || subCommands.getCount() == 0 )
37          {
38            // Command -> add it to the menu, set its ActionCommand to its ID and store it
in the menuMap
39            MenuItem mi = new MenuItem( xmlSpyCommand.getLabel().replace( "&", "" ) );
40            mi.setActionCommand( "" + xmlSpyCommand.getID() );
41            mi.addActionListener( this );
42            newMenu.add( mi );
43            menuMap.put( xmlSpyCommand.getID(), mi );
44          }
45          else
46          {
47            // Submenu -> create submenu and repeat recursively
48            Menu newSubMenu = new Menu();
49            fillMenu( newSubMenu, subCommands );
50            newSubMenu.setLabel( xmlSpyCommand.getLabel().replace( "&", "" ) );
51            newMenu.add( newSubMenu );
52          }
53        }
54      }
55    }
56
57     /**
58     * Action handler for the menu items
59     * Called when the user selects a menu item; the item's action command corresponds to
the command table for XMLSpy
60     */
61    public void actionPerformed( ActionEvent e )
62    {
63      try
64      {
65        int iCmd = Integer.parseInt( e.getActionCommand() );
66        // Handle explicitly the Close commands
67        switch ( iCmd )
68        {
69          case 57602:       // Close
```

```
70          case 34050:       // Close All
71            XMLSpyContainer.initXmlSpyDocument();
72            break;
73          default:
74            XMLSpyContainer.xmlSpyControl.exec( iCmd );
75            break;
76        }
77      }
78      catch ( Exception ex )
79      {
80        ex.printStackTrace();
81      }
82
83    }
```

29.4.5.2.6 UI Update Event Handling

The code listing below shows how a UI-Update event handler can be created.

```
01 /**
02    * Call-back from the XMLSpyControl.
03    * Called to enable/disable commands
04    */
05   @Override
06   public void onUpdateCmdUI() throws AutomationException
07   {
08     // A command should be enabled if the result of queryStatus contains the Supported
(1) and Enabled (2) flags
09     for ( java.util.Map.Entry<Integer, MenuItem> pair : menuMap.entrySet() )
10
      pair.getValue().setEnabled( XMLSpyContainer.xmlSpyControl.queryStatus( pair.getKey()
) > 2 );
11   }
12    /**
13    * Call-back from the XMLSpyControl.
14    * Usually called while enabling/disabling commands due to UI updates
15    */
16   @Override
17   public boolean onIsActiveEditor( String i_strFilePath ) throws AutomationException
18   {
19     try {
20       return
XMLSpyContainer.xmlSpyDocument.getDocument().getFullName().equalsIgnoreCase( i_strFilePath
);
21     } catch ( Exception e ) {
22       return false;
23     }
24   }
```

29.4.5.2.7 Creating an XML Tree

The listing below loads an XML data object as nodes in a tree.

© 2016-2022 Altova GmbH

```
01 // access required XMLSpy Java-COM classes
02 import com.altova.automation.XMLSpy.XMLData;
03
04 // access AWT and Swing components
05 import java.awt.*;
06 import javax.swing.*;
07 import javax.swing.tree.*;
08
09 /**
10  * A simple example of a tree control loading the structure from an XMLData object.
11  * The class receives an XMLData object, loads its nodes in a JTree, and prepares
12  * for modal activation.
13  *
14  * Feel free to modify and extend this sample.
15  *
16  * @author Altova GmbH
17  */
18 class XMLTreeDialog extends JDialog
19 {
20   /**
21    * The tree control
22    */
23   private JTree myTree;
24
25   /**
26    * Root node of the tree control
27    */
28   private DefaultMutableTreeNode top ;
29
30   /**
31    * Constructor that prepares the modal dialog containing the filled tree control
32    * @param xml   The data to be displayed in the tree
33    * @param parent  Parent frame
34    */
35   public XMLTreeDialog( XMLData xml, Frame parent )
36   {
37     // Construct the modal dialog
38     super( parent, "XML tree", true );
39     // Arrange controls in the dialog
40     top = new DefaultMutableTreeNode("root");
41     myTree = new JTree(top);
42     setContentPane( new JScrollPane( myTree ) );
43     // Build up the tree
44     fillTree( top, xml );
45     myTree.expandRow( 0 );
46   }
47
48   /**
49    * Loads the nodes of an XML element under a given tree node
50    * @param node  Target tree node
51    * @param elem  Source XML element
52    */
53   private void fillTree( DefaultMutableTreeNode node, XMLData elem)
54   {
55     try
56     {
57       // There are several ways to iterate through child elements: either using the
getFirstChild/getNextChild,
```

```
58        // or by incrementing an index up to countChildren and calling getChild [as shown
below].
59        // If you only want to get children of one kind, you should use
countChildrenKind/getChildKind,
60        // or provide a kind to the getFirstChild before iterating with the getNextChild.
61        int nSize = elem.countChildren() ;
62        for ( int i = 0 ; i < nSize ; ++i)
63        {
64          // Create a new tree node for each child element, and continue recursively
65          XMLData newElem = elem.getChild(i) ;
66          DefaultMutableTreeNode newNode = new DefaultMutableTreeNode( newElem.getName() )
;
67          node.add( newNode ) ;
68          fillTree( newNode, newElem ) ;
69        }
70      }
71      catch (Exception e)
72      {
73        e.printStackTrace();
74      }
75    }
76
77 }
```

29.4.6 Command Reference

This section lists the names and identifiers of all menu commands that are available within XMLSpy. Every sub-section lists the commands from the corresponding top-level menu of XMLSpy. The command tables are organized as follows:

- The "Menu Item" column shows the command's menu text as it appears in XMLSpy, to make it easier for you to identify the functionality behind the command.
- The "Command Name" column specifies the string that can be used to get an icon with the same name from **ActiveX\Images** folder of the XMLSpy installation directory.
- The "ID" column shows the numeric identifier of the column that must be supplied as argument to methods which execute or query this command.

To execute a command, use the `XMLSpyControl.Exec`[2099] or the `XMLSpyControlDocument.Exec`[2105] methods. To query the status of a command, use the `XMLSpyControl.QueryStatus`[2099] or `XMLSpyControlDocument.QueryStatus`[2106] methods.

Depending on the edition of XMLSpy you have installed, some of these commands might not be supported.

29.4.6.1 "File" Menu

The "File" menu has the following commands:

Menu item	Command name	ID
New...	ID_FILE_NEW	57600
Open...	ID_FILE_OPEN	57601

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Reload	IDC_FILE_RELOAD	34065
Encoding...	IDC_ENCODING	34061
Close	ID_FILE_CLOSE	57602
Close All	IDC_CLOSE_ALL	34050
Close All But Active	IDC_CLOSE_OTHERS	34271
Save	ID_FILE_SAVE	57603
Save As...	ID_FILE_SAVE_AS	57604
Save All	ID_FILE_SAVE_ALL	34208
Send by Mail...	ID_FILE_SEND_MAIL	57612
Print...	ID_FILE_PRINT	57607
Print Preview	IDC_PRINT_PREVIEW	34104
Print Setup...	ID_FILE_PRINT_SETUP	57606
Recent File	ID_FILE_MRU_FILE1	57616
Exit	ID_APP_EXIT	57665

29.4.6.2 "Edit" Menu

The "Edit" menu has the following commands:

Menu item	Command name	ID
Undo	ID_EDIT_UNDO	57643
Redo	ID_EDIT_REDO	57644
Cut	ID_EDIT_CUT	57635
Copy	ID_EDIT_COPY	57634
Paste	ID_EDIT_PASTE	57637
Delete	ID_EDIT_CLEAR	57632
Copy as XML Text	IDC_COPY_AS_XML_TEXT	33443
Copy as Structured Text	IDC_COPY_AS_STRUCTURED_TEXT	33442
Copy XPath	IDC_COPY_XPATH	33444

Menu item	Command name	ID
Copy XPointer	IDC_COPY_XPOINTER	33445
File Path...	IDC_EDIT_INSERT_PATH_STRING	34013
XInclude...	IDC_EDIT_INSERT_XINCLUDE_STRING	34017
Encoded External File...	IDC_EDIT_INSERT_ENCODED_BINARY_STRING	34273
Pretty-Print	IDC_PRETTY_PRINT	34101
Strip Whitespaces	IDC_STRIP_WHITESPACES	34296
Select All	ID_EDIT_SELECT_ALL	57642
Find...	ID_EDIT_FIND	57636
Find Next	ID_EDIT_REPEAT	57640
Replace...	ID_EDIT_REPLACE	57641
Find in Files...	IDC_FIND_IN_FILES	34000
Insert/Remove Bookmark	IDC_TOGGLE_BOOKMARK	34162
Remove All Bookmarks	IDC_REMOVEALLBOOKMARKS	34132
Go to Next Bookmark	IDC_GOTONEXTBOOKMARK	34070
Go to Previous Bookmark	IDC_GOTOPREVBOOKMARK	34071
Comment In/Out	IDC_TOGGLE_XML_COMMENT	34029

29.4.6.3 "Project" Menu

The "Project" menu has the following commands:

Menu item	Command name	ID
New Project	IDC_ICPROJECTGUI_NEW	37200
Open Project...	IDC_ICPROJECTGUI_OPEN	37201
Reload Project	IDC_ICPROJECTGUI_RELOAD	37202
Close Project	IDC_ICPROJECTGUI_CLOSE	37203
Save Project	IDC_ICPROJECTGUI_SAVE	37204
Save Project As...	IDC_ICPROJECTGUI_SAVE_AS	37207
Enable Source Control	ID_SCC_ENABLE	38602

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Add Files to Project...	IDC_ICPROJECTGUI_ADD_FILES_TO_PROJECT	37205
Add Global Resource to Project...	IDC_ICPROJECTGUI_ADD_GLOBAL_RESOURCE_TO_PROJECT	37239
Add URL to Project...	IDC_ICPROJECTGUI_ADD_URL_TO_PROJECT	37206
Add Active File to Project	IDC_ICPROJECTGUI_ADD_ACTIVE_FILE_TO_PROJECT	37208
Add Active and Related Files to Project	IDC_ICPROJECTGUI_ADD_ACTIVE_AND_RELATED_FILES_TO_PROJECT	37209
Add Project Folder to Project...	IDC_ICPROJECTGUI_ADD_FOLDER_TO_PROJECT	37210
Add External Folder to Project...	IDC_ICPROJECTGUI_ADD_EXT_FOLDER_TO_PROJECT	37211
Add External Web Folder to Project...	IDC_ICPROJECTGUI_ADD_EXT_URL_FOLDER_TO_PROJECT	37212
Script settings...	IDC_PROJECT_SCRIPT_SETTINGS	34136
Properties...	IDC_ICPROJECTGUI_PROJECT_PROPERTIES	37223
Recent Project	IDC_ICPROJECTGUI_RECENT	37224

29.4.6.4 "XML" Menu

The "XML" menu has the following commands:

Menu item	Command name	ID
Attribute	IDC_INSERT_ATTRIBUTE	33449
Element	IDC_INSERT_STRUCT	33459
Text	IDC_INSERT_TEXT	33460
CDATA	IDC_INSERT_CDATA	33450
Comment	IDC_INSERT_COMMENT	33451
XML	IDC_INSERT_XML	33461
Processing Instruction	IDC_INSERT_PI	33458

Menu item	Command name	ID
XInclude...	IDC_INSERT_XINCLUDE	34019
DOCTYPE	IDC_INSERT_DEF_DOCTYPE	33453
ExternalID	IDC_INSERT_DEF_EXTERNAL_ID	33456
ELEMENT	IDC_INSERT_DEF_ELEMENT	33454
ATTLIST	IDC_INSERT_DEF_ATTLIST	33452
ENTITY	IDC_INSERT_DEF_ENTITY	33455
NOTATION	IDC_INSERT_DEF_NOTATION	33457
Encoded External File...	IDC_INSERT_ENCODED_BINARY	34274
Attribute	IDC_APPEND_ATTRIBUTE	33415
Element	IDC_APPEND_STRUCT	33425
Text	IDC_APPEND_TEXT	33426
CDATA	IDC_APPEND_CDATA	33416
Comment	IDC_APPEND_COMMENT	33417
XML	IDC_APPEND_XML	33427
Processing Instruction	IDC_APPEND_PI	33424
XInclude...	IDC_APPEND_XINCLUDE	34026
DOCTYPE	IDC_APPEND_DEF_DOCTYPE	33419
ExternalID	IDC_APPEND_DEF_EXTERNAL_ID	33422
ELEMENT	IDC_APPEND_DEF_ELEMENT	33420
ATTLIST	IDC_APPEND_DEF_ATTLIST	33418
ENTITY	IDC_APPEND_DEF_ENTITY	33421
NOTATION	IDC_APPEND_DEF_NOTATION	33423
Encoded External File...	IDC_APPEND_ENCODED_BINARY	34276
Attribute	IDC_ADD_CHILD_ATTRIBUTE	33402
Element	IDC_ADD_CHILD_STRUCT	33412
Text	IDC_ADD_CHILD_TEXT	33413
CDATA	IDC_ADD_CHILD_CDATA	33403
Comment	IDC_ADD_CHILD_COMMENT	33404

© 2016-2022 Altova GmbH

Menu item	Command name	ID
XML	IDC_ADD_CHILD_XML	33414
Processing Instruction	IDC_ADD_CHILD_PI	33411
XInclude...	IDC_ADD_CHILD_XINCLUDE	34027
DOCTYPE	IDC_ADD_CHILD_DEF_DOCTYPE	33406
ExternalID	IDC_ADD_CHILD_DEF_EXTERNAL_ID	33409
ELEMENT	IDC_ADD_CHILD_DEF_ELEMENT	33407
ATTLIST	IDC_ADD_CHILD_DEF_ATTLIST	33405
ENTITY	IDC_ADD_CHILD_DEF_ENTITY	33408
NOTATION	IDC_ADD_CHILD_DEF_NOTATION	33410
Encoded External File...	IDC_ADD_CHILD_ENCODED_BINARY	34277
Attribute	IDC_CONVERT_TO_ATTRIBUTE	33429
Element	IDC_CONVERT_TO_STRUCT	33439
Text	IDC_CONVERT_TO_TEXT	33440
CDATA	IDC_CONVERT_TO_CDATA	33430
Comment	IDC_CONVERT_TO_COMMENT	33431
XML	IDC_CONVERT_TO_XML	33441
Processing Instruction	IDC_CONVERT_TO_PI	33438
DOCTYPE	IDC_CONVERT_TO_DEF_DOCTYPE	33433
ExternalID	IDC_CONVERT_TO_DEF_EXTERNAL_ID	33436
ELEMENT	IDC_CONVERT_TO_DEF_ELEMENT	33434
ATTLIST	IDC_CONVERT_TO_DEF_ATTLIST	33432
ENTITY	IDC_CONVERT_TO_DEF_ENTITY	33435
NOTATION	IDC_CONVERT_TO_DEF_NOTATION	33437
Display as Table	IDC_GRID_VIEW_AS_TABLE	34075
Insert Row	IDC_TABLE_INSERT_ROW	34158
Append Row	IDC_TABLE_APPEND_ROW	34157
Ascending Sort	IDC_TABLE_SORT_ASC	33464
Descending Sort	IDC_TABLE_SORT_DESC	33465

Menu item	Command name	ID
Move Left	IDC_MOVE_LEFT	34091
Move Right	IDC_MOVE_RIGHT	34092
Enclose in Element	IDC_ENCLOSE_IN_ELEMENT	33446
Evaluate XPath...	IDC_EVALUATE_XPATH	34007
Check Well-Formedness	IDC_CHECK_WELL_FORM	34049
Validate XML	IDC_VALIDATE	32954
Validate XML on Server (high-performance)	IDC_VALIDATE_RAPTOR	34309
Update Entry Helpers	IDC_UPDATE_ELEMENT_CHOICE	34173
Namespace Prefix...	IDC_NAMESPACE	33462
Create XML Signature...	IDC_XML_SIGNATURE_CREATE	34280
Verify XML Signature...	IDC_XML_SIGNATURE_VERIFY	34281

29.4.6.5 "DTD/Schema" Menu

The "DTD/Schema" menu has the following commands:

Menu item	Command name	ID
Assign DTD...	IDC_ASSIGN_DTD	34032
Assign Schema...	IDC_ASSIGN_SCHEMA	34033
Include Another DTD...	IDC_INCLUDE_DTD	34084
Go to DTD	IDC_GOTO_DTD	34072
Go to Schema	IDC_GOTO_SCHEMA	34074
Go to Definition	IDC_GOTO_DEFINITION	33447
Generate DTD/Schema...	IDC_GENERATE_DTD_SCHEMA	34068
Flatten DTD...	IDC_FLATTEN_DTD	34301
Convert DTD To Schema...	IDC_CONVERT_DTD_TO_SCHEMA	34299
Flatten Schema...	IDC_FLATTEN_SCHEMA	34302
Convert Schema To DTD...	IDC_CONVERT_SCHEMA_TO_DTD	34300
Convert to UML...	IDC_CONVERT_SCHEMA_TO_UML	34008

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Generate XML from DB, Excel, EDI with MapForce...	IDC_DTD_OPENIN_MAPFORCE	34056
Design HTML/PDF/Word Output with StyleVision...	IDC_DTD_OPENIN_STYLEVISION	34057
Generate Sample XML/JSON File...	IDC_GENERATE_XML_FROM_SCHEMA	34069
Generate Program Code...	IDC_GENERATE_CODE_FROM_SCHEMA	34067
Flush Memory Cache	IDC_FLUSH_CACHED_FILES	34066

29.4.6.6 "Schema design" Menu

The "Schema design" menu has the following commands:

Menu item	Command name	ID
Schema Settings...	IDC_SCHEMA_NAMESPACES	33571
Save Diagram...	IDC_SCHEMA_SAVE_DIAGRAM	33581
Generate Documentation...	IDC_SCHEMA_DOCUMENTATION	34146
Configure View...	IDC_SCHEMA_VIEW_CONFIG	33593
Zoom...	IDC_SCHEMA_ZOOM	34150
Display All Globals	IDC_SCHEMA_MODE_GLOBALS	34147
Display Diagram	IDC_SCHEMA_MODE_DIAGRAM	33570
Enable Oracle Schema Extensions	IDC_SCHEMA_ORACLE_EXTENSIONS	33577
Oracle Schema Settings...	IDC_SCHEMA_ORACLE_SCHEMA_SETTINGS	33578
Enable Microsoft SQL Server Schema Extensions	IDC_SCHEMA_SQLSERVER_EXTENSIONS	33588
Named Schema Relationships...	IDC_SCHEMA_SQLSERVER_GLOBAL_RELATIONSHIPS	33589
Unnamed Element Relationships...	IDC_SCHEMA_SQLSERVER_LOCAL_RELATIONSHIPS	33590
Connect to SchemaAgent Server...	IDC_SCHEMA_SCHEMAAGENT_SERVER_CONNECT	33582
Disconnect from SchemaAgent Server	IDC_SCHEMA_SCHEMAAGENT_SERVER_DISCONNECT	33583

Menu item	Command name	ID
File Only	IDC_SCHEMAAGENT_SHOW_FILE_ONLY	33504
File and All Directly Referenced Schema Files	IDC_SCHEMAAGENT_SHOW_WITH_DIRECTLY_REFERENCED_SCHEMAS	33608
File and All Directly Referencing Schema Files	IDC_SCHEMAAGENT_SHOW_WITH_DIRECTLY_REFERENCING_SCHEMAS	33602
File and All Directly Related Schema Files	IDC_SCHEMAAGENT_SHOW_WITH_DIRECTLY_RELATED_SCHEMAS	33613
SchemaAgent Validation...	IDC_SCHEMA_EXTVALID_MENU	33539
Create Schema Subset...	IDC_SCHEMA_CREATE_SUBSET	33650
Flatten Schema...	IDC_SCHEMA_FLATTEN	33651

29.4.6.7 "XSL/XQuery" Menu

The "XSL/XQuery" menu has the following commands:

Menu item	Command name	ID
XSL Transformation	IDC_TRANSFORM_XSL	33006
XSL Speed Optimizer	IDC_TRANSFORM_XSLPBO	34306
XSL-FO Transformation	IDC_TRANSFORM_XSLFO	33007
XSL Parameters / XQuery Variables...	IDC_TRANSFORM_XSL_PARAMS	33008
XQuery/Update Execution	IDC_TRANSFORM_XQUERY	34170
Enable Back Mapping	IDC_ENABLE_BACKMAPPING	34364
Enable XSLT/ XQuery Profiling....	IDC_PROFILING_OPTIONS	34105
Assign XSL...	IDC_ASSIGN_XSL	33001
Assign XSL-FO...	IDC_ASSIGN_XSLFO	33002
Assign Sample XML File...	IDC_ASSIGN_SAMPLE_XML	33000
Go to XSL	IDC_GOTO_XSL	33004
Start Debugger / Go	ID_PROCESS_XSL	34212
Stop Debugger	ID_XSLT_DEBUGGER_STOP	33017
Restart Debugger	ID_XSLT_DEBUGGER_RESTART	33013
End Debugger Session	ID_XSLT_DEBUGGER_END_SESSION	33011

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Step Into	ID_XSLT_DEBUGGER_STEP	33014
Step Out	ID_XSLT_DEBUGGER_STEP_OUT	33015
Step Over	ID_XSLT_DEBUGGER_STEP_OVER	33016
Show Current Execution Node	ID_XSLT_DEBUGGER_GO_TO_CURRENT_EXECUTION_NODES	33012
Insert/Remove Breakpoint	IDC_TOGGLE_BREAKPOINT	34246
Insert/Remove Tracepoint	IDC_TOGGLE_TRACEPOINT	34248
Enable/Disable Breakpoint	IDC_ENABLE_BREAKPOINT	34245
Enable/Disable Tracepoint	IDC_ENABLE_TRACEPOINT	34247
Breakpoints/Tracepoints...	ID_XSLTDEBUGGER_BREAKPOINTS	33009
Call Stack	ID_XSL_DEBUGWINDOWS_CALLSTACK	34238
XPath-Watch	ID_XSL_DEBUGWINDOWS_WATCH	34244
Context	ID_XSL_DEBUGWINDOWS_CONTEXT	34239
Variables	ID_XSL_DEBUGWINDOWS_VARIABLE	34243
Messages	ID_XSL_DEBUGWINDOWS_MESSAGES	34240
Templates	ID_XSL_DEBUGWINDOWS_TEMPLATES	34241
Info	ID_XSLXQUERY_DEBUGWINDOWS_INFO	34237
Trace	ID_XSL_DEBUGWINDOWS_TRACES	34242
Debug Settings...	ID_XSLTDEBUGGER_SETTINGS	33010

29.4.6.8 "Authentic" Menu

The "Authentic" menu has the following commands:

Menu item	Command name	ID
New Document...	IDC_AUTHENTIC_NEW_FILE	34036
Edit Database Data...	IDC_AUTHENTIC_EDIT_DB	34035
Assign a StyleVision Stylesheet...	IDC_ASSIGN_SPS	34034
Edit StyleVision Stylesheet	IDC_EDIT_SPS	34060
Select New Row with XML Data for Editing...	IDC_CHANGE_WORKING_DB_XML_CELL	32861

Menu item	Command name	ID
XML Signature...	IDC_AUTHENTICGUI_XMLSIGNATURE	32862
Define XML Entities...	IDC_DEFINE_ENTITIES	32805
Hide Markup	IDC_MARKUP_HIDE	32855
Show Small Markup	IDC_MARKUP_SMALL	32858
Show Large Markup	IDC_MARKUP_LARGE	32856
Show Mixed Markup	IDC_MARKUP_MIXED	32857
Toggle Bold	IDC_AUTHENTICGUI_RICHEDIT_TOGGLEBOLD	32813
Toggle Italic	IDC_AUTHENTICGUI_RICHEDIT_TOGGLEITALIC	32814
Toggle Underline	IDC_AUTHENTICGUI_RICHEDIT_TOGGLEUNDERLINE	32815
Toggle Strikethrough	IDC_AUTHENTICGUI_RICHEDIT_TOGGLESTRIKETHROUGH	32816
Foreground Color	IDC_AUTHENTICGUI_RICHEDIT_COLOR_FOREGROUND	32824
Background Color	IDC_AUTHENTICGUI_RICHEDIT_COLOR_BACKGROUND	32830
Align Left	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_LEFT	32818
Center	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_CENTER	32819
Align Right	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_RIGHT	32820
Append Row	IDC_ROW_APPEND	32806
Insert Row	IDC_ROW_INSERT	32809
Duplicate Row	IDC_ROW_DUPLICATE	32808
Move Row Up	IDC_ROW_MOVE_UP	32811
Move Row Down	IDC_ROW_MOVE_DOWN	32810
Delete Row	IDC_ROW_DELETE	32807
Generate an HTML document	IDC_PXF_GENERATE_HTML	34283
Generate an RTF document	IDC_PXF_GENERATE_RTF	34284
Generate a PDF document	IDC_PXF_GENERATE_PDF	34285

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Generate a Word 2007+ document	IDC_PXF_GENERATE_DOCX	34286
Generate a Text document	DC_PXF_GENERATE_TEXT	
Trusted Locations...	IDC_TRUSTED_LOCATIONS	34288

29.4.6.9 "DB" Menu

The "DB" menu has the following commands:

Menu item	Command name	ID
Query Database	IDC_QUERYDATABASE	34012
Manage XML Schemas...	IDC_DB_MANAGESCHEMAS	34014
Assign XML Schema...	IDC_DB_CHOOSEVALIDATIONSCHEMA	34016
Manage XML Schemas...	IDC_DB_MANAGESCHEMAS	34014
Manage XML Schemas...	IDC_DB_MANAGESCHEMAS	34014
Browse Oracle XML Documents...	ID_CONVERT_ORACLEXMLDB_BROWSE	34205

29.4.6.10 "Convert" Menu

The "Convert" menu has the following commands:

Menu item	Command name	ID
Import Text File...	IDC_IMPORT_TEXT	34082
Import Database Data...	IDC_IMPORT_DATABASE	34080
Import Microsoft Word Document...	IDC_IMPORT_WORD	34083
Create XML Schema from DB Structure	IDC_CREATE_DB_SCHEMA	34054
DB Import Based on XML Schema	IDC_IMPORT_DB_SCHEMA	34081
Create DB Structure from XML Schema	IDC_CREATE_DB_BASED_ON_SCHEMA	34053
Export to Text Files...	IDC_EXPORT_TEXTFILE	34064
Export to a Database...	IDC_EXPORT_DB	34003
Convert XML Instance to/from JSON...	IDC_JSON_CONVERT_TOFROM_XML	34135
Convert XML Schema to/from JSON Schema...	IDC_JSON_CONVERT_TOFROM_XSD	34350

29.4.6.11 "View" Menu

The "View" menu has the following commands:

Menu item	Command name	ID
Text View	IDC_VIEW_TEXT	34180
Enhanced Grid View	IDC_VIEW_GRID	34178
Schema Design View	IDC_VIEW_SCHEMA	34179
WSDL Design View	IDC_VIEW_WSDL	34117
XBRL Taxonomy View	IDC_VIEW_XBRL	34118
Authentic View	IDC_VIEW_CONTENT	34177
Browser View	IDC_VIEW_BROWSER	34176
Expand +	IDC_SEL_EXPAND	34152
Collapse -	IDC_SEL_COLLAPSE	34151
Expand Fully	IDC_SEL_EXPAND_ALL	33463
Collapse Unselected	IDC_COLLAPSE_UNSELECTED	33428
Optimal Widths	IDC_OPTIMAL_WIDTHS	34099
Word Wrap	IDC_WORD_WRAP	34181
Go to Line/Character	IDC_GOTO_LINE	34073
Go to File	IDC_GOTO_FILE	33448
Text View Settings	IDC_TEXTVIEW_SETTINGS	34119

29.4.6.12 "Browser" Menu

The "Browser" menu has the following commands:

Menu item	Command name	ID
Back	IDC_STEP_BACK	32958
Forward	IDC_STEP_FORWARD	32957
Stop	IDC_BROWSER_STOP	34047
Refresh	IDC_BROWSER_REFRESH	34046

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Largest	IDC_BROWSER_FONT_LARGEST	34041
Larger	IDC_BROWSER_FONT_LARGE	34040
Medium	IDC_BROWSER_FONT_MEDIUM	34042
Smaller	IDC_BROWSER_FONT_SMALL	34043
Smallest	IDC_BROWSER_FONT_SMALLEST	34044

29.4.6.13 "WSDL" Menu

The "WSDL" menu has the following commands:

Menu item	Command name	ID
Insert Message	ID_WSDL_MESSAGES_ADDNEWMESSAGE	33715
Delete Message	ID_WSDL_MESSAGES_DELETESELECTEDMESSAGE	33717
Add Message Part (Parameter)	ID_WSDL_MESSAGES_ADDMESSAGEPART	33714
Delete Message Part (Parameter)	ID_WSDL_MESSAGES_DELETEMESSAGEPART	33716
request-response	IDC_WSDL_OPERATION_APPENDREQUESTRESPONSE	33734
solicit-response	IDC_WSDL_OPERATION_APPENDSOLICITRESPONSE	33737
one-way	IDC_WSDL_OPERATION_APPENDONEWAY	33735
notification	IDC_WSDL_OPERATION_APPENDNOTIFICATION	33736
Empty Operation	ID_WSDL_OPERATIONS_APPENDAOPERATIONTOTHISPORTTYPE	33722
Delete Operation	ID_WSDL_OPERATIONS_DELETEOPERATION	33724
Add Input Element	ID_WSDL_OPERATIONS_ADDINPUTFUNCTION	33719
Add Output Element	ID_WSDL_OPERATIONS_ADDOUTPUTFUNCTION	33721
Add Fault Element	ID_WSDL_OPERATIONS_ADDFAULTFUNCTION	33718

Menu item	Command name	ID
Delete Input/Output/Fault Element	ID_WSDL_OPERATIONS_DELETEINPUTOUTPUTFUNCTION	33723
Add New Message to Input/Output/Fault Element	ID_WSDL_OPERATIONS_ADDNEWMESSAGETOTHISELEMENT	33720
Insert Port Type	ID_WSDL_PORTTYPE_INSERTAPORTTYPE	33727
Delete Port Type	ID_WSDL_PORTTYPE_DELETETHISPORTTYPE	33726
Insert Binding	ID_WSDL_BINDING_NEWBINDING	33713
Delete Binding	ID_WSDL_BINDING_DELETEBINDING	33711
soap:body	ID_WSDL_BINDING_APPENDEXTENSIBILITY_SOAPBODY	33706
soap:header	ID_WSDL_BINDING_APPENDEXTENSIBILITY_SOAPHEADER	33708
soap:headerfault	ID_WSDL_BINDING_APPENDEXTENSIBILITY_SOAPHEADERFAULT	33709
soap:fault	ID_WSDL_BINDING_APPENDEXTENSIBILITY_SOAPFAULT	33707
mime:content	ID_WSDL_BINDING_APPENDEXTENSIBILITY_MIMECONTENT	33702
mime:multipartrelated	ID_WSDL_BINDING_APPENDEXTENSIBILITY_MIMEMULTIPARTRELATED	33704
mime:part	ID_WSDL_BINDING_APPENDEXTENSIBILITY_MIMEPART	33705
mime:mimeXml	ID_WSDL_BINDING_APPENDEXTENSIBILITY_MIMEMIMEXML	33703
http:urlencoded	ID_WSDL_BINDING_APPENDEXTENSIBILITY_HTTPURLENCODED	33700
http:urlreplacement	ID_WSDL_BINDING_APPENDEXTENSIBILITY_HTTPURLREPLACEMENT	33701
Delete Extensibility Element	ID_WSDL_BINDING_DELETEEXTENSIBILITY	33712
Insert Service	ID_WSDL_SERVICE_INSERTSERVICE	33731
Delete Service	ID_WSDL_SERVICE_DELETETHISSERVICE	33729
Insert Port	ID_WSDL_SERVICE_INSERTNEWPORT	33730
Delete Port	ID_WSDL_SERVICE_DELETETHISPORT	33728

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Add New Interface	IDC_WSDL20_ADDINTERFACE	33794
Delete Interface	IDC_WSDL20_DELETEINTERFACE	33795
Add New Fault	IDC_WSDL20_ADDINTERFACEFAULT	33796
Delete Fault	IDC_WSDL20_DELETEINTERFACEFAULT	33808
In-only	IDC_WSDL20_ADDINTERFACEOPERATION_INONLY	33797
Robust-in-only	IDC_WSDL20_ADDINTERFACEOPERATION_ROBUSTINONLY	33798
In-out	IDC_WSDL20_ADDINTERFACEOPERATION_INOUT	33801
In-opt-out	IDC_WSDL20_ADDINTERFACEOPERATION_INOPTOUT	33802
Out-in	IDC_WSDL20_ADDINTERFACEOPERATION_OUTIN	33803
Out-opt-in	IDC_WSDL20_ADDINTERFACEOPERATION_OUTOPTIN	33804
Out-only	IDC_WSDL20_ADDINTERFACEOPERATION_OUTONLY	33800
Robust-out-only	IDC_WSDL20_ADDINTERFACEOPERATION_ROBUSTOUTONLY	33799
Empty Operation	IDC_WSDL20_ADDINTERFACEOPERATION_EMPTY	33805
Delete Operation	IDC_WSDL20_DELETEINTERFACEOPERATION	33809
Add New Binding	IDC_WSDL20_ADDBINDING	33820
Delete Binding	IDC_WSDL20_DELETEBINDING	33821
Add New Fault	IDC_WSDL20_ADDBINDINGFAULT	33822
Delete Fault	IDC_WSDL20_DELETEBINDINGFAULT	33826
Add New Operation	IDC_WSDL20_ADDBINDINGOPERATION	33823
Delete Operation	IDC_WSDL20_DELETEBINDINGOPERATION	33827
Add New Service	IDC_WSDL20_ADDSERVICE	33839
Delete Service	IDC_WSDL20_DELETESERVICE	33840

Menu item	Command name	ID
Add New Endpoint	IDC_WSDL20_ADDENDPOINT	33841
Delete Endpoint	IDC_WSDL20_DELETEENDPOINT	33842
New Schema	ID_WSDL_TYPES_NEWSCHEMA	33733
Embed Schema	ID_WSDL_TYPES_EMBEDSCHEMA	39456
Extract Schema(s)	ID_WSDL_TYPES_EXTRACTSCHEMAS	39459
Edit Schema(s) in Schema View	ID_WSDL_TYPES_EDITTHISSCHEMA	33732
Save Diagram...	IDC_WSDL_SAVE_DIAGRAM	39451
Generate Documentation...	ID_WSDL_GENERATEDOCUMENTATION	39452
Reparse WSDL Document	IDC_WSDL_REPARSE	33774
Convert to WSDL 2.0	IDC_WSDL_CONVERT_TO_WSDL20	39453
Generate WSDL Program Code with MapForce...	IDC_WSDL_GENERATE_CODE_MAPFORCE	34122

29.4.6.14 "SOAP" Menu

The "SOAP" menu has the following commands:

Menu item	Command name	ID
Create New SOAP Request...	ID_SOAP_GENERATESOAPMESSAGE	34224
Send Request to Server...	ID_SOAP_SENDREQUESTTOSERVER	34225
SOAP Request Settings...	ID_SOAP_SOAPREQUESTSETTINGS	34227
Soap Debugger Session	ID_SOAP_SOAPDEBUGGER	34226
Go	ID_SOAPDEBUGGER_BUTTONPLAY	34221
Single Step	ID_SOAPDEBUGGER_SINGLESTEP	34222
Break on Next Request	ID_SOAPDEBUGGER_BREAKONNEXTREQUEST	34219
Break on Next Response	ID_SOAPDEBUGGER_BREAKONNEXTRESPONSE	34220
Stop the Proxy Server	ID_SOAPDEBUGGER_STOPSERVER	34223
Soap Debugger Options	ID_SOAPDEBUGGEROPTIONS	34218

© 2016-2022 Altova GmbH

29.4.6.15 "XBRL" Menu

The "XBRL" menu has the following commands:

Menu item	Command name	ID
Arcroles...	IDC_XMLSPYXBRLEDITOR_ARCROLES	34114
Linkroles...	IDC_XMLSPYXBRLEDITOR_LINKROLES	34115
Namespace Prefixes...	IDC_XMLSPYXBRLEDITOR_NAMESPACES	34116
Set Target Namespace...	IDC_XMLSPYXBRLEDITOR_SET_TARGETNAMESPACE	34039
Parameter Values...	IDC_ICXBRLEDITOR_PARAMETER_VALUES	38913
Import/Reference...	IDC_XMLSPYXBRLEDITOR_IMPORT_REFERENCE	34137
Find Component By Id...	IDC_ICXBRLEDITOR_FIND_COMPONENT_BY_ID	38893
Generate Documentation...	IDC_XMLSPYXBRLEDITOR_GENERATEDOCUMENTATION	34125
View Settings...	IDC_XMLSPYXBRLEDITOR_VIEWSETTINGS	34113
Generate XBRL from DB, Excel, CSV with MapForce...	IDC_XBRL_GENERATE_WITH_MAPFORCE	34045
Present XBRL as HTML/PDF/Word with StyleVision...	IDC_XBRL_PRESENT_WITH_STYLEVISION	34121
Execute Formula...	IDC_XBRL_EXECUTE_FORMULA	34305
Execute Formula on Server (high-performance)...	IDC_XBRL_EXECUTE_FORMULA_RAPTOR	34352
Generate Table...	IDC_XBRL_GENERATE_TABLE	34304
Generate Table on Server (high-performance)...	IDC_XBRL_GENERATE_TABLE_RAPTOR	34353
Transform Inline XBRL	IDC_IXBRL_TRANSFORM	34354

29.4.6.16 "Tools" Menu

The "Tools" menu has the following commands:

Menu item	Command name	ID
Spelling...	IDC_SPELL_CHECK	34154

Menu item	Command name	ID
Spelling Options...	IDC_SPELL_OPTIONS	34155
Scripting Editor...	ID_SCRIPTFORMEDITOR_EDIT_PROJECT	39666
none	ID_SCRIPTFORMEDITOR_EXECUTE_MACRO_MENU_UPPDATE	39600
Compare Open File With...	ID_XMLDIFF_CHOOSE_FILES	34235
Compare Directories...	ID_XMLDIFF_CHOOSE_DIRECTORIES	34234
Compare Options...	ID_XMLDIFF_SETTINGS	34236
	IDC_TOOLS_ENTRY	34292
Global Resources	IDC_GLOBALRESOURCES	37401
	IDC_GLOBALRESOURCES_SUBMENUENTRY1	37408
Manage Raptor Servers ...	IDC_VALIDATE_RAPTOR_MANAGER	34311
none	IDC_VALIDATE_RAPTOR_NOCFG	34326
Customize...	IDC_APP_TOOLS_CUSTOMIZE	32959
Options...	IDC_SETTINGS	34133
	ID_SCRIPTING_MACROITEMS	34249

29.4.6.17 "Window" Menu

The "Window" menu has the following commands:

Menu item	Command name	ID
Cascade	ID_WINDOW_CASCADE	57650
Tile horizontally	ID_WINDOW_TILE_HORZ	57651
Tile vertically	ID_WINDOW_TILE_VERT	57652
Project window	IDC_PROJECT_WINDOW	34128
Info window	IDC_INFO_WINDOW	34085
Entry Helpers	IDC_ENTRY_HELPERS	34062
Output windows	IDC_OUTPUT_DIALOGBARS	34004
Project and Entry Helpers	IDC_PROJECT_ENTRYHELPERS	34006

© 2016-2022 Altova GmbH

Menu item	Command name	ID
All on/off	IDC_ALL_BARS	34031

29.4.6.18 "Help" Menu

The "Help" menu has the following commands:

Menu item	Command name	ID
Table of Contents...	IDC_HELP_CONTENTS	32966
Index...	IDC_HELP_INDEX	32967
Search...	IDC_HELP_SEARCH	32969
Keyboard Map...	IDC_HELP_KEYMAPDLG	32968
Software Activation...	IDC_ACTIVATION	32970
Order Form...	IDC_OPEN_ORDER_PAGE	32971
Registration...	IDC_REGISTRATION	32972
Check for Updates...	IDC_CHECK_FOR_UPDATES	32973
XMLSpy Product Comparison...	IDC_PRODUCT_COMPARISON	32955
Support Center...	IDC_OPEN_SUPPORT_PAGE	32961
FAQ on the Web...	IDC_OPEN_FAQ_PAGE	32962
Download Components and Free Tools...	IDC_OPEN_COMPONENTS_PAGE	32963
Authentic on the Internet..	IDC_OPEN_HOME_PAGE	32964
Authentic Training...	IDC_OPEN_TRAINING_PAGE	32965
About XMLSpy...	ID_APP_ABOUT	57664

29.4.7 Object Reference

Objects:
XMLSpyCommand 2092
XMLSpyCommands 2094
XMLSpyControl 2095
XMLSpyControlDocument 2102
XMLSpyControlPlaceHolder 2109

To give access to standard XMLSpy functionality, objects of the **XMLSpy automation interface** can be accessed as well. See XMLSpyControl.Application 2096, XMLSpyControlDocument.Document 2104 and XMLSpyControlPlaceHolder.Project 2110 for more information.

29.4.7.1 XMLSpyCommand

Properties:
ID 2093
Label 2093
Name 2093
IsSeparator 2093
ToolTip 2094
StatusText 2093
Accelerator 2092
SubCommands 2094

Description:
A command object can be one of the following: an executable command, a command container (for example, a menu, submenu, or toolbar), or a menu separator. To determine what kind of information is stored in the current `Command` object, query its `ID`, `IsSeparator`, and `SubCommands` properties, as follows.

The Command object is...	When...
An executable command	• `ID` is greater than zero • `IsSeparator` is false • `SubCommands` is empty
A command container	• `ID` is zero • `IsSeparator` is false • `SubCommands` contains a collection of `Command` objects.
Separator	• `ID` is zero • `IsSeparator` is true

29.4.7.1.1 Accelerator

Property: `Accelerator` as string

Description:
Returns the accelerator key defined for the command. If the command has no accelerator key assigned, this property returns the empty string. The string representation of the accelerator key has the following format:

```
[ALT+][CTRL+][SHIFT+]key
```

Where `key` is converted using the Windows Platform SDK function `GetKeyNameText`.

© 2016-2022 Altova GmbH

29.4.7.1.2 ID

Property: `ID` as long

Description:
This property gets the unique identifier of the command. A command's ID is required to execute the command (using `Exec`[2099]) or query its status (using `QueryStatus`[2099]). If the command is a container for other commands (for example, a top-level menu), or a separator, the ID is 0.

29.4.7.1.3 IsSeparator

Property: `IsSeparator` as boolean

Description:
The property returns `true` if the command object is a menu separator; `false` otherwise. See also Command[2092].

29.4.7.1.4 Label

Property: `Label` as string

Description:
This property gets the text of the command as it is displayed in the graphical user interface of XMLSpy. If the command is a separator, "Label" is an empty string. This property may also return an empty string for some toolbar commands that do not have any GUI text associated with them.

29.4.7.1.5 Name

Property: `Name` as string

Description:
This property gets the unique name of the command. This value can be used to get the icon file of the command, where it is available. The available icon files can be found in the folder **<ApplicationFolder>\Examples\ActiveX\Images** of your XMLSpy installation.

29.4.7.1.6 StatusText

Property: `Label` as string

Description:
The status text is the text shown in the status bar of XMLSpy when the command is selected. It applies only to command objects that are not separators or containers of other commands; otherwise, the property is an empty string.

29.4.7.1.7 SubCommands

Property: `SubCommands` as `Commands` 2094

Description:
The `SubCommands` property gets the collection of `Command` 2092 objects that are sub-commands of the current command. The property is applicable only to commands that are containers for other commands (menus, submenus, or toolbars). Such container commands have the `ID` set to 0, and the `IsSeparator` property set to `false`.

29.4.7.1.8 ToolTip

Property: `ToolTip` as string

Description:
This property gets the text that is shown as a tool-tip for each command. If the command does not have a tooltip text, the property returns an empty string.

29.4.7.2 XMLSpyCommands

Properties:
`Count` 2094
`Item` 2094

Description:
Collection of `Command` 2092 objects to get access to command labels and IDs of the XMLSpyControl. Those commands can be executed with the `Exec` 2099 method and their status can be queried with `QueryStatus` 2099.

29.4.7.2.1 Count

Property: `Count` as long

Description:
Number of `Command` 2092 objects on this level of the collection.

29.4.7.2.2 Item

Property: `Item` (`n` as long) as `Command` 2092

Description:
Gets the command with the index `n` in this collection. Index is 1-based.

© 2016-2022 Altova GmbH

29.4.7.3 XMLSpyControl

Properties:
IntegrationLevel 2097
Appearance 2096
Application 2096
BorderStyle 2096
CommandsList 2096
EnableUserPrompts 2097
MainMenu 2097
Toolbars 2098

Methods:
Open 2099
Exec 2099
QueryStatus 2099

Events:
OnUpdateCmdUI 2102
OnOpenedOrFocused 2101
OnCloseEditingWindow 2100
OnFileChangedAlert 2101
OnContextChanged 2100
OnDocumentOpened 2101
OnValidationWindowUpdated 2102

This object is a complete ActiveX control and should only be visible if the XMLSpy library is used in the Application Level mode.

```
CLSID: a258bba2-3835-4c16-8590-72b44f52c471
ProgID: Altova.XMLSpyControl
```

29.4.7.3.1 Properties

The following properties are defined:

IntegrationLevel 2097
EnableUserPrompts 2097
Appearance 2096
BorderStyle 2096

Command related properties:
CommandsList 2096
MainMenu 2097
Toolbars 2098

Access to XMLSpyAPI:
Application 2096

29.4.7.3.1.1 Appearance

Property: `Appearance` as short

Dispatch Id: *-520*

Description:
A value not equal to `0` displays a client edge around the control. Default value is `0`.

29.4.7.3.1.2 Application

Property: `Application` as Application

Dispatch Id: *1*

Description:
The `Application` property gives access to the `Application` object of the complete XMLSpy automation server API. The property is read-only.

29.4.7.3.1.3 BorderStyle

Property: `BorderStyle` as short

Dispatch Id: *-504*

Description:
A value of `1` displays the control with a thin border. Default value is `0`.

29.4.7.3.1.4 CommandsList

Property: `CommandList` as `Commands` 2094 (read-only)

Dispatch Id: *1004*

Description:
This property returns a flat list of all commands defined available with XMLSpyControl. To get commands organized according to their menu structure, use `MainMenu` 2097. To get toolbar commands, use `Toolbars` 2098.

```
public void GetAllXmlSpyCommands()
{
    // Get all commands from the XMLSpy ActiveX control assigned to the current form
    XMLSpyControlLib.XMLSpyCommands commands = this.axXMLSpyControl1.CommandList;
    // Iterate through all commands
    for (int i = 0; i < commands.Count; i++)
    {
```


© 2016-2022 Altova GmbH

```
      // Get each command by index and output it to the console
      XMLSpyControlLib.XMLSpyCommand cmd = axXMLSpyControl1.CommandList[i];
      Console.WriteLine("{0} {1} {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&", ""));
    }
}
```

C# example

29.4.7.3.1.5 EnableUserPrompts

Property: `EnableUserPrompts` as boolean

Dispatch Id: *1006*

Description:
Setting this property to *false*, disables user prompts in the control. The default value is *true*.

29.4.7.3.1.6 IntegrationLevel

Property: `IntegrationLevel` as `ICActiveXIntegrationLevel` 2111

Dispatch Id: *1000*

Description:
The `IntegrationLevel` property determines the operation mode of the control. See also Integration at Application Level 2055 and Integration at Document Level 2057 for more information.

Note: It is important to set this property immediately after the creation of the `XMLSpyControl` object.

29.4.7.3.1.7 MainMenu

Property: `MainMenu` as Command 2092 (read-only)

Dispatch Id: *1003*

Description:
This property provides information about the structure and commands available in the XMLSpyControl main menu, as a `Command` object. The `Command` object contains all available submenus of XMLSpy (for example "File", "Edit", "View" etc.). To access the submenu objects, use the `SubCommands` property of the `MainMenu` property. Each submenu is also a `Command` object. For each submenu, you can then further iterate through their `SubCommands` property in order to get their corresponding child commands and separators (this technique may be used, for example, to create the application menu programmatically). Note that some menu commands act as containers ("parents") for other menu commands, in which case they also have a `SubCommands` property. To get the structure of all menu commands programmatically, you will need a recursive function.

```
public void GetXmlSpyMenus()
{
    // Get the main menu from the XMLSpy ActiveX control assigned to the current form
    XMLSpyControlLib.XMLSpyCommand mainMenu = this.axXMLSpyControl1.MainMenu;

    // Loop through entries of the main menu (e.g. File, Edit, etc.)
    for (int i = 0; i < mainMenu.SubCommands.Count; i++)
    {
      XMLSpyControlLib.XMLSpyCommand menu = mainMenu.SubCommands[i];
      Console.WriteLine("{0} menu has {1} children items (including separators)",
menu.Label.Replace("&", ""), menu.SubCommands.Count);
    }
}
```

C# example

29.4.7.3.1.8 Toolbars

Property: `Toolbars` as Commands 2094 (read-only)

Dispatch Id: *1005*

Description:
This property provides information about the structure of XMLSpyControl toolbars, as a `Command` object. The `Command` object contains all available toolbars of XMLSpy. To access the toolbars, use the `SubCommands` property of the `Toolbars` property. Each toolbar is also a `Command` object. For each toolbar, you can then further iterate through their `SubCommands` property in order to get their commands (this technique may be used, for example, to create the application's toolbars programmatically).

```
public void GetXmlSpyToolbars()
{
    // Get the application toolbars from the StyleVision ActiveX control assigned to the
current form
    XMLSpyControlLib.XMLSpyCommands toolbars = this.axXMLSpyControl1.Toolbars;

    // Iterate through all toolbars
    for (int i = 0; i < toolbars.Count; i++)
    {
      XMLSpyControlLib.XMLSpyCommand toolbar = toolbars[i];
      Console.WriteLine();
      Console.WriteLine("The toolbar \"{0}\" has the following commands:",
toolbar.Label);

      // Iterate through all commands of this toolbar
      for (int j = 0; j < toolbar.SubCommands.Count; j++)
      {
         XMLSpyControlLib.XMLSpyCommand cmd = toolbar.SubCommands[j];
         // Output only command objects that are not separators
         if (!cmd.IsSeparator)
         {
            Console.WriteLine("{0}, {1}, {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&",
```


© 2016-2022 Altova GmbH

```
""));
            }
        }
    }
}
```

C# example

29.4.7.3.2 Methods

The following methods are defined:

Open 2099
Exec 2099
QueryStatus 2099

29.4.7.3.2.1 Exec

Method: `Exec` (`nCmdID` as long) as boolean

Dispatch Id: *6*

Description:
This method calls the XMLSpy command with the ID `nCmdID`. If the command can be executed, the method returns `true`. To get a list of all available commands, use `CommandsList` 2096. To retrieve the status of any command, use `QueryStatus` 2099.

29.4.7.3.2.2 Open

Method: `Open` (`strFilePath` as string) as boolean

Dispatch Id: *5*

Description:
The result of the method depends on the extension passed in the argument `strFilePath`. If the file extension is `.sps`, a new document is opened. If the file extension is `.svp`, the corresponding project is opened. If a different file extension is passed into the method, the control tries to load the file as a new component into the active document.

Do not use this method to load documents or projects when using the control in document-level integration mode. Instead, use `XMLSpyControlDocument.Open` 2105 and `XMLSpyControlPlaceHolder.OpenProject` 2110.

29.4.7.3.2.3 QueryStatus

Method: `QueryStatus` (`nCmdID` as long) as long

Dispatch Id: *7*

Description:
`QueryStatus` returns the enabled/disabled and checked/unchecked status of the command specified by `nCmdID`. The status is returned as a bit mask.

Bit	Value	Name	Meaning
0	1	Supported	Set if the command is supported.
1	2	Enabled	Set if the command is enabled (can be executed).
2	4	Checked	Set if the command is checked.

This means that if `QueryStatus` returns `0` the command ID is not recognized as a valid XMLSpy command. If `QueryStatus` returns a value of `1` or `5`, the command is disabled.

29.4.7.3.3 Events

The XMLSpyControl ActiveX control provides the following connection point events:

`OnUpdateCmdUI` 2102
`OnOpenedOrFocused` 2101
`OnCloseEditingWindow` 2100
`OnFileChangedAlert` 2101
`OnContextChanged` 2100

`OnDocumentOpened` 2101
`OnValidationWindowUpdated` 2102

29.4.7.3.3.1 OnCloseEditingWindow

Event: `OnCloseEditingWindow` (i_strFilePath as String) as boolean

Dispatch Id: *1002*

Description:
This event is triggered when XMLSpy needs to close an already open document. As an answer to this event, clients should close the editor window associated with *i_strFilePath*. Returning *true* from this event indicates that the client has closed the document. Clients can return *false* if no specific handling is required and XMLSpyControl should try to close the editor and destroy the associated document control.

29.4.7.3.3.2 OnContextChanged

Event: `OnContextChanged` (`i_strContextName` as String, `i_bActive` as bool) as bool

Dispatch Id: *1004*

Description:
This event is triggered when XMLSpy activates or de-actives one of the following operational contexts:

© 2016-2022 Altova GmbH

- XSLT Profiling - "XSLTProfiling" is passed as the context name
- XSLT / XQuery debugging - "DebuggingXSLT" is passed as the context name
- SOAP debugging - "DebuggingSOAP" is passed as the context name (Enterprise edition only)

29.4.7.3.3.3 OnDocumentOpened

Event: OnDocumentOpened (objDocument as Document)

Dispatch Id: *1*

Description:
This event is triggered whenever a document is opened. The argument objDocument is a Document object from the XMLSpy automation interface and can be used to query for more details about the document, or perform additional operations. When integrating on document-level, it is often better to use the event XMLSpyControlDocument.OnDocumentOpened[2108] instead.

29.4.7.3.3.4 OnFileChangedAlert

Event: OnFileChangedAlert (i_strFilePath as String) as bool

Dispatch Id: *1001*

Description:
This event is triggered when a file loaded with XMLSpyControl is changed on the hard disk by another application. Clients should return true, if they handled the event, or false, if XMLSpy should handle it in its customary way, i.e. prompting the user for reload.

29.4.7.3.3.5 OnLicenseProblem

Event: OnLicenseProblem (i_strLicenseProblemText as String)

Dispatch Id: *1005*

Description:
This event is triggered when XMLSpyControl detects that no valid license is available for this control. In case of restricted user licenses this can happen some time after the control has been initialized. Integrators should use this event to disable access to this control's functionality. After returning from this event, the control will block access to its functionality (e.g. show empty windows in its controls and return errors on requests).

29.4.7.3.3.6 OnOpenedOrFocused

Event: OnOpenedOrFocused (i_strFilePath as String, i_bOpenWithThisControl as bool)

Dispatch Id: *1000*

Description:

When integrating at application level, this event informs clients that a document has been opened, or made active by XMLSpy.

When integrating at document level, this event instructs the client to open the file `i_strFilePath` in a document window. If the file is already open, the corresponding document window should be made the active window.

if `i_bOpenWithThisControl` is `true`, the document must be opened with XMLSpyControl, since internal access is required. Otherwise, the file can be opened with different editors.

29.4.7.3.3.7 OnToolWindowUpdated

Event: `OnToolWindowUpdated(pToolWnd` as long `)`

Dispatch Id: *1006*

Description:
This event is triggered when the tool window is updated.

29.4.7.3.3.8 OnUpdateCmdUI

Event: `OnUpdateCmdUI ()`

Dispatch Id: *1003*

Description:
Called frequently to give integrators a good opportunity to check status of XMLSpy commands using `XMLSpyControl.QueryStatus`[2099]. Do not perform long operations in this callback.

29.4.7.3.3.9 OnValidationWindowUpdated

Event: `OnValidationWindowUpdated` ()

Dispatch Id: *3*

Description:
This event is triggered whenever the validation output window is updated with new information.

29.4.7.4 XMLSpyControlDocument

Properties:
`Appearance`[2103]
`BorderStyle`[2104]
`Document`[2104]
`IsModified`[2104]
`Path`[2104]
`ReadOnly`[2104]

© 2016-2022 Altova GmbH

Methods:
Exec 2105
New 2105
Open 2105
QueryStatus 2106
Reload 2106
Save 2106
SaveAs 2106

Events:
OnDocumentOpened 2108
OnDocumentClosed 2107
OnModifiedFlagChanged 2108
OnContextChanged 2107
OnFileChangedAlert 2108
OnActivate 2107

If the XMLSpyControl is integrated in the Document Level mode each document is displayed in an own object of type `XMLSpyControlDocument`. The `XMLSpyControlDocument` contains only one document at the time but can be reused to display different files one after another.

This object is a complete ActiveX control.

```
CLSID: 52A552E6-2AB8-4e3e-B545-BE998233DDA0
ProgID: Altova.XMLSpyControlDocument
```

29.4.7.4.1 Properties

The following properties are defined:

ReadOnly 2104
IsModified 2104
Path 2104
Appearance 2103
BorderStyle 2104

Access to XMLSpyAPI:
Document 2104

29.4.7.4.1.1 Appearance

Property: `Appearance` as short

Dispatch Id: *-520*

Description:
A value not equal to `0` displays a client edge around the document control. Default value is `0`.

29.4.7.4.1.2 BorderStyle

Property: `BorderStyle` as short

Dispatch Id: *-504*

Description:
A value of `1` displays the control with a thin border. Default value is `0`.

29.4.7.4.1.3 Document

Property: `Document` as Document

Dispatch Id: *1*

Description:
The `Document` property gives access to the `Document` object of the XMLSpy automation server API. This interface provides additional functionality which can be used with the document loaded in the control. The property is read-only.

29.4.7.4.1.4 IsModified

Property: `IsModified` as boolean (read-only)

Dispatch Id: *1006*

Description:
`IsModified` is *true* if the document content has changed since the last open, reload or save operation. It is *false*, otherwise.

29.4.7.4.1.5 Path

Property: `Path` as string

Dispatch Id: *1005*

Description:
Sets or gets the full path name of the document loaded into the control.

29.4.7.4.1.6 ReadOnly

Property: `ReadOnly` as boolean

Dispatch Id: *1007*

© 2016-2022 Altova GmbH

Description:
Using this property you can turn on and off the read-only mode of the document. If `ReadOnly` is `true` it is not possible to do any modifications.

29.4.7.4.2 Methods

The following methods are defined:

Document handling:
`New`[2105]
`Open`[2105]
`Reload`[2106]
`Save`[2106]
`SaveAs`[2106]

Command Handling:
`Exec`[2105]
`QueryStatus`[2106]

29.4.7.4.2.1 Exec

Method: `Exec` (`nCmdID` as long) as boolean

Dispatch Id: *8*

Description:
`Exec` calls the XMLSpy command with the ID `nCmdID`. If the command can be executed, the method returns `true`. This method should be called only if there is currently an active document available in the application.

To get commands organized according to their menu structure, use the `MainMenu`[2097] property of XMLSpyControl. To get toolbar commands, use the `Toolbars`[2098] property of the XMLSpyControl.

29.4.7.4.2.2 New

Method: `New` () as boolean

Dispatch Id: *1000*

Description:
This method initializes a new document inside the control.

29.4.7.4.2.3 Open

Method: `Open` (`strFileName` as string) as boolean

Dispatch Id: *1001*

Description:
`Open` loads the file `strFileName` as the new document into the control.

29.4.7.4.2.4 QueryStatus

Method: `QueryStatus` (`nCmdID` as long) as long

Dispatch Id: *9*

Description:
`QueryStatus` returns the enabled/disabled and checked/unchecked status of the command specified by `nCmdID`. The status is returned as a bit mask.

Bit	Value	Name	Meaning
0	1	Supported	Set if the command is supported.
1	2	Enabled	Set if the command is enabled (can be executed).
2	4	Checked	Set if the command is checked.

This means that if `QueryStatus` returns `0` the command ID is not recognized as a valid XMLSpy command. If `QueryStatus` returns a value of `1` or `5` the command is disabled. The client should call the `QueryStatus` method of the document control if there is currently an active document available in the application.

29.4.7.4.2.5 Reload

Method: `Reload` () as boolean

Dispatch Id: *1002*

Description:
`Reload` updates the document content from the file system.

29.4.7.4.2.6 Save

Method: `Save` () as boolean

Dispatch Id: *1003*

Description:
`Save` saves the current document at the location `Path` 2104.

29.4.7.4.2.7 SaveAs

Method: `SaveAs` (`strFileName` as string) as boolean

Dispatch Id: *1004*

© 2016-2022 Altova GmbH

Description:
`SaveAs` sets `Path` [2104] to *strFileName* and then saves the document to this location.

29.4.7.4.3 Events

The XMLSpyControlDocument ActiveX control provides following connection point events:

`OnDocumentOpened` [2108]
`OnDocumentClosed` [2107]
`OnModifiedFlagChanged` [2108]
`OnContextChanged` [2107]
`OnFileChangedAlert` [2108]
`OnActivate` [2107]
`OnSetEditorTitle` [2109]

29.4.7.4.3.1 OnActivate

Event: `OnActivate` ()

Dispatch Id: *1005*

Description:
This event is triggered when the document control is activated, has the focus, and is ready for user input.

29.4.7.4.3.2 OnContextChanged

Event: `OnContextChanged` (`i_strContextName` as String, `i_bActive` as bool) as bool

Dispatch Id: *1004*

Description:
This event is triggered when this document is shown in a different XMLSpy view. The following values are passed:

- Grid view - `"View_0"` is passed as the context name
- Text view - `"View_1"` is passed as the context name
- Browser view - `"View_2"` is passed as the context name
- Schema view - `"View_3"` is passed as the context name
- Authentic view - `"View_4"` is passed as the context name
- WSDL view - `"View_5"` is passed as the context name

29.4.7.4.3.3 OnDocumentClosed

Event: `OnDocumentClosed` (`objDocument` as Document)

Dispatch Id: *1001*

Description:
This event is triggered whenever the document loaded into this control is closed. The argument `objDocument` is a `Document` object from the XMLSpy automation interface and should be used with care.

29.4.7.4.3.4 OnDocumentOpened

Event: `OnDocumentOpened` (`objDocument` as Document)

Dispatch Id: *1000*

Description:
This event is triggered whenever a document is opened in this control. The argument `objDocument` is a `Document` object from the XMLSpy automation interface, and can be used to query for more details about the document, or perform additional operations.

29.4.7.4.3.5 OnDocumentSaveAs

Event: `OnContextDocumentSaveAs` (`i_strFileName` as String)

Dispatch Id: *1007*

Description:
This event is triggered when this document gets internally saved under a new name.

29.4.7.4.3.6 OnFileChangedAlert

Event: `OnFileChangedAlert` () as bool

Dispatch Id: *1003*

Description:
This event is triggered when the file loaded into this document control is changed on the hard disk by another application. Clients should return true, if they handled the event, or false, if XMLSpy should handle it in its customary way, i.e. prompting the user for reload.

29.4.7.4.3.7 OnModifiedFlagChanged

Event: `OnModifiedFlagChanged` (`i_bIsModified` as boolean)

Dispatch Id: *1002*

Description:
This event gets triggered whenever the document changes between modified and unmodified state. The parameter *i_bIsModifed* is *true* if the document contents differs from the original content, and *false*, otherwise.

© 2016-2022 Altova GmbH

29.4.7.4.3.8 OnSetEditorTitle

Event: OnSetEditorTitle ()

Dispatch Id: *1006*

Description:
This event is being raised when the contained document is being internally renamed.

29.4.7.5 XMLSpyControlPlaceHolder

Properties available for all kinds of placeholder windows:
`PlaceholderWindowID` 2110

Properties for project placeholder window:
`Project` 2110

Methods for project placeholder window:
`OpenProject` 2110
`CloseProject` 2110

The `XMLSpyControlPlaceHolder` control is used to show the additional XMLSpy windows like Overview, Library or Project window. It is used like any other ActiveX control and can be placed anywhere in the client application.

```
CLSID: 135DEEF4-6DF0-47c2-8F8C-F145F5F3F672
ProgID: Altova.XMLSpyControlPlaceHolder
```

29.4.7.5.1 Properties

The following properties are defined:

`PlaceholderWindowID` 2110

Access to XMLSpyAPI:
`Project` 2110

29.4.7.5.1.1 Label

Property: `Label` as String (read-only)

Dispatch Id: *1001*

Description:
This property gives access to the title of the placeholder. The property is read-only.

29.4.7.5.1.2 PlaceholderWindowID

Property: `PlaceholderWindowID as XMLSpyControlPlaceholderWindow`[2112]

Dispatch Id: *1*

Description:
This property specifies which XMLSpy window should be displayed in the client area of the control. The `PlaceholderWindowID` can be set at any time to any valid value of the `XMLSpyControlPlaceholderWindow`[2112] enumeration. The control changes its state immediately and shows the new XMLSpy window.

29.4.7.5.1.3 Project

Property: `Project` as Project (read-only)

Dispatch Id: *2*

Description:
The `Project` property gives access to the `Project` object of the XMLSpy automation server API. This interface provides additional functionality which can be used with the project loaded into the control. The property will return a valid project interface only if the placeholder window has `PlaceholderWindowID`[2110] with a value of `XMLSpyXProjectWindow (=3)`. The property is read-only.

29.4.7.5.2 Methods

The following method is defined:

`OpenProject`[2110]
`CloseProject`[2110]

29.4.7.5.2.1 OpenProject

Method: `OpenProject` (`strFileName` as `string`) as `boolean`
Dispatch Id: *3*
Description:
`OpenProject` loads the file `strFileName` as the new project into the control. The method will fail if the placeholder window has a **`PlaceholderWindowID`**[2110] different to `XMLSpyXProjectWindow (=3)`.

29.4.7.5.2.2 CloseProject

Method: `CloseProject ()`

Dispatch Id: *4*

© 2016-2022 Altova GmbH

Description:
`CloseProject` closes the project loaded by the control. The method will fail if the placeholder window has a `PlaceholderWindowID`[2110] different to `XMLSpyXProjectWindow (=3)`.

29.4.7.5.3 Events

The XMLSpyControlPlaceholder ActiveX control provides following connection point events:

`OnModifiedFlagChanged`[2111]

29.4.7.5.3.1 OnModifiedFlagChanged

Event: `OnModifiedFlagChanged` (`i_bIsModified` as boolean)

Dispatch Id: *1*

Description:
This event gets triggered only for placeholder controls with a `PlaceholderWindowID`[2110] of `XMLSpyXProjectWindow (=3)`. The event is fired whenever the project content changes between modified and unmodified state. The parameter *i_bIsModifed* is *true* if the project contents differs from the original content, and *false*, otherwise.

29.4.7.5.3.2 OnSetLabel

Event: `OnSetLabel(i_strNewLabel` as string)

Dispatch Id: *1000*

Description:
Raised when the title of the placeholder window is changed.

29.4.7.6 Enumerations

The following enumerations are defined:

`ICActiveXIntegrationLevel`[2111]
`XMLSpyControlPlaceholderWindow`[2112]

29.4.7.6.1 ICActiveXIntegrationLevel

Possible values for the IntegrationLevel[2097] property of the XMLSpyControl.

```
ICActiveXIntegrationOnApplicationLevel  = 0
ICActiveXIntegrationOnDocumentLevel     = 1
```

29.4.7.6.2 XMLSpyControlPlaceholderWindow

This enumeration contains the list of the supported additional XMLSpy windows.

```
XMLSpyControlNoToolWnd                          = -1
XMLSpyControlEntryHelperTopToolWnd       = 0
XMLSpyControlEntryHelperMiddleToolWnd           = 1
XMLSpyControlEntryHelperBottomToolWnd           = 2
XMLSpyControlValidatorOutputToolWnd             = 3
XMLSpyControlProjectWindowToolWnd               = 4
XMLSpyControlXSLTDebuggerContextToolWnd         = 5
XMLSpyControlXSLTDebuggerCallstackToolWnd= 6
XMLSpyControlXSLTDebuggerVariableToolWnd = 7
XMLSpyControlXSLTDebuggerWatchToolWnd           = 8
XMLSpyControlXSLTDebuggerTemplateToolWnd = 9
XMLSpyControlXSLTDebuggerInfoToolWnd            = 10
XMLSpyControlXSLTDebuggerMessageToolWnd         = 11
XMLSpyControlXSLTDebuggerTraceToolWnd           = 12
XMLSpyControlSOAPDebuggerToolWnd                = 13
XMLSpyControlXPathProfilerListToolWnd           = 14
XMLSpyControlXPathProfilerTreeToolWnd           = 15
XMLSpyControlXPathDialogToolWnd                 = 16
XMLSpyControlDBQueryManagerToolWnd       = 17
XMLSpyControlInfoToolWnd                        = 18
XMLSpyControlXSLOutlineToolWnd                  = 19
XMLSpyControlSchemaFindToolWnd                  = 20
XMLSpyControlXBRLFindToolWnd                    = 21
XMLSpyControlChartsToolWnd                      = 22
```


© 2016-2022 Altova GmbH

30 Anexos

Estos anexos contienen información técnica sobre XMLSpy e información importante sobre las licencias. Cada anexo contiene varios apartados.

Información sobre motores 2114

- Información sobre motores XSLT y XQuery 2114
- Funciones de extensión XSLT y XQuery 2120

Conversión de tipos de datos de BD y XML Schema 2248

- De BD a XML Schema 2233
- De XML Schema a BD 2240

Datos técnicos 2248

- Requisitos de sistema operativo y de memoria 2248
- Motores XSLT y XQuery de Altova 2248
- Compatibilidad con Unicode 2249
- Uso de Internet 2249

Información sobre licencias 2251

- Distribución electrónica de software 2251
- Activación del software y medición de licencias 2252
- Contrato de licencia para el usuario final 2253

30.1 Información sobre motores XSLT y XQuery

Los motores XSLT y XQuery de XMLSpy siguen las especificaciones del W3C y, por tanto, son más estrictos que otros motores anteriores de Altova, como los de las versiones antiguas de XMLSpy. Por consiguiente, XMLSpy señala algunos errores leves que antes no se notificaban en la versión anterior de estos motores.

Por ejemplo:

- Se notifica un error de tipo (`err:XPTY0018`) si el resultado de un operador de ruta de acceso contiene tanto nodos como no nodos.
- Se notifica un error de tipo (`err:XPTY0019`) si `E1` en una expresión XPath `E1/E2` no da como resultado una secuencia de nodos.

Si encuentra este tipo de errores, modifique el documento XSLT/XQuery o el documento de instancia según corresponda.

Esta sección describe características relacionadas con la implementación de los motores e incluye estos apartados:

- XSLT 1.0 2114
- XSLT 2.0 2114
- XSLT 3.0 2116
- XQuery 1.0 2117
- XQuery 3.1 2119

30.1.1 XSLT 1.0

El motor XSLT 1.0 de XMLSpy cumple con la recomendación XSLT 1.0 del 16 de noviembre de 1999 y con la recomendación XPath 1.0 del 16 de noviembre de 1999, ambas del W3C.

Nota sobre la implementación

Cuando el atributo `method` de `xsl:output` tiene el valor HTML o si selecciona de forma predeterminada el formato de salida HTML, los caracteres especiales del archivo XML o XSLT se insertan en el documento HTML como referencias de caracteres HTML. Por ejemplo, el carácter U+00A0 (la referencia de carácter hexadecimal para un espacio de no separación) se inserta en el código HTML como referencia de carácter (` ` o ` `) o como referencia de entidad (` `).

30.1.2 XSLT 2.0

Temas de este apartado:

- Especificaciones con las que cumple el motor 2115
- Compatibilidad con versiones antiguas 2115
- Espacios de nombres 2115
- Compatibilidad con esquemas 2116

© 2016-2022 Altova GmbH

- Comportamiento propio de esta implementación 2116

Especificaciones

El motor XSLT 2.0 de XMLSpy cumple con la recomendación XSLT 2.0 del 23 de enero de 2007 y la recomendación XPath 2.0 del 14 de diciembre de 2010, ambas del W3C.

Compatibilidad con versiones antiguas

El motor XSLT 2.0 es compatible con versiones previas. Esto solamente es relevante cuando se utiliza el motor XSLT 2.0 para procesar una hoja de estilos XSLT 1.0. Tenga en cuenta que los resultados obtenidos con el motor XSLT 1.0 pueden ser diferentes a los obtenidos con el motor XSLT 2.0 en modo de compatibilidad con versiones antiguas.

Espacios de nombres

En su hoja de estilos XSLT 2.0 debe declarar estos espacios de nombres para poder usar los constructores de tipo y las funciones disponibles en XSLT 2.0. Los prefijos que aparecen a continuación son los que se suelen usar, pero puede usar otros prefijos si quiere.

Espacio de nombres	Prefijo	URI del espacio de nombres
Tipos XML Schema	`xs:`	`http://www.w3.org/2001/XMLSchema`
Funciones XPath 2.0	`fn:`	`http://www.w3.org/2005/xpath-functions`

Estos espacios de nombres se suelen declarar en el elemento `xsl:stylesheet` o en el elemento `xsl:transform`:

```
<xsl:stylesheet version="2.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:fn="http://www.w3.org/2005/xpath-functions"
   ...
</xsl:stylesheet>
```

Es necesario tener en cuenta que:

- El motor XSLT 2.0 utiliza el espacio de nombres Funciones XPath 2.0 y XQuery 1.0 como **espacio de nombres de funciones predeterminado**. Esto significa que puede usar funciones XPath 2.0 y XSLT 2.0 en su hoja de estilos sin prefijos. Si declara el espacio de nombres Funciones XPath 2.0 en su hoja de estilos con un prefijo, podrá usar el prefijo asignado en la declaración.
- Cuando se usan constructores de tipo y tipos del espacio de nombres XML Schema, el prefijo utilizado en la declaración de espacio de nombres se debe usar en la llamada al constructor de tipo (por ejemplo, `xs:date`).
- Algunas funciones XPath 2.0 se llaman igual que algunos tipos de datos de XML Schema. Por ejemplo, las funciones XPath `fn:string` y `fn:boolean` y los tipos de datos de XML Schema `xs:string` y `xs:boolean`. Por tanto, si usa la expresión `string('Hello')`, la expresión se evalúa como `fn:string('Hello')` y no como `xs:string('Hello')`.

Compatibilidad con esquemas

El motor XSLT 2.0 está preparado para esquemas de modo que puede usar tipos de esquema definidos por el usuario y la instrucción `xsl:validate`.

Comportamiento propio de esta implementación

Más abajo puede ver cómo se ocupa el motor XSLT 2.0 de algunos aspectos del comportamiento de las funciones XSLT 2.0 relacionadas con esta implementación.

`xsl:result-document`
También son compatibles estas codificaciones específicas de Altova: `x-base16tobinary` y `x-base64tobinary`.

`function-available`
Esta función mira si hay funciones del ámbito disponibles (funciones XSLT, XPath y de extensión).

`unparsed-text`
El atributo `href` acepta (i) rutas de acceso relativas para archivos que estén en la carpeta del URI base y (ii) rutas de acceso absolutas con o sin el protocolo `file://`. También son compatibles estas codificaciones específicas de Altova: `x-binarytobase16` y `x-binarytobase64`.

`unparsed-text-available`
El atributo `href` acepta (i) rutas de acceso relativas para archivos que estén en la carpeta del URI base y (ii) rutas de acceso absolutas con o sin el protocolo `file://`. También son compatibles estas codificaciones específicas de Altova: `x-binarytobase16` y `x-binarytobase64`.

Nota: estos valores de codificación estaban implementados en el ya descatalogado AltovaXML pero ya no se utilizan (son obsoletos): `base16tobinary`, `base64tobinary`, `binarytobase16` y `binarytobase64`.

30.1.3 XSLT 3.0

El motor XSLT 3.0 de XMLSpy cumple con la propuesta de recomendación XSLT 3.0 del 8 de junio de 2017 y con la propuesta de recomendación XPath 3.1 del 21 de marzo de 2017 del consorcio W3C.

El motor XSLT 3.0 tiene las mismas características de implementación que el motor XSLT 2.0[2114]. Pero además ofrece compatibilidad con muchas de las nuevas funciones XSLT3.0, con las funciones y los operadores XPath/XQuery 3.1 y con la especificación XPath 3.1.

Nota: la característica opcional de transmisión por secuencias no es compatible por ahora. Todo el documento se cargará en memoria independientemente del valor del atributo `streamable`. Si no hay suficiente memoria, (i) se procesa todo el documento sin transmisión de secuencias, (ii) se procesan los constructores "guaranteed-streamable" como si se estuviera usando transmisión por secuencias y (iii) los errores de transmisión de secuencias no se reconocen. En las aplicaciones de 64 bits esto no debería causar problemas. Sin embargo, si se dan problemas de memoria, una solución sería añadir más memoria al sistema.

© 2016-2022 Altova GmbH

30.1.4 XQuery 1.0

Temas de este apartado:

- Especificaciones con las que cumple el motor 2117
- Compatibilidad con esquemas 2117
- Codificación 2117
- Espacios de nombres 2117
- Fuentes XML y validación 2118
- Comprobación de tipos estática y dinámica 2118
- Módulos biblioteca 2118
- Funciones externas 2119
- Intercalaciones 2119
- Precisión de datos numéricos 2119
- Compatibilidad con instrucciones XQuery 2119

Especificaciones compatibles

El motor XQuery 1.0 de XMLSpy cumple con la recomendación XQuery 1.0 del 14 de diciembre de 2010 del W3C. El estándar XQuery concede libertad a la hora de implementar muchas características. A continuación explicamos cómo se implementaron estas características en el motor XQuery 1.0 de XMLSpy.

Compatibilidad con esquemas

El motor XQuery 1.0 está preparado para esquemas.

Codificación

El motor XQuery 1.0 es compatible con las codificaciones de caracteres UTF-8 y UTF-16.

Espacios de nombres

Se predefinen estos URI de espacios de nombres y sus enlaces asociados.

Espacio de nombres	Prefijo	URI del espacio de nombres
Tipos XML Schema	`xs:`	`http://www.w3.org/2001/XMLSchema`
Schema instance	`xsi:`	`http://www.w3.org/2001/XMLSchema-instance`
Funciones integradas	`fn:`	`http://www.w3.org/2005/xpath-functions`
Funciones locales	`local:`	`http://www.w3.org/2005/xquery-local-functions`

Es importante tener en cuenta que:

- El motor XQuery 1.0 entiende que los prefijos de la tabla anterior están enlazados con los correspondientes espacios de nombres.
- Como el espacio de nombres de funciones integradas (ver tabla) es el espacio de nombres de funciones predeterminado de XQuery, no es necesario usar el prefijo `fn:` cuando se invocan funciones

integradas (por ejemplo, `string("Hello")` llamará a la función `fn:string`). No obstante, el prefijo `fn:` se puede utilizar para llamar a una función integrada sin necesidad de declarar el espacio de nombres en el prólogo de la consulta (por ejemplo: `fn:string("Hello")`).
- Puede cambiar el espacio de nombres de funciones predeterminado declarando la expresión `default function namespace` en el prólogo de la consulta.
- Cuando use tipos del espacio de nombres XML Schema, puede usar el prefijo `xs:` sin necesidad de declarar los espacios de nombres de forma explícita ni enlazar estos prefijos a los espacios de nombres en el prólogo de la consulta. (Ejemplo: `xs:date` y `xs:yearMonthDuration`.) Si quiere usar otros prefijos para el espacio de nombres de XML Schema, estos se deben declarar en el prólogo de la consulta. (Ejemplo: `declare namespace alt = "http://www.w3.org/2001/XMLSchema"; alt:date("2004-10-04").`)
- Recuerde que los tipos de datos `untypedAtomic`, `dayTimeDuration` y `yearMonthDuration` se movieron del espacio de nombres XPath Datatypes al espacio de nombres XML Schema (es decir, ahora es `xs:yearMonthDuration`.)

Si se asignaron mal los espacios de nombres para funciones, constructores de tipo, pruebas de nodo, etc., se emite un error. Sin embargo, recuerde que algunas funciones se llaman igual que los tipos de datos de esquema (p. ej. `fn:string` y `fn:boolean`.) (Se definen `xs:string` y `xs:boolean`.) El prefijo del espacio de nombres determina si se usa la función o el constructor de tipo.

Documento XML de origen y validación

Los documentos XML que se utilizan para ejecutar un documento XQuery con el motor XQuery 1.0 deben tener un formato XML correcto. Sin embargo, no es necesario que sean válidos con respecto a un esquema XML. Si el archivo no es válido, el archivo no válido se carga sin información de esquema. Si el archivo XML está asociado a un esquema externo y es válido con respecto a dicho esquema, se genera información posterior a la validación de esquema, que se utilizará para evaluar la consulta.

Comprobación de tipos estática y dinámica

En la fase de análisis estático se revisan aspectos de la consulta como la sintaxis, si existen referencias externas (p. ej. para módulos), si las funciones y variables que se invocan están definidas, etc. Si se detecta un error en la fase de análisis estático, se notifica y la ejecución se interrumpe.

La comprobación dinámica de tipos se realiza en tiempo de ejecución, cuando la consulta se ejecuta. Si un tipo no es compatible con los requisitos de una operación, se emite un error. Por ejemplo, la expresión `xs:string("1") + 1` devuelve un error porque la operación de suma no se puede llevar a cabo en un operando de tipo `xs:string`.

Módulos biblioteca

Los módulos biblioteca almacenan funciones y variables para poder volver a utilizarlas. El motor XQuery 1.0 es compatible con el uso de módulos almacenados en un **solo archivo XQuery externo**. Dicho archivo de módulo debe incluir una declaración `module` en su prólogo que apunte a un espacio de nombres de destino Por ejemplo:

```
module namespace libns="urn:module-library";
declare variable $libns:company := "Altova";
declare function libns:webaddress() { "http://www.altova.com" };
```

Todas las funciones y variables declaradas en el módulo pertenecen al espacio de nombres asociado al módulo. El módulo se importa en un archivo XQuery con la instrucción `import module` del prólogo de la

 © 2016-2022 Altova GmbH

consulta. La instrucción `import module` solamente importa funciones y variables declaradas directamente en el archivo de módulo biblioteca. Por ejemplo:

```
import module namespace modlib = "urn:module-library" at "modulefilename.xq";
if         ($modlib:company = "Altova")
then        modlib:webaddress()
else        error("No match found.")
```

Funciones externas

Las funciones externas son incompatibles con el motor XQuery 1.0, es decir, todas las expresiones que usen la palabra clave `external`. Por ejemplo:

```
declare function hoo($param as xs:integer) as xs:string external;
```

Intercalaciones

La intercalación predeterminada es la intercalación de puntos de código Unicode, que compara las cadenas de texto según sus puntos de código Unicode. Otras intercalaciones compatibles son las intercalaciones ICU que se enumeran aquí[2120]. Para usar una intercalación concreta, indique su URI tal y como aparece en la lista de intercalaciones compatibles[2120]. Las comparaciones de cadenas de texto, incluidas las comparaciones para las funciones `fn:max` y `fn:min`, se harán según la intercalación especificada. Si no se indica la opción de intercalación, se utiliza la intercalación de puntos de código Unicode predeterminada.

Precisión de tipos numéricos

- El tipo de datos `xs:integer` es de precisión arbitraria, es decir, puede representar un número de dígitos cualquiera.
- El tipo de datos `xs:decimal` tiene un límite de 20 dígitos después del punto decimal.
- Los tipos de datos `xs:float` y `xs:double` tienen una precisión limitada de 15 dígitos.

Compatibilidad con instrucciones XQuery

La instrucción `Pragma` no es compatible. Si se encuentra, se ignora y en su lugar se evalúa la expresión de reserva.

30.1.5 XQuery 3.1

El motor XQuery 3.1 de XMLSpy cumple con la propuesta de recomendación XQuery 3.1 del 21 de marzo de 2017 del consorcio W3C y es compatible con funciones XPath y XQuery 3.1. La especificación XQuery 3.1 es un supraconjunto de la especificación 3.0. El motor XQuery 3.1, por tanto, es compatible con las características de XQuery 3.0.

Tiene las mismas características de implementación que el motor XQuery 1.0[2117].

30.2 Funciones XSTL y XPath/XQuery

Esta sección enumera las funciones de extensión de Altova y otras funciones de extensión que se pueden utilizar con expresiones XPath y XQuery. Las funciones de extensión de Altova se pueden usar con los motores XSLT y XQuery de Altova y ofrecen algunas funciones más aparte de las que están disponibles en las bibliotecas de funciones definidas en los estándares del W3C.

Aspectos generales

Es necesario tener en cuenta estos puntos generales:

- A las funciones de las bibliotecas de funciones `core` definidas en las especificaciones W3C se les puede llamar sin un prefijo. Esto se debe a que los motores XSLT y XQuery leen funciones sin prefijo como si pertenecieran a un espacio de nombres de funciones predeterminado, que es el que se especifica en las especificaciones de las funciones XPath y XQuery `http://www.w3.org/2005/xpath-functions`. Si este espacio de nombres se declara explícitamente en un documento XSLT o XQuery, el prefijo utilizado en la declaración de espacio de nombres también se puede usar en el nombre de las funciones.
- Por lo general, si una función espera como argumento una secuencia de un elemento y se suministra una secuencia de más de un elemento, entonces se devuelve un error.
- Todas las comparaciones de cadena se realizan usando la intercalación de puntos de código Unicode.
- Los resultados que son QName se serializan de esta forma `[prefijo:]nombrelocal`.

Precisión de xs:decimal

El término *precisión* hace referencia al número de dígitos del número y la especificación exige un mínimo de 18 dígitos. Para las operaciones de división que dan un resultado de tipo `xs:decimal`, la precisión es de 19 dígitos después del punto decimal sin redondear.

Uso horario implícito

Cuando hay que comparar dos valores `date`, `time` o `dateTime`, es necesario conocer el uso horario de los valores que se deben comparar. Cuando el uso horario no se conoce de forma explícita, se usa el uso horario implícito, que se toma del reloj del sistema. Para probar cuál es su valor puede utilizar la función `fn:implicit-timezone()`.

Intercalaciones

La intercalación predeterminada es la intercalación de puntos de código Unicode, que compara las cadenas de texto según sus puntos de código Unicode. El motor usa el algoritmo de intercalación Unicode. Otras intercalaciones compatibles son las intercalaciones ICU que aparecen más abajo. Para usar una intercalación indique su URI tal y como aparece en la tabla más abajo. Las comparaciones de cadenas de texto (incluidas las que usan las funciones `fn:max` y `fn:min`) se harán según la intercalación especificada. Si no se especifica la opción de intercalación, se usa la intercalación predeterminada de puntos de código Unicode.

Idioma	Identificadores URI
`da`: danés	`da_DK`
`de`: alemán	`de_AT, de_BE, de_CH, de_DE, de_LI, de_LU`

© 2016-2022 Altova GmbH

en: inglés	en_AS, en_AU, en_BB, en_BE, en_BM, en_BW, en_BZ, en_CA, en_GB, en_GU, en_HK, en_IE, en_IN, en_JM, en_MH, en_MP, en_MT, en_MU, en_NA, en_NZ, en_PH, en_PK, en_SG, en_TT, en_UM, en_US, en_VI, en_ZA, en_ZW
es: español	es_419, es_AR, es_BO, es_CL, es_CO, es_CR, es_DO, es_EC, es_ES, es_GQ, es_GT, es_HN, es_MX, es_NI, es_PA, es_PE, es_PR, es_PY, es_SV, es_US, es_UY, es_VE
fr: francés	fr_BE, fr_BF, fr_BI, fr_BJ, fr_BL, fr_CA, fr_CD, fr_CF, fr_CG, fr_CH, fr_CI, fr_CM, fr_DJ, fr_FR, fr_GA, fr_GN, fr_GP, fr_GQ, fr_KM, fr_LU, fr_MC, fr_MF, fr_MG, fr_ML, fr_MQ, fr_NE, fr_RE, fr_RW, fr_SN, fr_TD, fr_TG
it: italiano	it_CH, it_IT
ja: jaoonés	ja_JP
nb: noruego bokmål	nb_NO
nl: holandés	nl_AW, nl_BE, nl_NL
nn: noruego nynorsk	nn_NO
pt: portugués	pt_AO, pt_BR, pt_GW, pt_MZ, pt_PT, pt_ST
ru: ruso	ru_MD, ru_RU, ru_UA
sv: sueco	sv_FI, sv_SE

Eje del espacio de nombres
El eje del espacio de nombres se dejó de utilizar en XPath 2.0. Sin embargo, los ejes de espacio de nombres son compatibles con la aplicación. Para acceder a la información sobre el espacio de nombres con mecanismos de XPath 2.0, utilice las funciones `fn:in-scope-prefixes()`, `fn:namespace-uri()` y `fn:namespace-uri-for-prefix()`.

30.2.1 Funciones de extensión de Altova

Puede utilizar todas las funciones de extensión de Altova en sus expresiones XPath/XQuery. Con ellas conseguirá funciones adicionales no disponibles en la biblioteca de funciones estándar de XPath, XQuery y XSLT. Las funciones de extensión de Altova están en el **espacio de nombres** `http://www.altova.com/xslt-extensions` y en esta sección se presentan con el prefijo, que se supone estará enlazado al espacio de nombres señalado. Tenga en cuenta que en futuras versiones del producto algunas funciones pueden dejar de ser compatibles o su comportamiento puede cambiar. Por tanto, consulte siempre la documentación del producto para conocer el funcionamiento de estas funciones en cada versión del producto.

Las funciones definidas en las especificaciones XPath/XQuery Functions del W3C se pueden usar en (i) expresiones XPath en contextos XSLT y en (ii) expresiones XQuery en documentos XQuery. En esta documentación las funciones que se pueden usar en el primer contexto (XPath en XSLT) llevan el símbolo `XP` y se les llama funciones XPath. Las funciones que se pueden usar en contextos XQuery llevan el símbolo `XQ` y

funcionan como funciones XQuery. Las especificaciones XSLT del W3C también definen funciones que se pueden usar en expresiones XPath en documentos XSLT. Estas funciones llevan el símbolo XSLT y se les denomina funciones XSLT. Por cada función se indica en qué versión de XPath/XQuery y XSLT se puede usar (ver símbolos más abajo). Las funciones de las bibliotecas de funciones XPath/XQuery y XSLT aparecen sin prefijo. Las funciones de extensión de otras bibliotecas, como las funciones de extensión de Altova, aparecen con un prefijo.

Funciones XPath *(en expresiones XPath en XSLT):*	XP1 XP2 XP3.1.1
Funciones XSLT *(en expresiones XPath en XSLT):*	XSLT1 XSLT2 XSLT3
Funciones XQuery *(en expresiones XQuery en XQuery):*	XQ1 XQ3.1

Funciones XSLT 2122

Las funciones XSLT solo se pueden utilizar en expresiones XPath en un contexto XSLT (igual que las funciones XSLT 2.0 `current-group()` o `key()`). Estas funciones no están pensadas para contextos no XSLT (p. ej. contextos XQuery) y, por tanto, no funcionarán en contextos que no sean XSLT. Recuerde que las funciones XSLT para XBRL solamente se pueden utilizar con ediciones de los productos de Altova compatibles con XBRL.

Funciones XPath/XQuery

Las funciones XPath/XQuery se pueden utilizar en expresiones XPath, en contextos XSLT y en expresiones XQuery:

- Funciones de fecha y hora 2125
- Funciones de geoubicación 2143
- Funciones relacionadas con imágenes 2155
- Funciones numéricas 2159
- Funciones de secuencia 2182
- Funciones de cadena 2190
- Funciones varias 2198

Funciones para gráficos (solo en las ediciones Enterprise y Server Edition)

Las funciones de extensión para gráficos de Altova 2200 son compatibles con las ediciones Enterprise y Server Edition de los productos de Altova solamente. Estas funciones permiten generar gráficos a partir de datos XML.

30.2.1.1 Funciones XSLT

Las **funciones de extensión XSLT** pueden utilizarse en expresiones XPath en contextos XSLT y no funcionan en contextos que no sean XSLT (por ejemplo, en contextos XQuery).

Nota sobre el nombre de las funciones y lenguajes

Puede utilizar todas las funciones de extensión de Altova en sus expresiones XPath/XQuery. Con ellas

© 2016-2022 Altova GmbH

conseguirá funciones adicionales no disponibles en la biblioteca de funciones estándar de XPath, XQuery y XSLT. Las funciones de extensión de Altova están en el **espacio de nombres** `http://www.altova.com/xslt-extensions` y en esta sección se presentan con el prefijo, que se supone estará enlazado al espacio de nombres señalado. Tenga en cuenta que en futuras versiones del producto algunas funciones pueden dejar de ser compatibles o su comportamiento puede cambiar. Por tanto, consulte siempre la documentación del producto para conocer el funcionamiento de estas funciones en cada versión del producto.

Funciones XPath (en expresiones XPath en XSLT):	XP1 XP2 XP3.1.1
Funciones XSLT (en expresiones XPath en XSLT):	XSLT1 XSLT2 XSLT3
Funciones XQuery (en expresiones XQuery en XQuery):	XQ1 XQ3.1

Funciones generales

▼ distinct-nodes [altova:]

altova:distinct-nodes(*node()****)** como **node()*** XSLT1 XSLT2 XSLT3
Toma un conjunto de nodos como entrada y devuelve el mismo conjunto menos los nodos que tengan el mismo valor (es decir, devuelve los nodos que son únicos). La comparación se hace con la función XPath/XQuery `fn:deep-equal`.

⊟ *Ejemplo*

- **altova:distinct-nodes**(country) devuelve todos los nodos secundarios `country` excepto los que tengan el mismo valor.

▼ evaluate [altova:]

altova:evaluate(ExpresiónXPath *como xs:string***[, ValorDe$p1, ... ValorDe$pN])** XSLT1 XSLT2 XSLT3
Toma una expresión XPath, pasada como cadena, como argumento obligatorio. Devuelve el resultado de la expresión evaluada. Por ejemplo, **evaluate('//Name[1]')** devuelve el contenido del primer elemento `Name` del documento. Observe que para pasar la expresión `//Name[1]` como cadena basta con ponerla entre comillas simples.

La función `altova:evaluate` puede tomar más argumentos, que son los valores de las variables del ámbito que se llaman `p1`, `p2`, `p3`... `pN`. Recuerde que (i) las variables deben definirse con nombres de tipo `pX`, siendo `X` un entero; (ii) los argumentos de la función `altova:evaluate` (*ver firma más abajo*), a partir del segundo argumento, ofrecen los valores de las variables, correspondiendo la secuencia de argumentos a la secuencia numérica de variables: `p1` corresponde a `pN` y el segundo argumento será el valor de la variable `p1`, el tercer argumento al de la variable `p2`, y así sucesivamente; (iii) los valores de las variables deben ser de tipo `item*`.

⊟ *Ejemplo*

```
<xsl:variable name="xpath" select="'$p3, $p2, $p1'" />
<xsl:value-of select="altova:evaluate($xpath, 10, 20, 'hi')" />
```

da el resultado `"hi 20 10"`

En el ejemplo anterior puede observar que:

- El segundo argumento de la expresión `altova:evaluate` es el valor asignado a la variable `$p1`, el tercer argumento es el valor asignado a la variable `$p2` y así sucesivamente.
- Observe que el cuarto argumento de la función es un valor de cadena porque va entre comillas simples.
- El atributo `select` del elemento `xs:variable` suministra la expresión XPath. Como esta expresión debe ser de tipo `xs:string`, se pone entre comillas simples.

Más ejemplos

- ```
  <xsl:variable name="xpath" select="'$p1'" />
  <xsl:value-of select="altova:evaluate($xpath, //Name[1])" />
  ```
  El resultado es el valor del primer elemento `Name`.

- ```
  <xsl:variable name="xpath" select="'$p1'" />
  <xsl:value-of select="altova:evaluate($xpath, '//Name[1]')" />
  ```
  El resultado es "`//Name[1]`*"*

La función de extensión `altova:evaluate()` es muy práctica cuando una expresión XPath de la hoja de estilos XSLT contiene partes que se deben evaluar de forma dinámica. Por ejemplo, imagine que el usuario selecciona un criterio de ordenación y este criterio se almacena en el atributo `UserReq/@sortkey`. En la hoja de estilos podría tener esta expresión:

```
<xsl:sort select="altova:evaluate(../UserReq/@sortkey)" order="ascending"/>
```

La función `altova:evaluate()` lee el atributo `sortkey` del elemento secundario `UserReq` del primario del nodo de contexto. Imagine que el valor del atributo `sortkey` es `Price`. En ese caso, la función `altova:evaluate()` devuelve `Price`, que se convierte en el valor del atributo `select`:

```
<xsl:sort select="Price" order="ascending"/>
```

Si esta instrucción `sort` aparece dentro del contexto de un elemento llamado `Order`, entonces los elementos `Order` se ordenan según el valor de los secundarios `Price`. Otra opción es que, si el valor de `@sortkey` fuera `Date`, por ejemplo, entonces los elementos `Order` se ordenarían según el valor de los secundarios `Date`. Es decir, el criterio de ordenación para `Order` se selecciona del atributo `sortkey` en tiempo de ejecución. Esto no sería posible con una expresión como:

```
<xsl:sort select="../UserReq/@sortkey" order="ascending"/>
```

En este caso, el criterio de ordenación sería el propio atributo `sortkey`, no `Price` ni `Date` (ni otro contenido actual de `sortkey`).

Nota: el contexto estático incluye espacios de nombres, tipos y funciones (pero no variables) del entorno de llamada. El URI base y el espacio de nombres predeterminado se heredan.

Más ejemplos

© 2016-2022 Altova GmbH

- Variables estáticas: `<xsl:value-of select="$i3, $i2, $i1" />`
  El resultado es los valores de las tres variables.

- Expresión XPath dinámica con variables dinámicas:
  `<xsl:variable name="xpath" select="'$p3, $p2, $p1'" />`
  `<xsl:value-of select="altova:evaluate($xpath, 10, 20, 30)" />`
  El resultado es "30 20 10"

- Expresión XPath dinámica sin variables dinámicas:
  `<xsl:variable name="xpath" select="'$p3, $p2, $p1'" />`
  `<xsl:value-of select="altova:evaluate($xpath)" />`
  Error: no se definió la variable para $p3.

▼ encode-for-rtf [altova:]

altova:encode-for-rtf(entrada *como xs:string*, **conservarEspaciosEnBlanco** *como xs:boolean*, **conservarLíneasNuevas** *como xs:boolean*) como **xs:string** XSLT2 XSLT3
Convierte la cadena de entrada en código para RTF. Los espacios en blanco y las líneas nuevas se conservan o no dependiendo del valor booleano especificado para los correspondientes parámetros.

[Subir 2122]

Funciones XBRL

Las funciones XBRL de Altova solo funcionan en las ediciones de los productos de Altova que son compatibles con XBRL.

▼ xbrl-footnotes [altova:]

altova:xbrl-footnotes(*node()***)** como **node()*** XSLT2 XSLT3
Toma un nodo como argumento de entrada y devuelve el conjunto de nodos de nota al pie XBRL al que hace referencia el nodo de entrada.

▼ xbrl-labels [altova:]

altova:xbrl-labels(*xs:QName*, *xs:string***)** como **node()*** XSLT2 XSLT3
Toma dos argumentos de entrada: un nombre de nodo y la ubicación del archivo de taxonomía en el que está el nodo. La función devuelve los nodos de etiqueta XBRL asociados al nodo de entrada.

[Subir 2122]

30.2.1.2 Funciones XPath/XQuery: de fecha y hora

Las funciones de extensión de fecha y hora de Altova se pueden usar en expresiones XPath y XQuery y permiten procesar datos almacenados en tipos de datos XML Schema de fecha y hora. Estas funciones se

pueden usar con los **motores XPath 3.0 y XQuery 3.0** de Altova y están disponibles en contextos XPath/XQuery.

Nota sobre el nombre de las funciones y lenguajes

Puede utilizar todas las funciones de extensión de Altova en sus expresiones XPath/XQuery. Con ellas conseguirá funciones adicionales no disponibles en la biblioteca de funciones estándar de XPath, XQuery y XSLT. Las funciones de extensión de Altova están en el **espacio de nombres** `http://www.altova.com/xslt-extensions` y en esta sección se presentan con el prefijo, que se supone estará enlazado al espacio de nombres señalado. Tenga en cuenta que en futuras versiones del producto algunas funciones pueden dejar de ser compatibles o su comportamiento puede cambiar. Por tanto, consulte siempre la documentación del producto para conocer el funcionamiento de estas funciones en cada versión del producto.

Funciones XPath (en expresiones XPath en XSLT):	XP1 XP2 XP3.1.1
Funciones XSLT (en expresiones XPath en XSLT):	XSLT1 XSLT2 XSLT3
Funciones XQuery (en expresiones XQuery en XQuery):	XQ1 XQ3.1

Funciones agrupadas según su funcionalidad

- Agregar una duración a xs:dateTime y devolver xs:dateTime 2127
- Agregar una duración a xs:date y devolver xs:date 2129
- Agregar una duración a xs:time y devolver xs:time 2131
- Recuperar duraciones y aplicarles formato 2130
- Quitar la zona horaria de las funciones que generan la fecha/hora actual 2132
- Devolver el número de días, horas, minutos y segundos de duraciones 2133
- Devolver el día de la semana de una fecha como número entero 2135
- Devolver el número de semana de una fecha como número entero 2136
- Generar la fecha, la hora y el tipo de duración a partir de los componentes léxicos de cada tipo 2137
- Construir un tipo date, dateTime o a partir de la cadena de entrada 2139
- Funciones para calcular la edad 2140
- Funciones para calcular el tiempo Unix 2142

Funciones por orden alfabético

altova:add-days-to-date 2129
altova:add-days-to-dateTime 2127
altova:add-hours-to-dateTime 2127
altova:add-hours-to-time 2131
altova:add-minutes-to-dateTime 2127
altova:add-minutes-to-time 2131
altova:add-months-to-date 2129
altova:add-months-to-dateTime 2127
altova:add-seconds-to-dateTime 2127
altova:add-seconds-to-time 2131
altova:add-years-to-date 2129
altova:add-years-to-dateTime 2127
altova:age 2140
altova:age-details 2140

© 2016-2022 Altova GmbH

altova:build-date 2137
altova:build-duration 2137
altova:build-time 2137
altova:current-dateTime-no-TZ 2132
altova:current-date-no-TZ 2132
altova:current-time-no-TZ 2132
altova:date-no-TZ 2132
altova:dateTime-from-epoch 2142
altova:dateTime-from-epoch-no-TZ 2142
altova:dateTime-no-TZ 2132
altova:days-in-month 2133
altova:epoch-from-dateTime 2142
altova:hours-from-dateTimeDuration-accumulated 2133
altova:minutes-from-dateTimeDuration-accumulated 2133
altova:seconds-from-dateTimeDuration-accumulated 2133
altova:format-duration 2130
altova:parse-date 2139
altova:parse-dateTime 2139
altova:parse-duration 2130
altova:parse-time 2139
altova:time-no-TZ 2132
altova:weekday-from-date 2135
altova:weekday-from-dateTime 2135
altova:weeknumber-from-date 2136
altova:weeknumber-from-dateTime 2136

[Subir 2125]

Agregar una duración a xs:dateTime XP3.1 XQ3.1

Estas funciones sirven para agregar una duración a **xs:dateTime** y devuelven **xs:dateTime**. El tipo xs:dateTime tiene el formato SSAA-MM-DDThh:mm:ss.sss. Se trata de la concatenación de los formatos xs:date y xs:time separados por la letra T. Si quiere puede usar un sufijo de zona horaria (por ejemplo **+01:00**).

▼ add-years-to-dateTime [altova:]

altova:add-years-to-dateTime(FechaHora *as xs:dateTime*, **Años** *as xs:integer*) como **xs:dateTime** XP3.1 XQ3.1

Añade una duración en años un valor de fecha y hora. El segundo argumento es el número de años que se debe añadir al valor de fecha y hora dado como primer argumento. El resultado es de tipo xs:dateTime.

Ejemplos

- **altova:add-years-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), 10) devuelve 2024-01-15T14:00:00
- **altova:add-years-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), -4) devuelve 2010-01-15T14:00:00

▼ add-months-to-dateTime [altova:]

altova:add-months-to-dateTime(FechaHora *as xs:dateTime*, **Meses** *as xs:integer*) como **xs:dateTime** **XP3.1** **XQ3.1**
Añade una duración en meses a un valor de fecha y hora. El segundo argumento es el número de meses que se debe añadir al valor de fecha y hora dado como primer argumento. El resultado es de tipo `xs:dateTime`.

Ejemplos

- **altova:add-months-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), 10) devuelve 2014-11-15T14:00:00
- **altova:add-months-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), -2) devuelve 2013-11-15T14:00:00

add-days-to-dateTime [altova:]

altova:add-days-to-dateTime(FechaHora *as xs:dateTime*, **Días** *as xs:integer*) como **xs:dateTime** **XP3.1** **XQ3.1**
Añade una duración en días a un valor de fecha y hora. El segundo argumento es el número de días que se deben añadir al valor de fecha y hora dado como primer argumento. El resultado es de tipo `xs:dateTime`.

Ejemplos

- **altova:add-days-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), 10) devuelve 2014-01-25T14:00:00
- **altova:add-days-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), -8) devuelve 2014-01-07T14:00:00

add-hours-to-dateTime [altova:]

altova:add-hours-to-dateTime(FechaHora *as xs:dateTime*, **Horas** *as xs:integer*) como **xs:dateTime** **XP3.1** **XQ3.1**
Añade una duración en horas a un valor de fecha y hora. El segundo argumento es el número de horas que se deben añadir al valor de fecha y hora dado como primer argumento. El resultado es de tipo `xs:dateTime`.

Ejemplos

- **altova:add-hours-to-dateTime**(xs:dateTime("2014-01-15T13:00:00"), 10) devuelve 2014-01-15T23:00:00
- **altova:add-hours-to-dateTime**(xs:dateTime("2014-01-15T13:00:00"), -8) devuelve 2014-01-15T05:00:00

add-minutes-to-dateTime [altova:]

altova:add-minutes-to-dateTime(FechaHora *as xs:dateTime*, **Minutos** *as xs:integer*) como **xs:dateTime** **XP3.1** **XQ3.1**
Añade una duración en minutos a un valor de fecha y hora. El segundo argumento es el número de minutos que se debe añadir al valor de fecha y hora dado como primer argumento. El resultado es de tipo `xs:dateTime`.

© 2016-2022 Altova GmbH

Ejemplos

- **altova:add-minutes-to-dateTime**(xs:dateTime("2014-01-15T14:10:00"), 45) devuelve 2014-01-15T14:55:00
- **altova:add-minutes-to-dateTime**(xs:dateTime("2014-01-15T14:10:00"), -5) devuelve 2014-01-15T14:05:00

add-seconds-to-dateTime [altova:]

altova:add-seconds-to-dateTime(FechaHora *as xs:dateTime*, **Segundos** *as xs:integer***)** como **xs:dateTime** **XP3.1** **XQ3.1**
Añade una duración en segundos a un valor de fecha y hora. El segundo argumento es el número de segundos que se debe añadir al valor de fecha y hora dado como primer argumento. El resultado es de tipo xs:dateTime.

Ejemplos

- **altova:add-seconds-to-dateTime**(xs:dateTime("2014-01-15T14:00:10"), 20) devuelve 2014-01-15T14:00:30
- **altova:add-seconds-to-dateTime**(xs:dateTime("2014-01-15T14:00:10"), -5) devuelve 2014-01-15T14:00:05

[Subir 2125]

Agregar una duración a xs:date **XP3.1** **XQ3.1**

Estas funciones agregan una duración a **xs:date** y devuelven **xs:date**. El tipo xs:date tiene el formato SSAA-MM-DD.

add-years-to-date [altova:]

altova:add-years-to-date(Fecha *as xs:date*, **Años** *as xs:integer***)** como **xs:date** **XP3.1** **XQ3.1**
Añade una duración en años a una fecha. El segundo parámetro es el número de años que se debe añadir a la fecha dada como primer argumento. El resultado es de tipo xs:date.

Ejemplos

- **altova:add-years-to-date**(xs:date("2014-01-15"), 10) devuelve 2024-01-15
- **altova:add-years-to-date**(xs:date("2014-01-15"), -4) devuelve 2010-01-15

add-months-to-date [altova:]

altova:add-months-to-date(Fecha *as xs:date*, **Meses** *as xs:integer***)** como **xs:date** **XP3.1** **XQ3.1**
Añade una duración en meses a una fecha. El segundo argumento es el número de meses que se debe añadir a la fecha dada como primer argumento. El resultado es de tipo xs:date.

Ejemplos

- **altova:add-months-to-date**(xs:date("2014-01-15"), 10) devuelve 2014-11-15

- **altova:add-months-to-date**(xs:date("2014-01-15"), -2) devuelve 2013-11-15

▼ add-days-to-date [altova:]

altova:add-days-to-date(Fecha *as xs:date*, **Días** *as xs:integer*) como **xs:date** XP3.1 XQ3.1
Añade una duración en días a una fecha. El segundo argumento es el número de días que se deben añadir a la fecha dad como primer argumento. El resultado es de tipo xs:date.

▣ *Ejemplos*

- **altova:add-days-to-date**(xs:date("2014-01-15"), 10) devuelve 2014-01-25
- **altova:add-days-to-date**(xs:date("2014-01-15"), -8) devuelve 2014-01-07

[Subir 2125]

Recuperar duraciones y aplicarles formato XP3.1 XQ3.1

Estas funciones analizan la entrada **xs:duration** o **xs:string** y devuelven, respectivamente, **xs:string** o **xs:duration**.

▼ format-duration [altova:]

altova:format-duration(Duración *como xs:duration*, **Imagen** *como xs:string*) como **xs:string** XP3.1 XQ3.1
Aplica formato a una duración, que se suministra como primer argumento, en base a la cadena de imagen dada como segundo argumento. El resultado es una cadena de texto cuyo formato se ajusta a la cadena de imagen.

▣ *Ejemplos*

- **altova:format-duration**(xs:duration("P2DT2H53M11.7S"), "Días:[D01] Horas:[H01] Minutos:[m01] Segundos:[s01] Fracciones:[f0]") devuelve "Días:02 Horas:02 Minutos:53 Segundos:11 Fracciones:7"
- **altova:format-duration**(xs:duration("P3M2DT2H53M11.7S"), "Meses:[M01] Días:[D01] Horas:[H01] Minutos:[m01]") devuelve "Meses:03 Días:02 Horas:02 Minutos:53"

▼ parse-duration [altova:]

altova:parse-duration(CadenaEntrada *como xs:string*, **Imagen** *como xs:string*) como **xs:duration** XP3.1 XQ3.1
Toma una cadena con patrón como primer argumento y una cadena de imagen como segundo argumento. La cadena de entrada se analiza en base a la cadena de imagen y se devuelve un xs:duration.

▣ *Ejemplos*

- **altova:parse-duration**("Días:02 Horas:02 Minutos:53 Segundos:11 Fracciones:7", "Días:[D01] Horas:[H01] Minutes:[m01] Segundos:[s01] Fracciones:[f0]") devuelve "P2DT2H53M11.7S"
- **altova:parse-duration**("Meses:03 Días:02 Horas:02 Minutos:53 Segundos:11 Fracciones:7", "Meses:[M01] Días:[D01] Horas:[H01] Minutos:[m01]") devuelve "P3M2DT2H53M"

© 2016-2022 Altova GmbH

[Subir 2125]

Agregar una duración a xs:time XP3.1 XQ3.1

Estas funciones agregan una duración a **xs:time** y devuelven **xs:time**. El tipo xs:time tiene un formato léxico de este tipo hh:mm:ss.sss. Si quiere, puede añadir un sufijo de zona horaria. La letra Z indica (UTC). Las demás zonas horarias se representan con la diferencia que hay entre ellas y la zona UTC: +hh:mm o -hh:mm. Si falta el valor de zona horaria, se entiende que se desconoce (no se da por hecho que es UTC)

add-hours-to-time [altova:]

altova:add-hours-to-time(Hora *as xs:time*, **Horas** *as xs:integer*) como **xs:time** XP3.1 XQ3.1
Añade una duración en horas a una hora. El segundo argumento es el número de horas que se debe añadir a la hora dada como primer argumento. El resultado es de tipo xs:time.

Ejemplos

- **altova:add-hours-to-time**(xs:time("11:00:00"), 10) devuelve 21:00:00
- **altova:add-hours-to-time**(xs:time("11:00:00"), -7) devuelve 04:00:00

add-minutes-to-time [altova:]

altova:add-minutes-to-time(Hora *as xs:time*, **Minutos** *as xs:integer*) como **xs:time** XP3.1 XQ3.1
Añade una duración en minutos a una hora. El segundo argumento es el número de minutos que se debe añadir a la hora dada como primer argumento. El resultado es de tipo xs:time.

Ejemplos

- **altova:add-minutes-to-time**(xs:time("14:10:00"), 45) devuelve 14:55:00
- **altova:add-minutes-to-time**(xs:time("14:10:00"), -5) devuelve 14:05:00

add-seconds-to-time [altova:]

altova:add-seconds-to-time(Hora *as xs:time*, **Segundos** *as xs:integer*) como **xs:time** XP3.1 XQ3.1
Añade una duración en segundos a una hora. El segundo argumento es el número de segundos que se debe añadir a la hora dada como primer argumento. El resultado es de tipo xs:time. El componente **Segundos** puede estar comprendido entre 0 y 59.999.

Ejemplos

- **altova:add-seconds-to-time**(xs:time("14:00:00"), 20) devuelve 14:00:20
- **altova:add-seconds-to-time**(xs:time("14:00:00"), 20.895) devuelve 14:00:20.895

[Subir 2125]

Quitar la parte de zona horaria de los tipos de datos date/time XP3.1 XQ3.1

Estas funciones quitan la zona horaria de los valores **xs:dateTime**, **xs:date** o **xs:time** actuales. Tenga en cuenta que la diferencia entre `xs:dateTime` y `xs:dateTimeStamp` es que en esta última la parte de zona horaria es obligatoria (mientras que en la primera es opcional). Es decir, el formato de un valor `xs:dateTimeStamp` puede ser `SSAA-MM-DDThh:mm:ss.sss±hh:mm` o `SSAA-MM-DDThh:mm:ss.sssZ`. Si la fecha y la hora se leen del reloj del sistema como `xs:dateTimeStamp`, la función `current-dateTime-no-TZ()` se puede usar para quitar la zona horaria.

current-date-no-TZ [altova:]

altova:current-date-no-TZ() como **xs:date** XP3.1 XQ3.1
Esta función no toma ningún argumento. Quita la parte de zona horaria de la función `current-date()` (que es la fecha actual según el reloj del sistema) y devuelve un valor de tipo `xs:date`.

Ejemplos

Si la fecha actual es **2014-01-15+01:00**:

- **altova:current-date-no-TZ()** devuelve `2014-01-15`

current-dateTime-no-TZ [altova:]

altova:current-dateTime-no-TZ() como **xs:dateTime** XP3.1 XQ3.1
Esta función no toma ningún argumento. Quita la parte de zona horaria de `current-dateTime()` (que es la fecha y hora actual según el reloj del sistema) y devuelve un valor de tipo `xs:dateTime`.

Ejemplos

Si la fecha y hora actual es **2014-01-15T14:00:00+01:00**:

- **altova:current-dateTime-no-TZ()** devuelve `2014-01-15T14:00:00`

current-time-no-TZ [altova:]

altova:current-time-no-TZ() **as** **xs:time** XP3.1 XQ3.1
Esta función no toma ningún argumento. Quita la parte de zona horaria de `current-time()` (que es la hora actual según el reloj del sistema) y devuelve un valor de tipo `xs:time`.

Ejemplos

Si la hora actual es **14:00:00+01:00**:

- **altova:current-time-no-TZ()** devuelve `14:00:00`

date-no-TZ [altova:]

altova:date-no-TZ(InputDate *as xs:date*) como **xs:date** XP3.1 XQ3.1
Esta función toma un argumento `xs:date`, del que elimina la parte `timezone` y devuelve un valor `xs:date`. Observe que la fecha permanece intacta.

Ejemplos

© 2016-2022 Altova GmbH

- **altova:date-no-TZ**(xs:date("2014-01-15+01:00")) devuelve 2014-01-15

▼ dateTime-no-TZ [altova:]

altova:dateTime-no-TZ(InputDateTime *as xs:dateTime*) como **xs:dateTime** XP3.1 XQ3.1
Esta función toma un argumento xs:dateTime, del que elimina la parte timezone, y devuelve un valor xs:dateTime. Observe que tanto la fecha como la hora permanecen intactas.

⊞ *Ejemplos*

- **altova:dateTime-no-TZ**(xs:date("2014-01-15T14:00:00+01:00")) devuelve 2014-01-15T14:00:00

▼ time-no-TZ [altova:]

altova:time-no-TZ(HoraEntrada *como xs:time*) como **xs:time** XP3.1 XQ3.1
Esta función toma un argumento xs:time, quita la parte de la zona horaria y devuelve un valor xs:time. Tenga en cuenta que la hora no se modifica.

⊟ *Ejemplos*

- **altova:time-no-TZ**(xs:time("14:00:00+01:00")) devuelve 14:00:00

[Subir 2125]

Devolver el número de días, horas, minutos y segundos de duraciones XP3.1 XQ3.1

Estas funciones devuelven el número de días en un mes y el número de horas, minutos y segundos de las duraciones correspondientes.

▼ days-in-month [altova:]

altova:days-in-month(Year *as xs:integer*, **Month** *as xs:integer*) como **xs:integer** XP3.1 XQ3.1
Devuelve el número de días en el mes indicado. El mes se indica con los argumentos Year y Month.

⊞ *Ejemplos*

- **altova:days-in-month**(2018, 10) devuelve 31
- **altova:days-in-month**(2018, 2) devuelve 28
- **altova:days-in-month**(2020, 2) devuelve 29

▼ hours-from-dayTimeDuration-accumulated

altova:hours-from-dayTimeDuration-accumulated(DayAndTime *como xs:duration*) como **xs:integer** XP3.1 XQ3.1
Devuelve el número total de horas de la duración enviada por el argumento DayAndTime (que es de tipo xs:duration). Las horas de los componentes Day y Time se agregan juntos para dar como resultado un número entero. Una hora nueva son 60 minutos enteros. Las duraciones negativas dan como resultado un

valor de hora negativo.

Ejemplos

- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5D") devuelve 120, que es el número total de horas en 5 días.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5DT2H") devuelve 122, que es el número total de horas en 5 días más 2 horas.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5DT2H60M") devuelve 123, que es el número total de horas en 5 días más 2 horas y 60 mins.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5DT2H119M") devuelve 123, que es el número total de horas en 5 días más 2 horas y 119 mins.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5DT2H120M") devuelve 124, que es el número total de horas en 5 días más 2 horas y 120 mins.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("-P5DT2H") devuelve -122

minutes-from-dayTimeDuration-accumulated

altova:minutes-from-dayTimeDuration-accumulated(DayAndTime *como xs:duration*) como **xs:integer** **XP3.1** **XQ3.1**
Devuelve el número total de minutos de la duración enviada por el argumento DayAndTime (que es de tipo xs:duration). Los minutos de los componentes Day y Time se agregan juntos para dar como resultado un número entero. Las duraciones negativas dan como resultado un valor de minutos negativo.

Ejemplos

- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("PT60M") devuelve 60
- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("PT1H") devuelve 60, que es el número total de minutos en 1 hora.
- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("PT1H40M") devuelve 100
- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("P1D") devuelve 1440, que es el número total de minutos en 1 día.
- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("-P1DT60M") devuelve -1500

seconds-from-dayTimeDuration-accumulated

altova:seconds-from-dayTimeDuration-accumulated(DayAndTime *como xs:duration*) como **xs:integer** **XP3.1** **XQ3.1**
Devuelve el número total de segundos de la duración enviada por el argumento DayAndTime (que es de tipo xs:duration). Los segundos de los componentes Day y Time se agregan juntos para dar como resultado un número entero. Las duraciones negativas dan como resultado un valor de segundos negativo.

Ejemplos

- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("PT1M") devuelve 60, que es el número total de segundos en 1 minuto.
- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("PT1H") devuelve 3600, que es el número total de segundos en 1 hora.
- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("PT1H2M") devuelve 3720

© 2016-2022 Altova GmbH

- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("P1D") devuelve 86400, que es el número total de segundos en 1 día.
- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("-P1DT1M") devuelve -86460

Obtener el día de la semana de **xs:dateTime** o **xs:date** XP3.1 XQ3.1

Estas funciones obtienen el día de la semana (como entero) de xs:dateTime o xs:date. Los días de la semana se numeran del 1 al 7 (usando el formato EE UU, es decir Domingo =1). En el formato europeo la semana empieza el lunes (es decir, Lunes=1). Para establecer el formato EE UU (Domingo=1) use el entero 0 allí donde se acepte un entero para indicar el formato.

weekday-from-dateTime [altova:]

altova:weekday-from-dateTime(DateTime *como xs:dateTime*) **como xs:integer** XP3.1 XQ3.1
Toma una fecha como único argumento y devuelve el día de la semana de la fecha dada como número entero. Los días de la semana se numeran del 1 al 7 empezando por Domingo=1. Si necesita usar el formato europeo (donde Lunes=1), utilice la otra firma de esta función (*ver más abajo*).

Ejemplos

- **altova:weekday-from-dateTime**(xs:dateTime("2014-02-03T09:00:00")) devuelve 2, lo cual significa "Lunes"..

altova:weekday-from-dateTime(DateTime *como xs:dateTime*, **Formato** *como xs:integer*) **como xs:integer** XP3.1 XQ3.1
Toma una fecha como primer argumento y devuelve el día de la semana de la fecha dada como número entero. Si el segundo argumento (número entero) es 0, entonces los días de la semana se numeran del 1 al 7 empezando por Domingo=1. Si el segundo argumento es un entero distinto de 0, entonces Lunes=1. Si falta el segundo argumento, la función se lee como en la firma anterior (*ver más arriba*).

Ejemplos

- **altova:weekday-from-dateTime**(xs:dateTime("2014-02-03T09:00:00"), 1) devuelve 1, lo cual significa "Lunes"
- **altova:weekday-from-dateTime**(xs:dateTime("2014-02-03T09:00:00"), 4) devuelve 1, lo cual significa "Lunes"
- **altova:weekday-from-dateTime**(xs:dateTime("2014-02-03T09:00:00"), 0) devuelve 2, lo cual significa "Lunes"

weekday-from-date [altova:]

altova:weekday-from-date(Date *como xs:date*) **como xs:integer** XP3.1 XQ3.1
Toma una fecha como único argumento y devuelve el día de la semana de la fecha dada como número entero. Los días de la semana se numeran del 1 al 7 empezando por Domingo=1. Si necesita usar el formato europeo (donde Lunes=1), utilice la otra firma de esta función (*ver más abajo*).

Ejemplos

- **altova:weekday-from-date**(xs:date("2014-02-03+01:00")) devuelve 2, lo cual significa lunes.

altova:weekday-from-date(Date *como xs:date*, **Formato** *como xs:integer*) **como xs:integer**
XP3.1 XQ3.1
Toma una fecha como primer argumento y devuelve el día de la semana de la fecha dada como número entero. Si el segundo argumento (`Formato`) es `0`, entonces los días de la semana se numeran del `1` al `7` empezando por `Domingo=1`. Si el segundo argumento es un entero distinto de `0`, entonces `Lunes=1`. Si falta el segundo argumento, la función se lee como en la firma anterior (*ver más arriba*).

Ejemplos

- `altova:weekday-from-date(xs:date("2014-02-03"), 1)` devuelve `1`, lo cual significa "Lunes"
- `altova:weekday-from-date(xs:date("2014-02-03"), 4)` devuelve `1`, lo cual significa "Lunes"
- `altova:weekday-from-date(xs:date("2014-02-03"), 0)` devuelve `2`, lo cual significa "Lunes".

[Subir 2125]

Devolver el número de semana de xs:dateTime o xs:date XP2 XQ1 XP3.1 XQ3.1

Estas funciones devuelven el número de semana (como número entero) de `xs:dateTime` o `xs:date`. El número de la semana está disponible en el formato de calendario estadounidense, europeo e islámico. La razón de que los números de semana difieran en cada uno de estos calendarios es que en cada uno de ellos se considera un día diferente para el inicio de la semana (p. ej. en el formato estadounidense el primer día de la semana es el domingo).

weeknumber-from-date [altova:]

altova:weeknumber-from-date(Fecha *como xs:date*, **Calendario** *como xs:integer*) como **xs:integer** XP2 XQ1 XP3.1 XQ3.1
Devuelve como número entero el número de semana del argumento **Fecha** dado. El segundo argumento (**Calendario**) indica el sistema de calendario que se debe seguir.
Estos son los valores permitidos para el argumento **Calendario**:

- **0 = Calendario estadounidense** (la semana comienza el domingo)
- **1 = Calendario estándar ISO o europeo** (la semana comienza el lunes)
- **2 = Calendario islámico** (la semana comienza el sábado)

El valor predeterminado es `0`.

Ejemplos

- `altova:weeknumber-from-date(xs:date("2014-03-23"), 0)` devuelve `13`
- `altova:weeknumber-from-date(xs:date("2014-03-23"), 1)` devuelve `12`
- `altova:weeknumber-from-date(xs:date("2014-03-23"), 2)` devuelve `13`
- `altova:weeknumber-from-date(xs:date("2014-03-23") )` devuelve `13`

El día de la fecha de los ejemplos anteriores (`2014-03-23`) es un domingo. Por tanto, en este caso, el calendario estadounidense y el islámico van una semana por delante del calendario europeo.

weeknumber-from-dateTime [altova:]

altova:weeknumber-from-dateTime(FechaHora *como xs:dateTime*, **Calendario** *como xs:integer*)

© 2016-2022 Altova GmbH

como `xs:integer` XP2 XQ1 XP3.1 XQ3.1
Devuelve como entero el día de la semana del argumento **FechaHora** dado. El segundo argumento (**Calendario**) indica el sistema de calendario que se debe seguir.
Estos son los valores permitidos para el argumento **Calendario**:

- **0 = Calendario estadounidense** (la semana comienza el domingo)
- **1 = Calendario estándar ISO o europeo** (la semana comienza el lunes)
- **2 = Calendario islámico** (la semana comienza el sábado)

El valor predeterminado es `0`.

Ejemplos

- **altova:weeknumber-from-dateTime**(xs:dateTime("2014-03-23T00:00:00"), 0) devuelve 13
- **altova:weeknumber-from-dateTime**(xs:dateTime("2014-03-23T00:00:00"), 1) devuelve 12
- **altova:weeknumber-from-dateTime**(xs:dateTime("2014-03-23T00:00:00"), 2) devuelve 13
- **altova:weeknumber-from-dateTime**(xs:dateTime("2014-03-23T00:00:00")) devuelve 13

El día de `dateTime` de los ejemplos anteriores (`2014-03-23T00:00:00`) es un domingo. Por tanto, en este caso, el calendario estadounidense y el islámico van una semana por delante del calendario europeo.

[Subir 2125]

Generar tipos de datos de fecha, hora y duración a partir de sus componentes léxicos XP3.1 XQ3.1

Estas funciones toman los componentes léxicos de los tipos de datos `xs:date`, `xs:time` y `xs:duration` como argumentos de entrada y los combinan para generar el tipo de datos correspondiente.

build-date [altova:]

altova:build-date(Año *as xs:integer*, **Mes** *as xs:integer*, **Fecha** *as xs:integer***)** como **xs:date** XP3.1 XQ3.1
Los argumentos son el año, el mes y la fecha respectivamente. Se combinan para generar un valor de tipo `xs:date`. Los valores de los enteros deben estar en el intervalo de esa fecha en particular. Por ejemplo, el segundo argumento (para el mes) no puede ser mayor que `12`.

Ejemplos

- **altova:build-date**(2014, 2, 03) devuelve 2014-02-03

build-time [altova:]

altova:build-time(Horas *as xs:integer*, **Minutos** *as xs:integer*, **Segundos** *as xs:integer***)** **como xs:time** XP3.1 XQ3.1
El primer, segundo y tercer argumentos son la hora (`0` - `23`), los minutos (`0` - `59`) y los segundos (`0` - `59`) respectivamente. Se combinan para generar un valor de tipo `xs:time`. Los valores de los enteros deben estar dentro del intervalo correcto de esa parte de tiempo concreta. Por ejemplo, el segundo argumento (`Minutos`) no puede ser mayor que `59`. Para añadir la parte de uso horario al valor, use la firma que aparece más abajo.

Ejemplos

- **altova:build-time**(23, 4, 57) devuelve 23:04:57

altova:build-time(Horas *como xs:integer*, **Minutos** *como xs:integer*, **Segundos** *as xs:integer*, **TimeZone** *como xs:string*) **como xs:time** XP3.1 XQ3.1

El primer, segundo y tercer argumentos son la hora (0 - 23), los minutos (0 - 59) y los segundos (0 - 59) respectivamente. El cuarto argumento es una cadena de texto que indica la parte del valor de la zona horaria. Este cuarto argumento se combina para generar un valor de tipo xs:time. Los valores de los enteros deben estar dentro del intervalo correcto de esa parte de tiempo concreta. Por ejemplo, el segundo argumento (Minutos) no puede ser mayor que 59.

Ejemplos

- **altova:build-time**(23, 4, 57, '+1') devuelve 23:04:57+01:00

build-duration [altova:]

altova:build-duration(Años *as xs:integer*, **Meses** *as xs:integer*) como **xs:yearMonthDuration** XP3.1 XQ3.1

Toma dos argumentos para generar un valor de tipo xs:yearMonthDuration. El primer argumento da la parte Years del valor de duración, mientras que el segundo da la parte Months. Si el segundo (Months) es mayor o igual que 12, el entero se divide por 12. El cociente se añade al primer argumento para aportar la parte Years del valor de duración, mientras que el resto (de la división) da la parte Months. Para generar una duración de tipo xs:dayTimeDuration, consulte la firma siguiente.

Ejemplos

- **altova:build-duration**(2, 10) devuelve P2Y10M
- **altova:build-duration**(14, 27) devuelve P16Y3M
- **altova:build-duration**(2, 24) devuelve P4Y

altova:build-duration(Días *as xs:integer*, **Horas** *as xs:integer*, **Minutos** *as xs:integer*, **Segundos** *as xs:integer*) como **xs:dayTimeDuration** XP3.1 XQ3.1

Toma cuatro argumentos y los combina para generar un valor de tipo xs:dayTimeDuration. El primer argumento da la parte Days del valor de duración, el segundo, el tercero y el cuarto dan las partes Hours, Minutes y Seconds respectivamente. Los tres argumentos de tiempo se convierten a un valor equivalente en cuanto a la unidad mayor siguiente y el resultado se utiliza para calcular el valor total de la duración. Por ejemplo, 72 segundos se convierte en 1M+12S (1 minuto y 12 segundos) y este valor se usa para calcular el valor total de la duración. Para generar una duración de tipo xs:yearMonthDuration, consulte la firma anterior.

Ejemplos

- **altova:build-duration**(2, 10, 3, 56) devuelve P2DT10H3M56S
- **altova:build-duration**(1, 0, 100, 0) devuelve P1DT1H40M
- **altova:build-duration**(1, 0, 0, 3600) devuelve P1DT1H

[Subir 2125]

© 2016-2022 Altova GmbH

Construir tipos de datos date, dateTime y time a partir de una cadena de entrada XP2 XQ1 XP3.1 XQ3.1

Estas funciones toman cadenas como argumentos y construyen tipos de datos `xs:date`, `xs:dateTime` o `xs:time`. La cadena de entrada se analiza para los componentes del tipo de datos en función del argumento patrón dado.

parse-date [altova:]

altova:parse-date(Fecha *como xs:string*, **PatrónFecha** *como xs:string*) como **xs:date** XP2 XQ1 XP3.1 XQ3.1

Devuelve la cadena de entrada **Fecha** como valor **xs:date**. El segundo argumento (**PatrónFecha**) indica el patrón (secuencia de componentes) de la cadena de entrada. El argumento **PatrónFecha** se describe con los especificadores que aparecen a continuación y con cualquier separador de componentes (consulte los ejemplos más abajo).

D	Día
M	Mes
Y	Año

El patrón **PatrónFecha** debe coincidir con el patrón de **Fecha**. Como el resultado es de tipo **xs:date**, el resultado siempre tendrá el formato léxico **YYYY-MM-DD**.

Ejemplos

- **altova:parse-date**(xs:string("09-12-2014"), "[D]-[M]-[Y]") devuelve 2014-12-09
- **altova:parse-date**(xs:string("09-12-2014"), "[M]-[D]-[Y]") devuelve 2014-09-12
- **altova:parse-date**("06/03/2014", "[M]/[D]/[Y]") devuelve 2014-06-03
- **altova:parse-date**("06 03 2014", "[M] [D] [Y]") devuelve 2014-06-03
- **altova:parse-date**("6 3 2014", "[M] [D] [Y]") devuelve 2014-06-03

parse-dateTime [altova:]

altova:parse-dateTime(FechaHora *como xs:string*, **PatrónFechaHora** *como xs:string*) como **xs:dateTime** XP2 XQ1 XP3.1 XQ3.1

Devuelve la cadena de entrada **FechaHora** como valor **xs:dateTime**. El segundo argumento (**PatrónFechaHora**) indica el patrón (secuencia de componentes) de la cadena de entrada. El argumento **PatrónFechaHora** se describe con los especificadores que aparecen a continuación y con cualquier separador de componentes (consulte los ejemplos más abajo).

D	Día
M	Mes
Y	Año
H	Hora
m	minutos
s	segundos

El patrón **PatrónFechaHora** debe coincidir con el patrón de **FechaHora**. Como el resultado es de tipo **xs:dateTime**, el resultado siempre tendrá el formato léxico **YYYY-MM-DDTHH:mm:ss**.

Ejemplos

- `altova:parse-dateTime(xs:string("09-12-2014 13:56:24"), "[M]-[D]-[Y] [H]:[m]:[s]")` devuelve `2014-09-12T13:56:24`
- `altova:parse-dateTime("time=13:56:24; date=09-12-2014", "time=[H]:[m]:[s]; date=[D]-[M]-[Y]")` devuelve `2014-12-09T13:56:24`

parse-time [altova:]

altova:parse-time(Hora *como xs:string*, **PatrónHora** *como xs:string*) como **xs:time** XP2 XQ1 XP3.1 XQ3.1

Devuelve la cadena de entrada **Hora** como valor **xs:time**. El segundo argumento (**PatrónHora**) indica el patrón (secuencia de componentes) de la cadena de entrada. El argumento **PatrónHora** se describe con los especificadores que aparecen a continuación y con cualquier separador de componentes (consulte los ejemplos más abajo).

H	Hora
m	minutos
s	segundos

El patrón **PatrónHora** debe coincidir con el patrón de **Hora**. Como el resultado es de tipo **xs:time**, el resultado siempre tendrá el formato léxico **HH:mm:ss**.

Ejemplos

- `altova:parse-time(xs:string("13:56:24"), "[H]:[m]:[s]")` devuelve `13:56:24`
- `altova:parse-time("13-56-24", "[H]-[m]")` devuelve `13:56:00`
- `altova:parse-time("time=13h56m24s", "time=[H]h[m]m[s]s")` devuelve `13:56:24`
- `altova:parse-time("time=24s56m13h", "time=[s]s[m]m[H]h")` devuelve `13:56:24`

[Subir 2125]

Funciones para calcular la edad XP3.1 XQ3.1

Estas funciones devuelven la edad que se calcula obteniendo la diferencia (i) entre la fecha del argumento de entrada y la fecha actual o (ii) entre las fechas de los dos argumentos de entrada. La función **age** devuelve la edad en años, mientras que la función **age-details** devuelve la edad en forma de una secuencia de tres enteros (años, meses y días).

age [altova:]

altova:age(FechaInicio *as xs:date*) como **xs:integer** XP3.1 XQ3.1

Devuelve un entero que es la edad *en años* de algún objeto, contando a partir de la fecha de inicio dada como argumento y hasta la fecha actual (tomada del reloj del sistema). Si el argumento de entrada es un año o más después que la fecha actual, el valor devuelto será negativo.

Ejemplos

© 2016-2022 Altova GmbH

Si la fecha actual es **2014-01-15**:

- **altova:age**(xs:date("2013-01-15")) devuelve 1
- **altova:age**(xs:date("2013-01-16")) devuelve 0
- **altova:age**(xs:date("2015-01-15")) devuelve -1
- **altova:age**(xs:date("2015-01-14")) devuelve 0

altova:age(FechaInicio *as xs:date*, **FechaFinal** *as xs:date*) como **xs:integer** **XP3.1** **XQ3.1**
Devuelve un entero que es la edad *en años* de algún objeto, contando a partir de la fecha de inicio dada como primer argumento y hasta la fecha dada como segundo argumento. El valor devuelto será negativo si el primer argumento es un año o más después que el segundo argumento.

Ejemplos

- **altova:age**(xs:date("2000-01-15"), xs:date("2010-01-15")) devuelve 10
- **altova:age**(xs:date("2000-01-15"), current-date()) devuelve 14 si la fecha actual es 2014-01-15
- **altova:age**(xs:date("2014-01-15"), xs:date("2010-01-15")) devuelve -4

age-details [altova:]

altova:age-details(FechaEntrada *as xs:date*) como (**xs:integer**)* **XP3.1** **XQ3.1**
Devuelve tres enteros que son los años, meses y días respectivamente que hay entre la fecha dada como argumento y la fecha actual (tomada del reloj del sistema). La suma del valor devuelto nos da el tiempo total transcurrido entre ambas fechas (entre la fecha dada y la fecha actual). La fecha de entrada puede tener un valor anterior o posterior a la fecha actual, pero esto no se indica en el valor devuelto por medio de un signo negativo o positivo. El valor devuelto siempre es positivo.

Ejemplos

Si la fecha actual es **2014-01-15**:

- **altova:age-details**(xs:date("2014-01-16")) devuelve (0 0 1)
- **altova:age-details**(xs:date("2014-01-14")) devuelve (0 0 1)
- **altova:age-details**(xs:date("2013-01-16")) devuelve (1 0 1)
- **altova:age-details**(current-date()) devuelve (0 0 0

altova:age-details(Fecha1 *as xs:date*, **Fecha2** *as xs:date*) como (**xs:integer**)* **XP3.1** **XQ3.1**
Devuelve tres enteros que son los años, meses y días que hay entre las dos fechas dadas por los argumentos. La suma del valor devuelto nos da el tiempo total transcurrido entre las dos fechas de entrada. Da igual cuál de las dos fechas se da como primer argumento, la más antigua o la más reciente. El valor devuelto no indica si la fecha de entrada es anterior o posterior a la fecha actual. Es decir, el valor devuelto siempre es positivo.

Ejemplos

- **altova:age-details**(xs:date("2014-01-16"), xs:date("2014-01-15")) devuelve (0 0 1)
- **altova:age-details**(xs:date("2014-01-15"), xs:date("2014-01-16")) devuelve (0 0 1)

[Subir 2125]

Funciones para calcular el tiempo Unix XP3.1 XQ3.1

El tiempo Unix es una medida de tiempo que se usa en sistemas Unix. Se define como la cantidad de segundos transcurridos desde las 00:00:00 UTC del 1 de enero de 1970. Estas funciones convierten valores **xs:dateTime** en tiempo Unix y viceversa.

dateTime-from-epoch [altova:]

altova:dateTime-from-epoch(Epoch *como xs:decimal* **como xs:dateTime** XP3.1 XQ3.1
El tiempo Unix es una medida de tiempo que se usa en sistemas Unix. Se define como la cantidad de segundos transcurridos desde las 00:00:00 UTC del 1 de enero de 1970. La función **dateTime-from-epoch** devuelve el equivalente en **xs:dateTime** de un instante de tiempo Unix, lo ajusta a la zona horaria local e incluye la información de esa zona horaria en el resultado.

La función toma un argumento `xs:decimal` y devuelve un valor `xs:dateTime` que incluye una parte **TZ**, que indica la zona horaria. Para obtener el resultado se calcula el equivalente en **dateTime** UTC del instante de tiempo Unix y se añade a la zona horaria local (que se obtiene del reloj del sistema). Por ejemplo, si la función se ejecuta en un equipo cuya configuración sitúa en una zona horaria de +01:00 (con respecto a UTC), una vez se ha calculado el equivalente en **dateTime** se le añade una hora al resultado. La información de la zona horaria, que es una parte léxica opcional del resultado de **xs:dateTime**, también se incluye en el resultado **dateTime**. Compare este resultado con el de **dateTime-from-epoch-no-TZ** y consulte la función **epoch-from-dateTime**.

Ejemplos

La zona horaria local de los ejemplos siguientes es UTC +01:00. En consecuencia, el equivalente en **dateTime** UTC del instante de tiempo Unix indicado aumentará en una hora. La zona horaria se indica en el resultado.

- **altova:dateTime-from-epoch**(34) devuelve `1970-01-01T01:00:34+01:00`
- **altova:dateTime-from-epoch**(62) devuelve `1970-01-01T01:01:02+01:00`

dateTime-from-epoch-no-TZ [altova:]

altova:dateTime-from-epoch-no-TZ(Epoch *como xs:decimal* **como xs:dateTime** XP3.1 XQ3.1
El tiempo Unix es una medida de tiempo que se usa en sistemas Unix. Se define como la cantidad de segundos transcurridos desde las 00:00:00 UTC del 1 de enero de 1970. La función **dateTime-from-epoch-no-TZ** devuelve el equivalente en **xs:dateTime** de un instante de tiempo Unix y lo ajusta a la zona horaria local pero no incluye la información de esa zona horaria en el resultado.

La función toma un argumento `xs:decimal` y devuelve un valor `xs:dateTime` que no incluye la parte **TZ**, que indica la zona horaria. Para obtener el resultado se calcula el equivalente en **dateTime** UTC del instante de tiempo Unix y se añade a la zona horaria local (que se obtiene del reloj del sistema). Por ejemplo, si la función se ejecuta en un equipo cuya configuración sitúa en una zona horaria de +01:00 (con respecto a UTC), una vez se ha calculado el equivalente en **dateTime** se le añade una hora al resultado. La información de la zona horaria, que es una parte léxica opcional del resultado de **xs:dateTime**, no se incluye en el resultado **dateTime**. Compare este resultado con el de **dateTime-from-epoch** y consulte la función **epoch-from-dateTime**.

© 2016-2022 Altova GmbH

Ejemplos

La zona horaria local de los ejemplos siguientes es UTC +01:00. En consecuencia, el equivalente en **dateTime** UTC del instante de tiempo Unix indicado aumentará en una hora. La zona horaria no se indica en el resultado.

- **altova:dateTime-from-epoch**(34) devuelve 1970-01-01T01:00:34
- **altova:dateTime-from-epoch**(62) devuelve 1970-01-01T01:01:02

epoch-from-dateTime [altova:]

altova:epoch-from-dateTime(dateTimeValue *como xs:dateTime*) **como xs:decimal** **XP3.1** **XQ3.1**
El tiempo Unix es una medida de tiempo que se usa en sistemas Unix. Se define como la cantidad de segundos transcurridos desde las 00:00:00 UTC del 1 de enero de 1970. La función **epoch-from-dateTime** devuelve el equivalente en tiempo Unix del valor **xs:dateTime** que se indica en el argumento de la función. Tenga en cuenta que puede que deba generar de forma explícita el valor **xs:dateTime**. El valor **xs:dateTime** puede o no contener la parte opcional **TZ**, que indica la zona horaria.

Tanto si se indica la parte de la zona horaria como parte del argumento como si no, la diferencia que esta indica se obtiene del reloj del sistema y se resta al argumento **dateTimeValue** indicado. El resultado es el tiempo UTC a partir del cual se calcula el equivalente en tiempo Unix. Por ejemplo, si la función se ejecuta en un equipo cuya configuración sitúa en una zona horaria de +01:00 (con respecto a UTC), se resta una hora al valor **dateTimeValue** indicado antes de calcular el valor en tiempo Unix. Consulte también la función **dateTime-from-epoch**.

Ejemplos

La zona horaria local de los ejemplos siguientes es UTC +01:00. En consecuencia, se le restará una hora al valor **dateTime** indicado antes de calcular el tiempo Unix.

- **altova:epoch-from-dateTime**(xs:dateTime("1970-01-01T01:00:34+01:00")) devuelve 34
- **altova:epoch-from-dateTime**(xs:dateTime("1970-01-01T01:00:34")) devuelve 34
- **altova:epoch-from-dateTime**(xs:dateTime("2021-04-01T11:22:33")) devuelve 1617272553

[Subir 2125]

30.2.1.3 Funciones XPath/XQuery: de geoubicación

Las funciones de extensión XPath/XQuery de geoubicación son compatibles con la versión actual de XMLSpy y se pueden utilizar en (i) expresiones XPath en contextos XSLT o (ii) expresiones XQuery en documentos XQuery.

Nota sobre el nombre de las funciones y lenguajes

Puede utilizar todas las funciones de extensión de Altova en sus expresiones XPath/XQuery. Con ellas conseguirá funciones adicionales no disponibles en la biblioteca de funciones estándar de XPath, XQuery y XSLT. Las funciones de extensión de Altova están en el **espacio de nombres** **http://www.altova.com/xslt-extensions** y en esta sección se presentan con el prefijo, que se supone

estará enlazado al espacio de nombres señalado. Tenga en cuenta que en futuras versiones del producto algunas funciones pueden dejar de ser compatibles o su comportamiento puede cambiar. Por tanto, consulte siempre la documentación del producto para conocer el funcionamiento de estas funciones en cada versión del producto.

Funciones XPath (en expresiones XPath en XSLT):	XP1 XP2 XP3.1.1
Funciones XSLT (en expresiones XPath en XSLT):	XSLT1 XSLT2 XSLT3
Funciones XQuery (en expresiones XQuery en XQuery):	XQ1 XQ3.1

format-geolocation [altova:]

altova:format-geolocation(Latitude *como xs:decimal*, **Longitude** *como xs:decimal*, **GeolocationOutputStringFormat** *como xs:integer*) como **xs:string** XP3.1 XQ3.1

Toma la latitud y la longitud como los dos primeros argumentos y da como resultado la geoubicación como cadena. El tercer argumento, **GeolocationOutputStringFormat**, es el formato de la cadena de resultado de la geoubicación: usa valores enteros del 1 al 4 para identificar el formato de la cadena de resultado (consulte más abajo "*Formatos de la cadena de resultado geoubicación").* Los valores de latitud oscilan entre `+90` y `-90 (N a S)`. Los valores de longitud oscilan entre +`180` y `-180 (E a O)`.

Nota: la función `image-exif-data` y el atributo de metadatos Exif se pueden usar para suministrar las cadenas de entrada.

Ejemplos

- **altova:format-geolocation**(33.33, -22.22, 4) devuelve el xs:string "33.33 -22.22"
- **altova:format-geolocation**(33.33, -22.22, 2) devuelve el xs:string "33.33N 22.22W"
- **altova:format-geolocation**(-33.33, 22.22, 2) devuelve el xs:string "33.33S 22.22E"
- **altova:format-geolocation**(33.33, -22.22, 1) devuelve el xs:string "33°19'48.00"S 22° 13'12.00"E"

Formato de las cadenas de salida de las geoubicaciones:

A la latitud y longitud suministradas se les aplica un formato de salida de los que se indican más abajo. El formato deseado se identifica con un identificador comprendido entre `1` y `4`. Los valores de latitud pueden estar comprendidos entre `+90` y `-90` (`N` a `S`). Los valores de longitud pueden estar comprendidos entre `+180` y `-180` (`E` a `W`).

1
Grados, minutos y segundos decimales + orientación como sufijo (`N/S, E/W`) **D°M'S.SS"N/S D°M'S.SS"E/W** *Ejemplo*: **33°55'11.11"N 22°44'66.66"W**

2
Grados decimales + orientación como sufijo (`N/S, E/W`) **D.DDN/S D.DDE/W**

© 2016-2022 Altova GmbH

Ejemplo: `33.33N 22.22W`

3
Grados, minutos y segundos decimales + prefijo (`+/-`). El signo + para (`N/E`) es opcional `+/-D°M'S.SS" +/-D°M'S.SS"` *Ejemplo*: `33°55'11.11" -22°44'66.66"`

4
Grados decimales + prefijo (`+/-`). El signo + para (`N/E`) es opcional `+/-D.DD +/-D.DD` *Ejemplo*: `33.33 -22.22`

- *Atributo Exif de Altova: Geolocation*

  El motor XPath/XQuery de Altova genera el atributo personalizado **`Geolocation`** a partir de las etiquetas de metadatos Exif estándar. Este atributo es una concatenación de cuatro etiquetas Exif (`GPSLatitude`, `GPSLatitudeRef`, `GPSLongitude`, `GPSLongitudeRef`) seguidas de unidades:

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

- parse-geolocation [altova:]

  `altova:parse-geolocation(CadenaEntradaGeoubicación` *como xs:string*`)` como **`xs:decimal+`**
  XP3.1 **XQ3.1**
  Analiza el argumento `CadenaEntradaGeoubicación` y devuelve la latitud y la longitud (en ese orden) de la geoubicación en forma de secuencia de dos elementos `xs:decimal`. Más abajo puede ver en qué formatos se puede suministrar la cadena de entrada de la geoubicación.

  Nota: la función `image-exif-data` 2155 y el atributo `@Geolocation` 2155 de los metadatos Exif se pueden utilizar para suministrar la cadena de entrada de la geoubicación (*ver ejemplos*).

  - *Ejemplos*

    - **`altova:parse-geolocation`**`("33.33 -22.22")` devuelve la secuencia de dos `xs:decimals` `(33.33, 22.22)`
    - **`altova:parse-geolocation`**`("48°51'29.6""N 24°17'40.2""")` devuelve la secuencia de dos `xs:decimals` `(48.8582222222222, 24.2945)`
    - **`altova:parse-geolocation`**`('48°51''29.6"N 24°17''40.2"')` devuelve la secuencia de dos `xs:decimals` `(48.8582222222222, 24.2945)`
    - **`altova:parse-geolocation`**`( image-exif-data(//MisImágenes/Imagen20141130.01)/@Geolocation )` devuelve una secuencia de dos `xs:decimals`

Formato de las cadenas de entrada de geoubicaciones:

La cadena de entrada de la geoubicación debe contener la latitud y la longitud (en ese orden) se paradas por un espacio en blanco. Ambas pueden estar en cualquier formato de los que se indican más abajo y puede combinar formatos distintos. Es decir, la latitud puede estar en un formato y la longitud en otro. Los valores de la latitud deben estar comprendidos entre `+90` y `-90` (`N` a `S`). Los valores de longitud deben estar comprendidos entre `+180` y `-180` (`E` a `W`).

Nota: si utiliza comillas simples o dobles para delimitar el argumento de la cadena de entrada, esto dará lugar a un conflicto con las comillas simples o dobles que se utilizan, respectivamente, para indicar los valores de los minutos y los segundos. Si esto ocurre, debe añadir caracteres de escape a las comillas utilizadas para los minutos y segundos (esto se hace duplicando las comillas.

- Grados, minutos y segundos decimales + orientación como sufijo (`N/S`, `E/W`)
  `D°M'S.SS"N/S D°M'S.SS"W/E`
  Ejemplo: **`33°55'11.11"N 22°44'55.25"W`**

- Grados, minutos y segundos decimales + prefijo (`+/-`). El signo + para (**`N/E`**) es opcional
  `+/-D°M'S.SS" +/-D°M'S.SS"`
  Ejemplo: **`33°55'11.11" -22°44'55.25"`**

- Grados y minutos decimales + orientación como sufijo (`N/S`, `E/W`)
  `D°M.MM'N/S D°M.MM'W/E`
  Ejemplo: **`33°55.55'N 22°44.44'W`**

- Grados y minutos decimales + prefijo (`+/-`). El signo + para (**`N/E`**) es opcional
  `+/-D°M.MM' +/-D°M.MM'`
  Ejemplo: **`+33°55.55' -22°44.44'`**

- Grados decimales + orientación como sufijo (`N/S`, `E/W`)
  `D.DDN/S D.DDW/E`
  Ejemplo: **`33.33N 22.22W`**

- Grados decimales + prefijo (`+/-`). El signo + para (**`N/S, E/W`**) es opcional
  `+/-D.DD +/-D.DD`
  Ejemplo: **`33.33 -22.22`**

Ejemplos de combinación de formatos:

```
33.33N  -22°44'55.25"
33.33  22°44'55.25"W
33.33  22.45
```

Atributo Exif de Altova: Geolocation

El motor XPath/XQuery de Altova genera el atributo personalizado **`Geolocation`** a partir de las etiquetas de metadatos Exif estándar. Este atributo es una concatenación de cuatro etiquetas Exif (`GPSLatitude`, `GPSLatitudeRef`, `GPSLongitude`, `GPSLongitudeRef`) seguidas de unidades:

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

© 2016-2022 Altova GmbH

▼ geolocation-distance-km [altova:]

altova:geolocation-distance-km(CadenaEntradaGeoubicación-1 *como xs:string*, **CadenaEntradaGeoubicación-2** *como xs:string*) como **xs:decimal** **XP3.1** **XQ3.1**
Calcula la distancia en km que existe entre dos geoubicaciones. El formato que puede utilizarse para dar las cadenas de entrada aparece más abajo. Los valores de latitud están comprendidos entre `+90` y `-90` (`N` a `S`). Los valores de longitud están comprendidos entre `+180` y `-180` (`E` a `W`).

Nota: la función image-exif-data[2155] y el atributo de metadatos Exif @Geolocation[2155] pueden utilizarse para suministrar las cadenas de entrada de geoubicaciones.

▣ *Ejemplos*

- **altova:geolocation-distance-km**(`"33.33 -22.22"`, `"48°51'29.6""N 24°17'40.2"""`) devuelve el `xs:decimal 4183.08132372392`

▣ *Formato de las cadenas de entrada de geoubicaciones:*

La cadena de entrada de la geoubicación debe contener la latitud y la longitud (en ese orden) se paradas por un espacio en blanco. Ambas pueden estar en cualquier formato de los que se indican más abajo y puede combinar formatos distintos. Es decir, la latitud puede estar en un formato y la longitud en otro. Los valores de la latitud deben estar comprendidos entre `+90` y `-90` (`N` a `S`). Los valores de longitud deben estar comprendidos entre `+180` y `-180` (`E` a `W`).

Nota: si utiliza comillas simples o dobles para delimitar el argumento de la cadena de entrada, esto dará lugar a un conflicto con las comillas simples o dobles que se utilizan, respectivamente, para indicar los valores de los minutos y los segundos. Si esto ocurre, debe añadir caracteres de escape a las comillas utilizadas para los minutos y segundos (esto se hace duplicando las comillas.

- Grados, minutos y segundos decimales + orientación como sufijo (`N/S`, `E/W`)
  `D°M'S.SS"N/S D°M'S.SS"W/E`
  Ejemplo: **`33°55'11.11"N 22°44'55.25"W`**

- Grados, minutos y segundos decimales + prefijo (`+/-`). El signo + para (**`N/E`**) es opcional
  `+/-D°M'S.SS" +/-D°M'S.SS"`
  Ejemplo: **`33°55'11.11" -22°44'55.25"`**

- Grados y minutos decimales + orientación como sufijo (`N/S`, `E/W`)
  `D°M.MM'N/S D°M.MM'W/E`
  Ejemplo: **`33°55.55'N 22°44.44'W`**

- Grados y minutos decimales + prefijo (`+/-`). El signo + para (**`N/E`**) es opcional
  `+/-D°M.MM' +/-D°M.MM'`
  Ejemplo: **`+33°55.55' -22°44.44'`**

- Grados decimales + orientación como sufijo (`N/S`, `E/W`)
  `D.DDN/S D.DDW/E`
  Ejemplo: **`33.33N 22.22W`**

- Grados decimales + prefijo (+/-). El signo + para (N/S, E/W) es opcional
  +/-D.DD +/-D.DD
  Ejemplo: 33.33 -22.22

Ejemplos de combinación de formatos:
33.33N -22°44'55.25"
33.33 22°44'55.25"W
33.33 22.45

Atributo Exif de Altova: Geolocation

El motor XPath/XQuery de Altova genera el atributo personalizado **Geolocation** a partir de las etiquetas de metadatos Exif estándar. Este atributo es una concatenación de cuatro etiquetas Exif (GPSLatitude, GPSLatitudeRef, GPSLongitude, GPSLongitudeRef) seguidas de unidades:

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

geolocation-distance-mi [altova:]

altova:geolocation-distance-mi(CadenaEntradaGeoubicación-1 *como xs:string*, **CadenaEntradaGeoubicación-2** *como xs:string*) como **xs:decimal** XP3.1 XQ3.1
Calcula la distancia en millas que existe entre dos geoubicaciones. El formato que puede utilizarse para dar las cadenas de entrada aparece más abajo. Los valores de latitud están comprendidos entre +90 y -90 (N a S). Los valores de longitud están comprendidos entre +180 y -180 (E a W).

Nota: la función image-exif-data 2155 y el atributo de metadatos Exif @Geolocation 2155 pueden utilizarse para suministrar las cadenas de entrada de geoubicaciones.

Ejemplos

- **altova:geolocation-distance-mi**("33.33 -22.22", "48°51'29.6""N 24°17'40.2""")
  devuelve el xs:decimal 2599.40652340653

Formato de las cadenas de entrada de geoubicaciones:

La cadena de entrada de la geoubicación debe contener la latitud y la longitud (en ese orden) se paradas por un espacio en blanco. Ambas pueden estar en cualquier formato de los que se indican más abajo y puede combinar formatos distintos. Es decir, la latitud puede estar en un formato y la longitud en otro. Los valores de la latitud deben estar comprendidos entre +90 y -90 (N a S). Los valores de longitud deben estar comprendidos entre +180 y -180 (E a W).

Nota: si utiliza comillas simples o dobles para delimitar el argumento de la cadena de entrada, esto dará lugar a un conflicto con las comillas simples o dobles que se utilizan, respectivamente, para indicar los valores de los minutos y los segundos. Si esto ocurre, debe añadir caracteres de escape a las comillas utilizadas para los minutos y segundos (esto se hace duplicando las comillas.

© 2016-2022 Altova GmbH

- Grados, minutos y segundos decimales + orientación como sufijo (N/S, E/W)
  D°M'S.SS"N/S D°M'S.SS"W/E
  Ejemplo: **33°55'11.11"N 22°44'55.25"W**

- Grados, minutos y segundos decimales + prefijo (+/-). El signo + para (**N/E**) es opcional
  +/-D°M'S.SS" +/-D°M'S.SS"
  Ejemplo: **33°55'11.11" -22°44'55.25"**

- Grados y minutos decimales + orientación como sufijo (N/S, E/W)
  D°M.MM'N/S D°M.MM'W/E
  Ejemplo: **33°55.55'N 22°44.44'W**

- Grados y minutos decimales + prefijo (+/-). El signo + para (**N/E**) es opcional
  +/-D°M.MM' +/-D°M.MM'
  Ejemplo: **+33°55.55' -22°44.44'**

- Grados decimales + orientación como sufijo (N/S, E/W)
  D.DDN/S D.DDW/E
  Ejemplo: **33.33N 22.22W**

- Grados decimales + prefijo (+/-). El signo + para (**N/S, E/W**) es opcional
  +/-D.DD +/-D.DD
  Ejemplo: **33.33 -22.22**

Ejemplos de combinación de formatos:
33.33N -22°44'55.25"
33.33 22°44'55.25"W
33.33 22.45

- *Atributo Exif de Altova: Geolocation*

  El motor XPath/XQuery de Altova genera el atributo personalizado **Geolocation** a partir de las etiquetas de metadatos Exif estándar. Este atributo es una concatenación de cuatro etiquetas Exif (GPSLatitude, GPSLatitudeRef, GPSLongitude, GPSLongitudeRef) seguidas de unidades:

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

altova:geolocations-bounding-rectangle(Geolocations como *xs:sequence*, **GeolocationOutputStringFormat como** *xs:integer***)** como **xs:string** **XP3.1** **XQ3.1**
Toma una secuencia de cadenas de texto como primer argumento; cada cadena de esa secuencia es una geoubicación. La función devuelve una secuencia de dos cadenas que son, respectivamente, las coordinadas de geoubicación de la parte superior izquierda e inferior derecha de un rectángulo delimitado que tiene el tamaño exacto para contener las geoubicaciones suministradas en el primer argumento. Más abajo se enumeran los formatos en que se puede dar una cadena de entrada de geoubicación (*véase "Formato de las cadenas de entrada de geoubicaciones"*). Los valores de latitud están comprendidos entre +90 y -90 (N a S). Los valores de longitud están comprendidos entre +180 y -180 (E a W).

El segundo argumento de la función indica el formato de las dos cadenas de geoubicación de la secuencia de salida. El argumento toma un valor entero entre `1` y `4`, donde cada valor representa un formato distinto de las cadenas de entrada de geoubicaciones (*véase "Formato de las cadenas de salida de geoubicaciones"*).

Nota: la función `image-exif-data` [2155] y los atributos de metadatos Exif se pueden usar para suministrar las cadenas de entrada.

Ejemplos

- **`altova:geolocations-bounding-rectangle`**`(("48.2143531 16.3707266", "51.50939 -0.11832"), 1)` devuelve la secuencia `("51°30'33.804"N 0°7'5.952"W", "48°12'51.67116"N 16°22'14.61576"E")`
- **`altova:geolocations-bounding-rectangle`**`(("48.2143531 16.3707266", "51.50939 -0.11832", "42.5584577 -70.8893334"), 4)` devuelve la secuencia `("51.50939 -70.8893334", "42.5584577 16.3707266")`

Formato de las cadenas de entrada de geoubicaciones:

La cadena de entrada de la geoubicación debe contener la latitud y la longitud (en ese orden) se paradas por un espacio en blanco. Ambas pueden estar en cualquier formato de los que se indican más abajo y puede combinar formatos distintos. Es decir, la latitud puede estar en un formato y la longitud en otro. Los valores de la latitud deben estar comprendidos entre `+90` y `-90` (`N` a `S`). Los valores de longitud deben estar comprendidos entre `+180` y `-180` (`E` a `W`).

Nota: si utiliza comillas simples o dobles para delimitar el argumento de la cadena de entrada, esto dará lugar a un conflicto con las comillas simples o dobles que se utilizan, respectivamente, para indicar los valores de los minutos y los segundos. Si esto ocurre, debe añadir caracteres de escape a las comillas utilizadas para los minutos y segundos (esto se hace duplicando las comillas.

- Grados, minutos y segundos decimales + orientación como sufijo (`N/S`, `E/W`)
  `D°M'S.SS"N/S D°M'S.SS"W/E`
  Ejemplo: **`33°55'11.11"N 22°44'55.25"W`**

- Grados, minutos y segundos decimales + prefijo (`+/-`). El signo + para (**`N/E`**) es opcional
  `+/-D°M'S.SS" +/-D°M'S.SS"`
  Ejemplo: **`33°55'11.11" -22°44'55.25"`**

- Grados y minutos decimales + orientación como sufijo (`N/S`, `E/W`)
  `D°M.MM'N/S D°M.MM'W/E`
  Ejemplo: **`33°55.55'N 22°44.44'W`**

- Grados y minutos decimales + prefijo (`+/-`). El signo + para (**`N/E`**) es opcional
  `+/-D°M.MM' +/-D°M.MM'`
  Ejemplo: **`+33°55.55' -22°44.44'`**

- Grados decimales + orientación como sufijo (`N/S`, `E/W`)
  `D.DDN/S D.DDW/E`
  Ejemplo: **`33.33N 22.22W`**

- Grados decimales + prefijo (`+/-`). El signo + para (**`N/S, E/W`**) es opcional
  `+/-D.DD +/-D.DD`
  Ejemplo: **`33.33 -22.22`**

© 2016-2022 Altova GmbH

Ejemplos de combinación de formatos:

```
33.33N  -22°44'55.25"
33.33  22°44'55.25"W
33.33  22.45
```

- *Formato de las cadenas de salida de las geoubicaciones:*

A la latitud y longitud suministradas se les aplica un formato de salida de los que se indican más abajo. El formato deseado se identifica con un identificador comprendido entre `1` y `4`. Los valores de latitud pueden estar comprendidos entre `+90` y `-90` (`N` a `S`). Los valores de longitud pueden estar comprendidos entre `+180` y `-180` (`E` a `W`).

1
Grados, minutos y segundos decimales + orientación como sufijo (`N/S, E/W`) `D°M'S.SS"N/S  D°M'S.SS"E/W` *Ejemplo*: `33°55'11.11"N  22°44'66.66"W`

2
Grados decimales + orientación como sufijo (`N/S`, `E/W`) `D.DDN/S  D.DDE/W` *Ejemplo*: `33.33N  22.22W`

3
Grados, minutos y segundos decimales + prefijo (`+/-`). El signo + para (`N/E`) es opcional `+/-D°M'S.SS"  +/-D°M'S.SS"` *Ejemplo*: `33°55'11.11"  -22°44'66.66"`

4
Grados decimales + prefijo (`+/-`). El signo + para (`N/E`) es opcional `+/-D.DD  +/-D.DD` *Ejemplo*: `33.33 -22.22`

- *Atributo Exif de Altova: Geolocation*

El motor XPath/XQuery de Altova genera el atributo personalizado `Geolocation` a partir de las etiquetas de metadatos Exif estándar. Este atributo es una concatenación de cuatro etiquetas Exif (`GPSLatitude`, `GPSLatitudeRef`, `GPSLongitude`, `GPSLongitudeRef`) seguidas de unidades:

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

▼ geolocation-within-polygon [altova:]

altova:geolocation-within-polygon(Geoubicación *como xs:string*, **((PuntoDePolígono** *como xs:string*)+)) como **xs:boolean** XP3.1 XQ3.1

Determina si **Geoubicación** (primer argumento) está dentro del área poligonal descrita por los argumentos **PuntoDePolígono**. Si los argumentos PuntoDePolígono no forman una figura cerrada (la figura se cierra cuando el primer y el último punto son el mismo), entonces el primer punto se añade implícitamente como último punto a fin de cerrar la figura. Todos los argumentos (Geoubicación y PuntoDePolígono+) se dan como cadenas de entrada de geoubicación (*formatos permitidos más abajo*). Si el argumento Geoubicación está dentro del área poligonal, entonces la función devuelve true(). De lo contrario, devuelve false(). Los valores de latitud están comprendidos entre +90 y -90 (N a S). Los valores de longitud están comprendidos entre +180 y -180 (E a W).

Nota: la función image-exif-data 2155 y el atributo de metadatos Exif @Geolocation 2155 pueden utilizarse para suministrar las cadenas de entrada de geoubicaciones.

▼ *Ejemplos*

- **altova:geolocation-within-polygon("33 -22", ("58 -32", "-78 -55", "48 24", "58 -32"))** devuelve true()
- **altova:geolocation-within-polygon("33 -22", ("58 -32", "-78 -55", "48 24"))** devuelve true()
- **altova:geolocation-within-polygon("33 -22", ("58 -32", "-78 -55", "48°51'29.6""N 24°17'40.2""))** devuelve true()

▼ *Formato de las cadenas de entrada de geoubicaciones:*

La cadena de entrada de la geoubicación debe contener la latitud y la longitud (en ese orden) se paradas por un espacio en blanco. Ambas pueden estar en cualquier formato de los que se indican más abajo y puede combinar formatos distintos. Es decir, la latitud puede estar en un formato y la longitud en otro. Los valores de la latitud deben estar comprendidos entre +90 y -90 (N a S). Los valores de longitud deben estar comprendidos entre +180 y -180 (E a W).

Nota: si utiliza comillas simples o dobles para delimitar el argumento de la cadena de entrada, esto dará lugar a un conflicto con las comillas simples o dobles que se utilizan, respectivamente, para indicar los valores de los minutos y los segundos. Si esto ocurre, debe añadir caracteres de escape a las comillas utilizadas para los minutos y segundos (esto se hace duplicando las comillas.

- Grados, minutos y segundos decimales + orientación como sufijo (N/S, E/W)
  D°M'S.SS"N/S D°M'S.SS"W/E
  Ejemplo: **33°55'11.11"N 22°44'55.25"W**

- Grados, minutos y segundos decimales + prefijo (+/-). El signo + para (**N/E**) es opcional
  +/-D°M'S.SS" +/-D°M'S.SS"
  Ejemplo: **33°55'11.11" -22°44'55.25"**

- Grados y minutos decimales + orientación como sufijo (N/S, E/W)
  D°M.MM'N/S D°M.MM'W/E
  Ejemplo: **33°55.55'N 22°44.44'W**

- Grados y minutos decimales + prefijo (+/-). El signo + para (**N/E**) es opcional
  +/-D°M.MM' +/-D°M.MM'

© 2016-2022 Altova GmbH

Ejemplo: `+33°55.55' -22°44.44'`

- Grados decimales + orientación como sufijo (`N/S`, `E/W`)
  `D.DDN/S D.DDW/E`
  Ejemplo: `33.33N 22.22W`

- Grados decimales + prefijo (`+/-`). El signo + para (`N/S, E/W`) es opcional
  `+/-D.DD +/-D.DD`
  Ejemplo: `33.33 -22.22`

Ejemplos de combinación de formatos:
`33.33N -22°44'55.25"`
`33.33 22°44'55.25"W`
`33.33 22.45`

Atributo Exif de Altova: Geolocation

El motor XPath/XQuery de Altova genera el atributo personalizado **`Geolocation`** a partir de las etiquetas de metadatos Exif estándar. Este atributo es una concatenación de cuatro etiquetas Exif (`GPSLatitude`, `GPSLatitudeRef`, `GPSLongitude`, `GPSLongitudeRef`) seguidas de unidades:

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

geolocation-within-rectangle [altova:]

`altova:geolocation-within-rectangle(Geoubicación` *como xs:string*, **`ÁnguloRectángulo-1`** *como xs:string*, **`ÁnguloRectángulo-2`** *como xs:string*`)` como **`xs:boolean`** **XP3.1** **XQ3.1**
Determina si **`Geoubicación`** (primer argumento) está dentro del rectángulo definido por el segundo y el tercer argumento (**`ÁnguloRectángulo-1`** y **`ÁnguloRectángulo-2`**), que indican ángulos opuestos del rectángulo. Todos los argumentos de la función se dan como cadenas de entrada de geoubicación (*formatos permitidos más abajo*). Si el argumento `Geoubicación` está dentro del rectángulo, entonces la función devuelve `true()`. De lo contrario, devuelve `false()`. Los valores de latitud están comprendidos entre `+90` y `-90` (`N` a `S`). Los valores de longitud están comprendidos entre `+180` y `-180` (`E` a `W`).

Nota: la función `image-exif-data` 2155 y el atributo de metadatos Exif `@Geolocation` 2155 pueden utilizarse para suministrar las cadenas de entrada de geoubicaciones.

Ejemplos

- **`altova:geolocation-within-rectangle(`**`"33 -22", "58 -32", "-48 24"`**`)`** devuelve `true()`
- **`altova:geolocation-within-rectangle(`**`"33 -22", "58 -32", "48 24"`**`)`** devuelve `false()`
- **`altova:geolocation-within-rectangle(`**`"33 -22", "58 -32", "48°51'29.6""S 24° 17'40.2""")` devuelve `true()`

Formato de las cadenas de entrada de geoubicaciones:

La cadena de entrada de la geoubicación debe contener la latitud y la longitud (en ese orden) se

paradas por un espacio en blanco. Ambas pueden estar en cualquier formato de los que se indican más abajo y puede combinar formatos distintos. Es decir, la latitud puede estar en un formato y la longitud en otro. Los valores de la latitud deben estar comprendidos entre `+90` y `-90` (`N` a `S`). Los valores de longitud deben estar comprendidos entre `+180` y `-180` (`E` a `W`).

Nota: si utiliza comillas simples o dobles para delimitar el argumento de la cadena de entrada, esto dará lugar a un conflicto con las comillas simples o dobles que se utilizan, respectivamente, para indicar los valores de los minutos y los segundos. Si esto ocurre, debe añadir caracteres de escape a las comillas utilizadas para los minutos y segundos (esto se hace duplicando las comillas.

- Grados, minutos y segundos decimales + orientación como sufijo (`N/S`, `E/W`)
  `D°M'S.SS"N/S D°M'S.SS"W/E`
  Ejemplo: **`33°55'11.11"N 22°44'55.25"W`**

- Grados, minutos y segundos decimales + prefijo (`+/-`). El signo + para (**`N/E`**) es opcional
  `+/-D°M'S.SS" +/-D°M'S.SS"`
  Ejemplo: **`33°55'11.11" -22°44'55.25"`**

- Grados y minutos decimales + orientación como sufijo (`N/S`, `E/W`)
  `D°M.MM'N/S D°M.MM'W/E`
  Ejemplo: **`33°55.55'N 22°44.44'W`**

- Grados y minutos decimales + prefijo (`+/-`). El signo + para (**`N/E`**) es opcional
  `+/-D°M.MM' +/-D°M.MM'`
  Ejemplo: **`+33°55.55' -22°44.44'`**

- Grados decimales + orientación como sufijo (`N/S`, `E/W`)
  `D.DDN/S D.DDW/E`
  Ejemplo: **`33.33N 22.22W`**

- Grados decimales + prefijo (`+/-`). El signo + para (**`N/S, E/W`**) es opcional
  `+/-D.DD +/-D.DD`
  Ejemplo: **`33.33 -22.22`**

Ejemplos de combinación de formatos:

```
33.33N  -22°44'55.25"
33.33  22°44'55.25"W
33.33  22.45
```

Atributo Exif de Altova: Geolocation

El motor XPath/XQuery de Altova genera el atributo personalizado **`Geolocation`** a partir de las etiquetas de metadatos Exif estándar. Este atributo es una concatenación de cuatro etiquetas Exif (`GPSLatitude`, `GPSLatitudeRef`, `GPSLongitude`, `GPSLongitudeRef`) seguidas de unidades:

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

© 2016-2022 Altova GmbH

[Subir 2143]

30.2.1.4 Funciones XPath/XQuery: relacionadas con imágenes

Las funciones de extensión XPath/XQuery para trabajar con imágenes son compatibles con la versión actual de XMLSpy y se pueden utilizar en (i) expresiones XPath en contextos XSLT o (ii) expresiones XQuery en documentos XQuery.

Nota sobre el nombre de las funciones y lenguajes

Puede utilizar todas las funciones de extensión de Altova en sus expresiones XPath/XQuery. Con ellas conseguirá funciones adicionales no disponibles en la biblioteca de funciones estándar de XPath, XQuery y XSLT. Las funciones de extensión de Altova están en el **espacio de nombres** `http://www.altova.com/xslt-extensions` y en esta sección se presentan con el prefijo, que se supone estará enlazado al espacio de nombres señalado. Tenga en cuenta que en futuras versiones del producto algunas funciones pueden dejar de ser compatibles o su comportamiento puede cambiar. Por tanto, consulte siempre la documentación del producto para conocer el funcionamiento de estas funciones en cada versión del producto.

Funciones XPath (en expresiones XPath en XSLT):	XP1 XP2 XP3.1.1
Funciones XSLT (en expresiones XPath en XSLT):	XSLT1 XSLT2 XSLT3
Funciones XQuery (en expresiones XQuery en XQuery):	XQ1 XQ3.1

▼ suggested-image-file-extension [altova:]

altova:suggested-image-file-extension(CadenaBase64 *como string*) como **string?** XP3.1 XQ3.1
Toma la codificación base64 de un archivo de imagen como argumento y devuelve la extensión de archivo de la imagen registrada en la codificación base64 de la imagen. El valor devuelto es una sugerencia basada en la información sobre el tipo de imagen disponible en la codificación. Si esta información no está disponible, entonces devuelve una cadena vacía. Esta función es muy práctica a la hora de guardar una imagen base64 como archivo y recuperar de forma dinámica una extensión de archivo adecuada.

Ejemplos

- **altova:suggested-image-file-extension**(/MisImágenes/TeléfonoMóvil/Imagen20141130.01) devuelve 'jpg'
- **altova:suggested-image-file-extension**($XML1/Personal/Persona/@photo) devuelve ''

En los ejemplos anteriores, se da por hecho que los nodos suministrados como argumento de la función contienen una imagen codificada en base64. El primer ejemplo recupera `jpg` como tipo de imagen y como extensión de archivo. En el segundo ejemplo, la codificación base64 dada no ofrece información sobre la extensión del archivo.

▼ image-exif-data [altova:]

altova:image-exif-data(CadenaBinariaBase64 *como string*) como **element?** XP3.1 XQ3.1

Toma una imagen JPEG codificada en base64 como argumento y devuelve un elemento llamado **Exif** que contiene los metadatos Exif de la imagen. Los metadatos Exif se crean como pares atributo-valor del elemento `Exif`. El nombre de los atributos son las etiquetas de datos Exif encontradas en la codificación base64. La lista de etiquetas Exif aparece más abajo. Si en lo datos Exif hay etiquetas de terceros, estas etiquetas y sus valores también se devuelven en un par atributo-valor. Además de las etiquetas de metadatos Exif estándar (*lista más abajo*), también se generan pares atributo-valor de Altova. Estos atributos Exif de Altova también se enumeran más abajo.

Ejemplos

- Para acceder a un atributo, utilice la función de esta manera:
  ```
  image-exif-data(//MisImágenes/Imagen20141130.01)/@GPSLatitude
  image-exif-data(//MisImágenes/Imagen20141130.01)/@Geolocation
  ```
- Para acceder a todos los atributos, utilice la función de esta manera:
  ```
  image-exif-data(//MisImágenes/Imagen20141130.01)/@*
  ```
- Para acceder al nombre de todos los atributos, utilice esta expresión:
  ```
  for $i in image-exif-data(//MisImágenes/Imagen20141130.01)/@* return name($i)
  ```
  Esto es muy práctico a la hora de averiguar el nombre de los atributos que devuelve la función.

Atributo Exif de Altova: Geolocation

El motor XPath/XQuery de Altova genera el atributo personalizado **Geolocation** a partir de las etiquetas de metadatos Exif estándar. Este atributo es una concatenación de cuatro etiquetas Exif (`GPSLatitude`, `GPSLatitudeRef`, `GPSLongitude`, `GPSLongitudeRef`) seguidas de unidades:

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

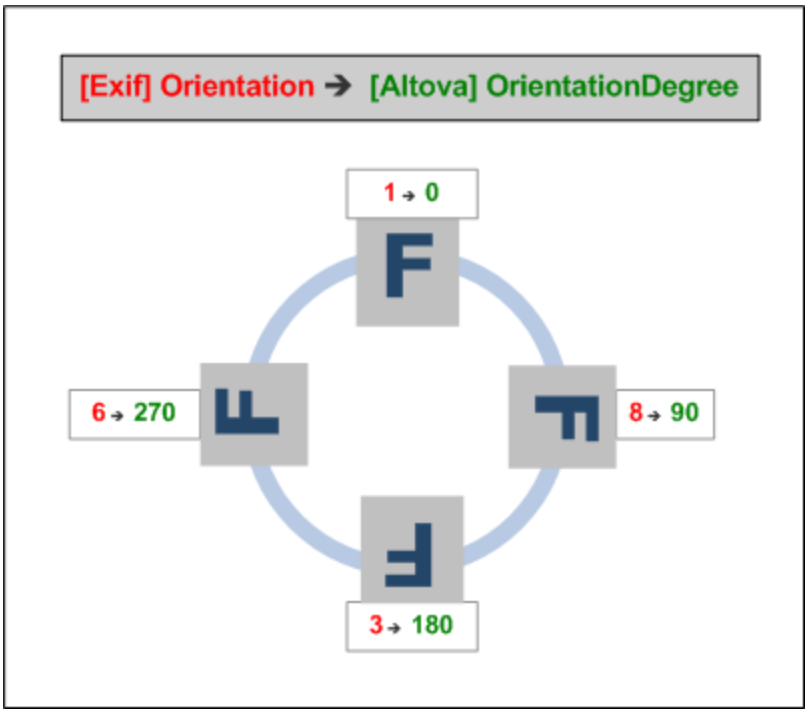
Atributo Exif de Altova: OrientationDegree

El motor XPath/XQuery de Altova genera el atributo personalizado **OrientationDegree** a partir de la etiqueta de metadatos Exif **Orientation**.

Este atributo transforma el valor entero de la etiqueta Exif `Orientation` (`1`, `8`, `3` o `6`) en el correspondiente valor en grados (`0`, `90`, `180`, `270`), tal y como describe el diagrama más abajo.

Debe tener en cuenta que los valores `2`, `4`, `5`, `7` de `Orientation` no se pueden traducir. Estas orientaciones se obtienen invirtiendo la imagen `1` en su eje central vertical para obtener la imagen con un valor de `2` e invirtiendo después esta imagen por pasos de 90 grados en el sentido de las agujas del reloj para obtener los valores de `7`, `4` y `5`, respectivamente.

© 2016-2022 Altova GmbH



- *Lista de etiquetas Exif estándar*
  - ImageWidth
  - ImageLength
  - BitsPerSample
  - Compression
  - PhotometricInterpretation
  - Orientation
  - SamplesPerPixel
  - PlanarConfiguration
  - YCbCrSubSampling
  - YCbCrPositioning
  - XResolution
  - YResolution
  - ResolutionUnit
  - StripOffsets
  - RowsPerStrip
  - StripByteCounts
  - JPEGInterchangeFormat
  - JPEGInterchangeFormatLength
  - TransferFunction
  - WhitePoint
  - PrimaryChromaticities
  - YCbCrCoefficients
  - ReferenceBlackWhite
  - DateTime
  - ImageDescription
  - Make

- Model
- Software
- Artist
- Copyright

- ExifVersion
- FlashpixVersion
- ColorSpace
- ComponentsConfiguration
- CompressedBitsPerPixel
- PixelXDimension
- PixelYDimension
- MakerNote
- UserComment
- RelatedSoundFile
- DateTimeOriginal
- DateTimeDigitized
- SubSecTime
- SubSecTimeOriginal
- SubSecTimeDigitized
- ExposureTime
- FNumber
- ExposureProgram
- SpectralSensitivity
- ISOSpeedRatings
- OECF
- ShutterSpeedValue
- ApertureValue
- BrightnessValue
- ExposureBiasValue
- MaxApertureValue
- SubjectDistance
- MeteringMode
- LightSource
- Flash
- FocalLength
- SubjectArea
- FlashEnergy
- SpatialFrequencyResponse
- FocalPlaneXResolution
- FocalPlaneYResolution
- FocalPlaneResolutionUnit
- SubjectLocation
- ExposureIndex
- SensingMethod
- FileSource
- SceneType
- CFAPattern
- CustomRendered
- ExposureMode
- WhiteBalance
- DigitalZoomRatio
- FocalLengthIn35mmFilm
- SceneCaptureType

© 2016-2022 Altova GmbH

- GainControl
- Contrast
- Saturation
- Sharpness
- DeviceSettingDescription
- SubjectDistanceRange
- ImageUniqueID

- GPSVersionID
- GPSLatitudeRef
- GPSLatitude
- GPSLongitudeRef
- GPSLongitude
- GPSAltitudeRef
- GPSAltitude
- GPSTimeStamp
- GPSSatellites
- GPSStatus
- GPSMeasureMode
- GPSDOP
- GPSSpeedRef
- GPSSpeed
- GPSTrackRef
- GPSTrack
- GPSImgDirectionRef
- GPSImgDirection
- GPSMapDatum
- GPSDestLatitudeRef
- GPSDestLatitude
- GPSDestLongitudeRef
- GPSDestLongitude
- GPSDestBearingRef
- GPSDestBearing
- GPSDestDistanceRef
- GPSDestDistance
- GPSProcessingMethod
- GPSAreaInformation
- GPSDateStamp
- GPSDifferential

[Subir 2155]

30.2.1.5 Funciones XPath/XQuery: numéricas

Las funciones de extensión numéricas de Altova pueden utilizarse en expresiones XPath y XQuery y ofrecen funciones adicionales para el procesamiento de datos. Estas funciones se pueden usar con los motores **XPath 3.0** y **XQuery 3.0** de Altova. Están disponibles en contextos XPath/XQuery.

Nota sobre el nombre de las funciones y lenguajes

Puede utilizar todas las funciones de extensión de Altova en sus expresiones XPath/XQuery. Con ellas conseguirá funciones adicionales no disponibles en la biblioteca de funciones estándar de XPath, XQuery y XSLT. Las funciones de extensión de Altova están en el **espacio de nombres** `http://www.altova.com/xslt-extensions` y en esta sección se presentan con el prefijo, que se supone estará enlazado al espacio de nombres señalado. Tenga en cuenta que en futuras versiones del producto algunas funciones pueden dejar de ser compatibles o su comportamiento puede cambiar. Por tanto, consulte siempre la documentación del producto para conocer el funcionamiento de estas funciones en cada versión del producto.

Funciones XPath (en expresiones XPath en XSLT):	XP1 XP2 XP3.1.1
Funciones XSLT (en expresiones XPath en XSLT):	XSLT1 XSLT2 XSLT3
Funciones XQuery (en expresiones XQuery en XQuery):	XQ1 XQ3.1

Funciones de numeración automática

generate-auto-number [altova:]

altova:generate-auto-number(ID *como xs:string*, **EmpiezaPor** *como xs:double*, **Incremento** *como xs:double*, **RestaurarAlCambiar** *como xs:string*) como `xs:integer` XP1 XP2 XQ1 XP3.1 XQ3.1

Genera un número cada vez que se llama a la función. El primer número, que se genera cuando se llama a la función por primera vez, viene dado por el argumento `EmpiezaPor`. Cada llamada posterior genera un número nuevo, que se incrementa en función del valor especificado en el argumento `Incremento`. De hecho, la función `generate-auto-number` crea un contador llamado como indique el argumento `ID` y este contador se incrementa cada vez que se llama a la función. Si el valor del argumento `RestaurarAlCambiar` cambia con respecto al valor que tenía en la llamada anterior, entonces el valor del número que se debe generar se restablece con el valor de `EmpiezaPor`. También puede restablecer la numeración automática con la función `altova:reset-auto-number`.

Ejemplo

- `altova:generate-auto-number("ChapterNumber", 1, 1, "SomeString")`

Devuelve un número cada vez que se llama a la función, empezando por `1` y con un incremento de `1` con cada llamada a función. Si el cuarto argumento continúa siendo `"SomeString"` en las llamadas posteriores, el incremento continuará. Cuando cambie el valor del cuarto argumento, se restaura el valor 1 del contador (llamado `ChapterNumber`). El valor de `ChapterNumber` también se puede restaurar llamando a la función `altova:reset-auto-number("ChapterNumber")`.

reset-auto-number [altova:]

altova:reset-auto-number(ID *como xs:string*) XP1 XP2 XQ1 XP3.1 XQ3.1

Esta función restaura el número del contador de numeración automática especificado en el argumento `ID`. El número se reemplaza con el número indicado en el argumento `EmpiezaPor` de la función `altova:generate-auto-number` que creó el contador especificado en el argumento `ID`.

Ejemplos

- `altova:reset-auto-number("ChapterNumber")` restablece el número del contador de

© 2016-2022 Altova GmbH

numeración automática llamado `ChapterNumber` que se creó con la función `altova:generate-auto-number`. El número se reemplaza con el valor del argumento `EmpiezaPor` de la función `altova:generate-auto-number` que creó `ChapterNumber`.

[Subir 2159]

Funciones numéricas

hex-string-to-integer [altova:]

altova:hex-string-to-integer(CadenaHex *as xs:string***)** como **xs:integer** XP3.1 XQ3.1

Toma un argumento de cadena que es el equivalente Base-16 de un entero del sistema decimal (Base-10) y devuelve un entero decimal.

Ejemplos

- **altova:hex-string-to-integer**('1') devuelve 1
- **altova:hex-string-to-integer**('9') devuelve 9
- **altova:hex-string-to-integer**('A') devuelve 10
- **altova:hex-string-to-integer**('B') devuelve 11
- **altova:hex-string-to-integer**('F') devuelve 15
- **altova:hex-string-to-integer**('G') devuelve un error
- **altova:hex-string-to-integer**('10') devuelve 16
- **altova:hex-string-to-integer**('01') devuelve 1
- **altova:hex-string-to-integer**('20') devuelve 32
- **altova:hex-string-to-integer**('21') devuelve 33
- **altova:hex-string-to-integer**('5A') devuelve 90
- **altova:hex-string-to-integer**('USA') devuelve un error

integer-to-hex-string [altova:]

altova:integer-to-hex-string(Entero *as xs:integer***)** como **xs:string** XP3.1 XQ3.1

Toma el argumento **Entero** y devuelve su equivalente Base-16 en forma de cadena.

Ejemplos

- **altova:integer-to-hex-string**(1) devuelve '1'
- **altova:integer-to-hex-string**(9) devuelve '9'
- **altova:integer-to-hex-string**(10) devuelve 'A'
- **altova:integer-to-hex-string**(11) devuelve 'B'
- **altova:integer-to-hex-string**(15) devuelve 'F'
- **altova:integer-to-hex-string**(16) devuelve '10'
- **altova:integer-to-hex-string**(32) devuelve '20'
- **altova:integer-to-hex-string**(33) devuelve '21'
- **altova:integer-to-hex-string**(90) devuelve '5A'

[Subir 2159]

Funciones de formato numérico

[Subir 2159]

30.2.1.6 Funciones XPath/XQuery: de esquema

Las funciones de extensión de Altova que enumeramos a continuación devuelven información del esquema. Más adelante verá descripciones de las funciones, junto con (i) ejemplos y (ii) una lista de los componentes del esquema y sus correspondientes propiedades. Estas funciones se pueden usar con los motores de Altova **XPath 3.0** y **XQuery 3.0**, y están disponibles en contextos XPath/XQuery.

Información sobre el esquema proveniente de documentos de esquema
La función **altova:schema** tiene dos argumentos: uno que no tiene argumentos y otro que tiene dos. La función que no tiene argumentos devuelve todo el esquema. A partir de ahí puede navegar por el esquema para encontrar los componentes que necesite. La función con dos argumentos devuelve un tipo concreto de componente al que se identifica por su QName. En ambos casos el valor de retorno es una función. Para ir al componente devuelto debe seleccionar una de sus propiedades. Si esta propiedad es un elemento no atómico (es decir, si es un componente), entonces puede seleccionar también una propiedad de este componente para seguir navegando. Si la propiedad seleccionada sí es un elemento atómico, entonces se devuelve el valor del elemento y no puede seguir navegando.

Nota: en las expresiones XPath de debe importar primero el esquema en el entorno de procesamiento (por ejemplo, XSLT), con la instrucción **xslt:import-schema**. En las expresiones XQuery, el esquema se debe importar de forma explícita.

Información sobre el esquema proveniente de nodos XML
La función **altova:type** envía el nodo de un documento XML y devuelve la información del tipo del modo desde el PSVI (Conjunto de información posterior a la validación de esquemas).

Nota sobre el nombre de las funciones y lenguajes

Puede utilizar todas las funciones de extensión de Altova en sus expresiones XPath/XQuery. Con ellas conseguirá funciones adicionales no disponibles en la biblioteca de funciones estándar de XPath, XQuery y XSLT. Las funciones de extensión de Altova están en el **espacio de nombres** **http://www.altova.com/xslt-extensions** y en esta sección se presentan con el prefijo, que se supone estará enlazado al espacio de nombres señalado. Tenga en cuenta que en futuras versiones del producto algunas funciones pueden dejar de ser compatibles o su comportamiento puede cambiar. Por tanto, consulte siempre la documentación del producto para conocer el funcionamiento de estas funciones en cada versión del producto.

Funciones XPath (en expresiones XPath en XSLT):	XP1 XP2 XP3.1.1
Funciones XSLT (en expresiones XPath en XSLT):	XSLT1 XSLT2 XSLT3
Funciones XQuery (en expresiones XQuery en XQuery):	XQ1 XQ3.1

© 2016-2022 Altova GmbH

altova:schema() como (function(xs:string) como item()*)? XP3.1 XQ3.1
Devuelve el componente **schema** al completo. Para navegar por este componente seleccione una de sus propiedades.

- Si esta propiedad es un componente seleccione una de sus propiedades para navegar hasta el siguiente nivel de profundidad. Puede repetir este paso para seguir navegando por el esquema.
- Si el componente es un valor atómico se devuelve este valor y no puede seguir navegando.

Las propiedades del componente **schema** son:

```
"type definitions"
"attribute declarations"
"element declarations"
"attribute group definitions"
"model group definitions"
"notation declarations"
"identity-constraint definitions"
```

Más abajo encontrará las propiedades del resto de tipos de componente.

Nota: en las expresiones XQuery, el esquema se debe importar de forma explícita. En las expresiones XPath debe importar primero el esquema en el entorno de procesamiento, por ejemplo en XSLT con la instrucción **xslt:import**.

Ejemplos

- `import schema "" at "C:\Test\ExpReport.xsd"; for $typedef in altova:schema() ("type definitions")`
  `return $typedef ("name")` devuelve los nombres de todos los tipos simples o complejos del esquema

- `import schema "" at "C:\Test\ExpReport.xsd";`
  `altova:schema() ("type definitions")[1]("name")` devuelve el nombre del primero de los tipos simples o complejos del esquema

Componentes y sus propiedades

Assertion

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Assertion"
test	Registro de propiedades XPath	

Attribute Declaration

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Attribute Declaration"

name	Cadena	Nombre local del atributo
target namespace	Cadena	URI del espacio de nombres del atributo
type definition	Simple Type o Complex Type	
scope	Una función con propiedades ("class":"Scope", "variety": "global" o "local", "parent": el Complex Type o Attribute Group contenedor)	
value constraint	Si está presente, una función con propiedades ("class": "Value Constraint", "variety": "fixed" o "default", "value": atomic value, "lexical form": string. Tenga en cuenta que la propiedad "value" no está disponible para los tipos namespace-sensitive	
inheritable	Booleano	

Attribute Group Declaration

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Attribute Group Definition"
name	Cadena	Nombre local del grupo de atributos
target namespace	Cadena	URI del espacio de nombres del grupo de atributos
attribute uses	Secuencia de (Attribute Use)	
attribute wildcard	Comodín de atributo opcional	

Attribute Use

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Attribute Use"
required	Booleano	true si el atributo es obligatorio, false si es opcional
value constraint	Véase la declaración de atributos	
inheritable	Booleano	

Attribute Wildcard

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Wildcard"
namespace constraint	Función con propiedades ("class":	

© 2016-2022 Altova GmbH

	"Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": secuencia de elementos xs:anyURI, "disallowed names": lista que contiene QNames y/o las cadenas "defined" y "definedSiblings"	
process contents	Cadena ("strict"|"lax"|"skip")	

- Complex Type

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Complex Type"
name	Cadena	Nombre local del tipo (vacío si es anónimo)
target namespace	Cadena	URI del espacio de nombres del tipo (vacío si es anónimo)
base type definition	Definición del Complex Type	
final	Secuencia de cadenas ("restriction"|"extension")	
context	Secuencia vacía (not implemented)	
derivation method	Cadena ("restriction"|"extension")	
abstract	Booleano	
attribute uses	Secuencia de elementos Attribute Use	
attribute wildcard	Comodín de atributo opcional	
content type	Función con propiedades: ("class":"Content Type", "variety":string ("element-only"|"empty"|"mixed"|"simple"), particle: partícula opcional, "open content": función con propiedades ("class":"Open Content", "mode": string ("interleave"|"suffix"), "wildcard": Wildcard), "simple type definition": Simple Type)	
prohibited substitutions	Secuencia de cadenas ("restriction"|"extension")	
assertions	Secuencia de elementos Assertion	

- Element Declaration

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Complex Type"
name	Cadena	Nombre local del tipo (vacío si es

		anónimo)
target namespace	Cadena	Namespace URI del tipo (vacío si es anónimo)
type definition	Simple Type o Complex Type	
type table	Función con propiedades ("class":"Type Table", "alternatives": secuencia de elementos Type Alternative, "default type definition": Simple Type o Complex Type)	
scope	Función con propiedades ("class":"Scope", "variety": ("global"|"local"), "parent": Complex Type opcional)	
value constraint	véase Attribute Declaration	
nillable	Booleano	
identity-constraint definitions	Secuencia de restricciones de identidad	
substitution group affiliations	Secuencia de declaraciones de elementos	
substitution group exclusions	Secuencia de cadenas ("restriction"|"extension")	
disallowed substitutions	Secuencia de cadenas ("restriction"|"extension"|"substitution")	
abstract	Booleano	

Element Wildcard

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Wildcard"
namespace constraint	Función con propiedades ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": secuencia de xs:anyURI, "disallowed names": lista que contiene QNames y/o las cadenas "defined" y "definedSiblings"	
process contents	Cadena ("strict"|"lax"|"skip")	

Facet

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	El nombre de la faceta, por ejemplo "minLength" o "enumeration"
value	Depende de la faceta	El valor de la faceta

© 2016-2022 Altova GmbH

fixed	Booleano	
typed-value	Solo para facetas de enumeración, Array(xs:anyAtomicType*)	Una matriz que contiene los valores de la enumeración, cada uno de los cuales puede ser una secuencia de valores atómicos. (Nota: para la faceta de enumeración, la propiedad "value" es un secuencia de cadenas, independientemente del tipo)

Identity Constraint

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Identity-Constraint Definition"
name	Cadena	Nombre local de la restricción
target namespace	Cadena	URI del espacio de nombres de la restricción
identity-constraint category	Cadena ("key"|"unique"|"keyRef")	
selector	Registro de propiedades XPath	
fields	Secuencia de registros de propiedades XPath	
referenced key	(Solo para keyRef): Identity Constraint	La restricción clave correspondiente

Model Group

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Model Group"
compositor	Cadena ("sequence"|"choice"|"all")	
particles	Secuencia de partículas	

Model Group Definition

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Model Group Definition"
name	Cadena	Nombre local del grupo de modelos
target namespace	Cadena	URI del espacio de nombres del grupo de modelos
model group	Model Group	

Notation

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Notation Declaration"
name	Cadena	Nombre local de la notación
target namespace	Cadena	URI del espacio de nombres de la notación
system identifier	anyURI	
public identifier	Cadena	

Particle

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Particle"
min occurs	Número entero	
max occurs	Número entero o cadena ("unbounded")	
term	Element Declaration, Element Wildcard o ModelGroup	

Simple Type

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Simple Type Definition"
name	Cadena	Nombre local del tipo (vacío si es anónimo)
target namespace	Cadena	URI del espacio de nombres del tipo (vacío si es anónimo)
final	Secuencia de cadenas ("restriction"|"extension"|"list"|"union")	
context	Componente contenedor	
base type definition	Simple Type	
facets	Secuencia de facetas	
fundamental facets	Secuencia vacía (no implementada)	
variety	Cadena ("atomic"|"list"|"union")	
primitive type definition	Simple Type	
item type definition	(Solo para tipos de lista) Simple Type	
member type definitions	(Solo para tipos de unión) Secuencia de elementos Simple Type	

Type Alternative

© 2016-2022 Altova GmbH

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Type Alternative"
test	Registro de propiedades XPath	
type definition	Simple Type o Complex Type	

XPath Property Record

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
namespace bindings	Secuencia de funciones con propiedades ("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	El URI de base estático de la expresión XPath
expression	Cadena	La expresión XPath como cadena de texto

altova:schema(ComponentKind *as xs:string*, **Name** *as xs:QName*) **como (function(xs:string) como item()*)?** **XP3.1 XQ3.1**

Devuelve el tipo de componente que se indica en el primer argumento que tiene el mismo nombre que el que se indica en el segundo argumento. Para seguir navegando seleccione una de las propiedades del componente.

- Si esta propiedad es un componente seleccione una de sus propiedades para navegar hasta el siguiente nivel de profundidad. Puede repetir este paso para seguir navegando por el esquema.
- Si el componente es un valor atómico se devuelve este valor y no puede seguir navegando.

Nota: en las expresiones XQuery, el esquema se debe importar de forma explícita. En las expresiones XPath debe importar primero el esquema en el entorno de procesamiento, por ejemplo en XSLT con la instrucción **xslt:import**.

Ejemplos

- ```
  import schema "" at "C:\Test\ExpReport.xsd";
  altova:schema("element declaration", xs:QName("OrgChart"))("type definition")
  ("content type")("particles")[3]!.("term")("kind")
  ```
  devuelve la propiedad `kind` del término del tercer componente `particles`. Este componente desciende de la declaración de elementos que tiene un `QName` de `OrgChart`

- ```
  import schema "" at "C:\Test\ExpReport.xsd";
  let $typedef := altova:schema("type definition", xs:QName("emailType"))
  for $facet in $typedef ("facets")
  return [$facet ("kind"), $facet("value")]
  ```
  devuelve, por cada **facet** de cada componente **emailType**, una matriz que contiene el tipo y el valor de ese elemento **facet**

Componentes y sus propiedades

- Assertion

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Assertion"
test	Registro de propiedades XPath	

- Attribute Declaration

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Attribute Declaration"
name	Cadena	Nombre local del atributo
target namespace	Cadena	URI del espacio de nombres del atributo
type definition	Simple Type o Complex Type	
scope	Una función con propiedades ("class":"Scope", "variety": "global" o "local", "parent": el Complex Type o Attribute Group contenedor)	
value constraint	Si está presente, una función con propiedades ("class": "Value Constraint", "variety": "fixed" o "default", "value": atomic value, "lexical form": string. Tenga en cuenta que la propiedad "value" no está disponible para los tipos namespace-sensitive	
inheritable	Booleano	

- Attribute Group Declaration

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Attribute Group Definition"
name	Cadena	Nombre local del grupo de atributos
target namespace	Cadena	URI del espacio de nombres del grupo de atributos
attribute uses	Secuencia de (Attribute Use)	
attribute wildcard	Comodín de atributo opcional	

- Attribute Use

Nombre de la	Tipo de la propiedad	Valor de la propiedad

© 2016-2022 Altova GmbH

propiedad		
kind	Cadena	"Attribute Use"
required	Booleano	true si el atributo es obligatorio, false si es opcional
value constraint	Véase la declaración de atributos	
inheritable	Booleano	

- Attribute Wildcard

Nombre de la propiedad	**Tipo de la propiedad**	**Valor de la propiedad**
kind	Cadena	"Wildcard"
namespace constraint	Función con propiedades ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": secuencia de elementos xs:anyURI, "disallowed names": lista que contiene QNames y/o las cadenas "defined" y "definedSiblings"	
process contents	Cadena ("strict"|"lax"|"skip")	

- Complex Type

Nombre de la propiedad	**Tipo de la propiedad**	**Valor de la propiedad**
kind	Cadena	"Complex Type"
name	Cadena	Nombre local del tipo (vacío si es anónimo)
target namespace	Cadena	URI del espacio de nombres del tipo (vacío si es anónimo)
base type definition	Definición del Complex Type	
final	Secuencia de cadenas ("restriction"|"extension")	
context	Secuencia vacía (not implemented)	
derivation method	Cadena ("restriction"|"extension")	
abstract	Booleano	
attribute uses	Secuencia de elementos Attribute Use	
attribute wildcard	Comodín de atributo opcional	
content type	Función con propiedades: ("class":"Content Type", "variety":string ("element-only"|"empty"|"mixed"|"simple"), particle: partícula opcional, "open content": función con propiedades	

	("class":"Open Content", "mode": string ("interleave"|"suffix"), "wildcard": Wildcard), "simple type definition": Simple Type)	
prohibited substitutions	Secuencia de cadenas ("restriction"|"extension")	
assertions	Secuencia de elementos Assertion	

Element Declaration

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Complex Type"
name	Cadena	Nombre local del tipo (vacío si es anónimo)
target namespace	Cadena	Namespace URI del tipo (vacío si es anónimo)
type definition	Simple Type o Complex Type	
type table	Función con propiedades ("class":"Type Table", "alternatives": secuencia de elementos Type Alternative, "default type definition": Simple Type o Complex Type)	
scope	Función con propiedades ("class":"Scope", "variety": ("global"|"local"), "parent": Complex Type opcional)	
value constraint	véase Attribute Declaration	
nillable	Booleano	
identity-constraint definitions	Secuencia de restricciones de identidad	
substitution group affiliations	Secuencia de declaraciones de elementos	
substitution group exclusions	Secuencia de cadenas ("restriction"|"extension")	
disallowed substitutions	Secuencia de cadenas ("restriction"|"extension"|"substitution")	
abstract	Booleano	

Element Wildcard

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Wildcard"

© 2016-2022 Altova GmbH

namespace constraint	Función con propiedades ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": secuencia de xs:anyURI, "disallowed names": lista que contiene QNames y/o las cadenas "defined" y "definedSiblings"	
process contents	Cadena ("strict"|"lax"|"skip")	

Facet

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	El nombre de la faceta, por ejemplo "minLength" o "enumeration"
value	Depende de la faceta	El valor de la faceta
fixed	Booleano	
typed-value	Solo para facetas de enumeración, Array(xs:anyAtomicType*)	Una matriz que contiene los valores de la enumeración, cada uno de los cuales puede ser una secuencia de valores atómicos. (Nota: para la faceta de enumeración, la propiedad "value" es un secuencia de cadenas, independientemente del tipo)

Identity Constraint

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Identity-Constraint Definition"
name	Cadena	Nombre local de la restricción
target namespace	Cadena	URI del espacio de nombres de la restricción
identity-constraint category	Cadena ("key"|"unique"|"keyRef")	
selector	Registro de propiedades XPath	
fields	Secuencia de registros de propiedades XPath	
referenced key	(Solo para keyRef): Identity Constraint	La restricción clave correspondiente

Model Group

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Model Group"

compositor	Cadena ("sequence"|"choice"|"all")	
particles	Secuencia de partículas	

Model Group Definition

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Model Group Definition"
name	Cadena	Nombre local del grupo de modelos
target namespace	Cadena	URI del espacio de nombres del grupo de modelos
model group	Model Group	

Notation

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Notation Declaration"
name	Cadena	Nombre local de la notación
target namespace	Cadena	URI del espacio de nombres de la notación
system identifier	anyURI	
public identifier	Cadena	

Particle

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Particle"
min occurs	Número entero	
max occurs	Número entero o cadena ("unbounded")	
term	Element Declaration, Element Wildcard o ModelGroup	

Simple Type

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Simple Type Definition"
name	Cadena	Nombre local del tipo (vacío si es anónimo)
target namespace	Cadena	URI del espacio de nombres del tipo (vacío si es anónimo)

© 2016-2022 Altova GmbH

final	Secuencia de cadenas ("restriction"|"extension"|"list"|"union")	
context	Componente contenedor	
base type definition	Simple Type	
facets	Secuencia de facetas	
fundamental facets	Secuencia vacía (no implementada)	
variety	Cadena ("atomic"|"list"|"union")	
primitive type definition	Simple Type	
item type definition	(Solo para tipos de lista) Simple Type	
member type definitions	(Solo para tipos de unión) Secuencia de elementos Simple Type	

Type Alternative

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Type Alternative"
test	Registro de propiedades XPath	
type definition	Simple Type o Complex Type	

XPath Property Record

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
namespace bindings	Secuencia de funciones con propiedades ("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	El URI de base estático de la expresión XPath
expression	Cadena	La expresión XPath como cadena de texto

altova:type(Node *as item?***) como (function(xs:string) como item()*)?** XP3.1 XQ3.1

La función **altova:type** indica un nodo de elemento o atributo de un documento XML y devuelve la información del tipo de nodo del PSVI (Conjunto de información posterior a la validación de esquemas).

Nota: el documento XML debe tener una declaración de esquema para que se pueda hacer referencia al esquema.

Ejemplos

- `for $element in //Email`

```
let $type := altova:type($element)
return $type
```
devuelve una función que contiene información sobre el tipo de nodo

- ```
  for $element in //Email
  let $type := altova:type($element)
  return $type ("kind")
  ```
  toma el componente de tipo del nodo (tipo simple o complejo) y devuelve el valor de la propiedad **kind** del componente

El parámetro **"_props"** devuelve las propiedades del componente seleccionado. Por ejemplo:

- ```
  for $element in //Email
  let $type := altova:type($element)
  return ($type ("kind"), $type ("_props"))
  ```
  toma el componente de tipo del nodo (tipo simple o complejo) y devuelve (i) el valor de la propiedad **kind** del componente y después (ii) las propiedades de ese componente

Componentes y sus propiedades

Assertion

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Assertion"
test	Registro de propiedades XPath	

Attribute Declaration

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Attribute Declaration"
name	Cadena	Nombre local del atributo
target namespace	Cadena	URI del espacio de nombres del atributo
type definition	Simple Type o Complex Type	
scope	Una función con propiedades ("class":"Scope", "variety": "global" o "local", "parent": el Complex Type o Attribute Group contenedor)	
value constraint	Si está presente, una función con propiedades ("class": "Value Constraint", "variety": "fixed" o "default", "value": atomic value, "lexical form": string. Tenga en cuenta que la propiedad "value" no está disponible para los tipos namespace-sensitive	

© 2016-2022 Altova GmbH

inheritable	Booleano	

Attribute Group Declaration

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Attribute Group Definition"
name	Cadena	Nombre local del grupo de atributos
target namespace	Cadena	URI del espacio de nombres del grupo de atributos
attribute uses	Secuencia de (Attribute Use)	
attribute wildcard	Comodín de atributo opcional	

Attribute Use

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Attribute Use"
required	Booleano	true si el atributo es obligatorio, false si es opcional
value constraint	Véase la declaración de atributos	
inheritable	Booleano	

Attribute Wildcard

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Wildcard"
namespace constraint	Función con propiedades ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": secuencia de elementos xs:anyURI, "disallowed names": lista que contiene QNames y/o las cadenas "defined" y "definedSiblings"	
process contents	Cadena ("strict"|"lax"|"skip")	

Complex Type

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Complex Type"
name	Cadena	Nombre local del tipo (vacío si es anónimo)

target namespace	Cadena	URI del espacio de nombres del tipo (vacío si es anónimo)
base type definition	Definición del Complex Type	
final	Secuencia de cadenas ("restriction"|"extension")	
context	Secuencia vacía (not implemented)	
derivation method	Cadena ("restriction"|"extension")	
abstract	Booleano	
attribute uses	Secuencia de elementos Attribute Use	
attribute wildcard	Comodín de atributo opcional	
content type	Función con propiedades: ("class":"Content Type", "variety":string ("element-only"|"empty"|"mixed"|"simple"), particle: partícula opcional, "open content": función con propiedades ("class":"Open Content", "mode": string ("interleave"|"suffix"), "wildcard": Wildcard), "simple type definition": Simple Type)	
prohibited substitutions	Secuencia de cadenas ("restriction"|"extension")	
assertions	Secuencia de elementos Assertion	

- Element Declaration

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Complex Type"
name	Cadena	Nombre local del tipo (vacío si es anónimo)
target namespace	Cadena	Namespace URI del tipo (vacío si es anónimo)
type definition	Simple Type o Complex Type	
type table	Función con propiedades ("class":"Type Table", "alternatives": secuencia de elementos Type Alternative, "default type definition": Simple Type o Complex Type)	
scope	Función con propiedades ("class":"Scope", "variety": ("global"|"local"), "parent": Complex Type opcional)	
value constraint	véase Attribute Declaration	
nillable	Booleano	

© 2016-2022 Altova GmbH

identity-constraint definitions	Secuencia de restricciones de identidad	
substitution group affiliations	Secuencia de declaraciones de elementos	
substitution group exclusions	Secuencia de cadenas ("restriction"|"extension")	
disallowed substitutions	Secuencia de cadenas ("restriction"|"extension"|"substitution")	
abstract	Booleano	

- Element Wildcard

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Wildcard"
namespace constraint	Función con propiedades ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": secuencia de xs:anyURI, "disallowed names": lista que contiene QNames y/o las cadenas "defined" y "definedSiblings"	
process contents	Cadena ("strict"|"lax"|"skip")	

- Facet

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	El nombre de la faceta, por ejemplo "minLength" o "enumeration"
value	Depende de la faceta	El valor de la faceta
fixed	Booleano	
typed-value	Solo para facetas de enumeración, Array(xs:anyAtomicType*)	Una matriz que contiene los valores de la enumeración, cada uno de los cuales puede ser una secuencia de valores atómicos. (Nota: para la faceta de enumeración, la propiedad "value" es un secuencia de cadenas, independientemente del tipo)

- Identity Constraint

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Identity-Constraint Definition"

name	Cadena	Nombre local de la restricción
target namespace	Cadena	URI del espacio de nombres de la restricción
identity-constraint category	Cadena ("key"|"unique"|"keyRef")	
selector	Registro de propiedades XPath	
fields	Secuencia de registros de propiedades XPath	
referenced key	(Solo para keyRef): Identity Constraint	La restricción clave correspondiente

Model Group

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Model Group"
compositor	Cadena ("sequence"|"choice"|"all")	
particles	Secuencia de partículas	

Model Group Definition

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Model Group Definition"
name	Cadena	Nombre local del grupo de modelos
target namespace	Cadena	URI del espacio de nombres del grupo de modelos
model group	Model Group	

Notation

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Notation Declaration"
name	Cadena	Nombre local de la notación
target namespace	Cadena	URI del espacio de nombres de la notación
system identifier	anyURI	
public identifier	Cadena	

Particle

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad

© 2016-2022 Altova GmbH

kind	Cadena	"Particle"
min occurs	Número entero	
max occurs	Número entero o cadena ("unbounded")	
term	Element Declaration, Element Wildcard o ModelGroup	

- Simple Type

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Simple Type Definition"
name	Cadena	Nombre local del tipo (vacío si es anónimo)
target namespace	Cadena	URI del espacio de nombres del tipo (vacío si es anónimo)
final	Secuencia de cadenas ("restriction"|"extension"|"list"|"union")	
context	Componente contenedor	
base type definition	Simple Type	
facets	Secuencia de facetas	
fundamental facets	Secuencia vacía (no implementada)	
variety	Cadena ("atomic"|"list"|"union")	
primitive type definition	Simple Type	
item type definition	(Solo para tipos de lista) Simple Type	
member type definitions	(Solo para tipos de unión) Secuencia de elementos Simple Type	

- Type Alternative

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
kind	Cadena	"Type Alternative"
test	Registro de propiedades XPath	
type definition	Simple Type o Complex Type	

- XPath Property Record

Nombre de la propiedad	Tipo de la propiedad	Valor de la propiedad
namespace bindings	Secuencia de funciones con propiedades ("prefix": string, "namespace": anyURI)	

default namespace	anyURI	
base URI	anyURI	El URI de base estático de la expresión XPath
expression	Cadena	La expresión XPath como cadena de texto

30.2.1.7 Funciones XPath/XQuery: de secuencia

Las funciones de extensión de Altova para trabajar con secuencias pueden utilizarse en expresiones XPath y XQuery y ofrecen funciones adicionales para el procesamiento de datos. Estas funciones se pueden usar con los motores **XPath 3.0** y **XQuery 3.0** de Altova. Están disponibles en contextos XPath/XQuery.

Nota sobre el nombre de las funciones y lenguajes

Puede utilizar todas las funciones de extensión de Altova en sus expresiones XPath/XQuery. Con ellas conseguirá funciones adicionales no disponibles en la biblioteca de funciones estándar de XPath, XQuery y XSLT. Las funciones de extensión de Altova están en el **espacio de nombres** `http://www.altova.com/xslt-extensions` y en esta sección se presentan con el prefijo, que se supone estará enlazado al espacio de nombres señalado. Tenga en cuenta que en futuras versiones del producto algunas funciones pueden dejar de ser compatibles o su comportamiento puede cambiar. Por tanto, consulte siempre la documentación del producto para conocer el funcionamiento de estas funciones en cada versión del producto.

Funciones XPath (en expresiones XPath en XSLT):	XP1 XP2 XP3.1.1
Funciones XSLT (en expresiones XPath en XSLT):	XSLT1 XSLT2 XSLT3
Funciones XQuery (en expresiones XQuery en XQuery):	XQ1 XQ3.1

▼ attributes [altova:]

altova:attributes(NombreAtributo *as xs:string***)** como **attribute()*** XP3.1 XQ3.1
Devuelve todos los atributos cuyo nombre local coincida con el nombre dado como argumento de entrada (`NombreAtributo`). La búsqueda tiene en cuenta el uso de mayúsculas y minúsculas y se lleva a cabo en el eje `attribute::`.

- *Ejemplos*
  - **altova:attributes**("MiAtributo") devuelve `MiAtributo()*`

altova:attributes(NombreAtributo *as xs:string*, **OpcionesBúsqueda** *as xs:string***)** como **attribute()*** XP3.1 XQ3.1
Devuelve todos los atributos cuyo nombre local coincida con el nombre dado como argumento de entrada

© 2016-2022 Altova GmbH

(NombreAtributo). La búsqueda tiene en cuenta el uso de mayúsculas y minúsculas y se lleva a cabo en el eje attribute::. El segundo argumento es una cadena con marcas de búsqueda. Estas son las marcas disponibles:

r = habilita la búsqueda de expresiones regulares. En este caso, NombreAtributo debe ser una cadena de búsqueda de expresión regular;
i = la búsqueda no tiene en cuenta el uso de mayúsculas y minúsculas;
p = incluye el prefijo de espacio de nombres en la búsqueda. En este caso, NombreAtributo debe contener el prefijo de espacio de nombres (p. ej.: MiAtributo).

Las marcas pueden escribirse en cualquier orden y no hace falta utilizar todas. Si usa marcas no válidas, se genera un error. También puede usar una cadena vacía para el segundo argumento. Esto tiene el mismo efecto que usar solo el primer argumento. Sin embargo, no está permitido usar una secuencia vacía.

Ejemplos

- **altova:attributes**("MiAtributo", "rip") devuelve MiAtributo()*
- **altova:attributes**("MiAtributo", "pri") devuelve MiAtributo()*
- **altova:attributes**("MiAtributo", "") devuelve MiAtributo()*
- **altova:attributes**("MiAtributo", "Rip") devuelve un error de marca desconocida.
- **altova:attributes**("MiAtributo",) devuelve un error diciendo que falta el segundo argumento.

elements [altova:]

altova:elements(NombreElemento *as xs:string***)** como **elemento()*** **XP3.1** **XQ3.1**
Devuelve todos los elementos cuyo nombre local coincida con el nombre dado como argumento de entrada (NombreElemento). La búsqueda tiene en cuenta el uso de mayúsculas y minúsculas y se lleva a cabo en el eje child::.

Ejemplos

- **altova:elements**("MiElemento") devuelve MiElemento()*

altova:elements(NombreElemento *as xs:string*, **OpcionesBúsqueda** *as xs:string***)** como **elemento()*** **XP3.1** **XQ3.1**
Devuelve todos los elementos cuyo nombre local coincida con el nombre dado como argumento de entrada (NombreElemento). La búsqueda tiene en cuenta el uso de mayúsculas y minúsculas y se lleva a cabo en el eje child::. El segundo argumento es una cadena con marcas de búsqueda. Estas son las marcas disponibles:

r = habilita la búsqueda de expresiones regulares. En este caso, NombreElemento debe ser una cadena de búsqueda de expresión regular;
i = la búsqueda no tiene en cuenta el uso de mayúsculas y minúsculas;
p = incluye el prefijo de espacio de nombres en la búsqueda. En este caso, NombreElemento debe contener el prefijo de espacio de nombres (p. ej.: MiElemento).

Las marcas pueden escribirse en cualquier orden y no hace falta utilizar todas. Si usa marcas no válidas, se genera un error. También puede usar una cadena vacía para el segundo argumento. Esto tiene el mismo efecto que usar solo el primer argumento. Sin embargo, no está permitido usar una secuencia

vacía.

Ejemplos

- **altova:elements**("MiElemento", "rip") devuelve MiElemento()*
- **altova:elements**("MiElemento", "pri") devuelve MiElemento()*
- **altova:elements**("MiElemento", "") devuelve MiElemento()*
- **altova:elements**("MiElemento", "Rip") devuelve un error de marca desconocida.
- **altova:elements**("MiElemento",) devuelve un error diciendo que falta el segundo argumento.

find-first [altova:]

altova:find-first((Secuencia *()**)*, **(Condición(Elemento-Secuencia** *como xs:boolean*))
como **item()?** **XP3.1** **XQ3.1**

Esta función toma dos argumentos. El primero es una secuencia de uno o varios elementos de cualquier tipo de datos. El segundo argumento, **Condición**, es una referencia a una función XPath que toma un argumento (es decir, su aridad es 1) y devuelve un valor binario. Cada elemento de **Secuencia** se envía a su vez a la función a la que se hace referencia en **Condición**. Nota: recuerde que esta función solo toma un argumento. El primer elemento de **Secuencia** que consiga que la función de **Condición** dé **true()** como resultado se devuelve como resultado de **find-first** y la iteración se detiene.

Ejemplos

- **altova:find-first**(5 to 10, function($a) {$a mod 2 = 0}) devuelve xs:integer 6
  El argumento **Condición** remite a la función inline XPath 3.0 **function()**, que declara una función inline llamada **$a** y después la define. Cada elemento del argumento Secuencia de **find-first** se envía a su vez como valor de entrada a **$a**. El valor de entrada se prueba en la condición en la definición de función ($a mod 2 = 0). El primer valor de entrada que cumpla la condición se devuelve como resultado de **find-first** (en este caso **6**).

- **altova:find-first**((1 to 10), (function($a) {$a+3=7})) devuelve xs:integer 4

Más ejemplos

Si existe el archivo **C:\Temp\Customers.xml**:

- **altova:find-first**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) devuelve xs:string C:\Temp\Customers.xml

Si no existe el archivo **C:\Temp\Customers.xml** pero existe **http://www.altova.com/index.html**:

- **altova:find-first**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) devuelve xs:string http://www.altova.com/index.html

Si no existe el archivo **C:\Temp\Customers.xml** y tampoco existe **http://www.altova.com/index.html**:

- **altova:find-first**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) no devuelve ningún resultado

© 2016-2022 Altova GmbH

Notas sobre los ejemplos anteriores

- La función XPath 3.0 `doc-available` toma un solo argumento de cadena, que se usa como URI, y devuelve `true` si en el URI dado se encuentra un nodo de documento. El documento que está en el URI dado debe ser un documento XML.
- La función `doc-available` se puede usar para **Condición**, el segundo argumento de `find-first`, porque solamente toma un argumento (aridad=1), porque toma un `item()` como entrada (una cadena que se usa como URI) y devuelve un valor binario.
- Recuerde que solamente se hace referencia a la función `doc-available` pero no se le llama. El sufijo `#1` que se anexa a la función indica una función cuya aridad es 1. Es decir, `doc-available#1` simplemente significa "*Utilizar la función doc-availabe() que tiene aridad=1, pasándole como solo argumento a su vez cada uno de los elementos de la primera secuencia.*" Como resultado, se pasarán las dos cadenas a **doc-available()**, que utiliza la cadena como URI y prueba si existe un nodo de documento en el URI. Si existe, entonces `doc-available()` da como resultado `true()` y esa cadena se devuelve como resultado de la función **find-first**. Nota sobre la función `doc-available()`: las rutas de acceso relativas se resuelven en relación al URI base actual, que es por defecto el URI del documento XML desde el que se carga la función.

▼ find-first-combination [altova:]

altova:find-first-combination((Sec-01 *como item()**)**,* **(Sec-02** *como item()**)**,* **(Condición(Elem-Sec-01, Elem-Sec-02** *como xs:boolean***))** como **item()*** **XP3.1** **XQ3.1**

Esta función toma tres argumentos:

- Los dos primeros (**Sec-01** y **Sec-02**) son secuencias de uno o más elementos de cualquier tipo de datos.
- El tercero (**Condición**) es una referencia a una función XPath que toma dos argumentos (su aridad es 2) y devuelve un valor binario.

Los elementos de **Sec-01** y **Sec-02** se pasan en pares ordenados (cada par está formado por un elemento de cada secuencia) como argumentos de la función de **Condición**. Los pares se ordenan de la siguiente manera:

```
Si   Sec-01 = X1, X2, X3 ... Xn
Y  Sec-02 = Y1, Y2, Y3 ... Yn
Entonces (X1 Y1), (X1 Y2), (X1 Y3) ... (X1 Yn), (X2 Y1), (X2 Y2) ... (Xn Yn)
```

El primer par ordenado que consiga que la función de **Condición** dé como resultado **true()** se devuelve como resultado de **find-first-combination**. Recuerde que (i) si la función de **Condición** recorre los pares de argumentos dados y no consigue dar **true()** como resultado ni una vez, entonces **find-first-combination** devuelve *Sin resultados;* (ii) el resultado de **find-first-combination** siempre será un par de elementos (de cualquier tipo de datos) o ningún elemento.

Ejemplos

- **altova:find-first-pair**(11 to 20, 21 to 30, function($a, $b) {$a+$b = 32}) devuelve la secuencia de `xs:integers (11, 21)`
- **altova:find-first-pair**(11 to 20, 21 to 30, function($a, $b) {$a+$b = 33}) devuelve la secuencia de `xs:integers (11, 22)`
- **altova:find-first-pair**(11 to 20, 21 to 30, function($a, $b) {$a+$b = 34}) devuelve

la secuencia de `xs:integers (11, 23)`

▼ find-first-pair [altova:]

altova:find-first-pair((Sec-01 *como item()**)**, (Sec-02** *como item()**)**, (Condición(Elem-Sec-01, Elem-Sec-02** *como xs:boolean*)) como **item()*** **XP3.1** **XQ3.1**
Esta función toma tres argumentos:

- Los dos primeros (**Sec-01** y **Sec-02**) son secuencias de uno o más elementos de cualquier tipo de datos.
- El tercero (**Condición**) es una referencia a una función XPath que toma dos argumentos (su aridad es 2) y devuelve un valor binario.

Los elementos de **Sec-01** y **Sec-02** se pasan en pares ordenados como argumentos de la función de **Condición**. Los pares se ordenan de la siguiente manera:

```
Si   Sec-01 = X1, X2, X3 ... Xn
Y  Sec-02 = Y1, Y2, Y3 ... Yn
Entonces (X1 Y1), (X2 Y2), (X3 Y3) ... (Xn Yn)
```

El primer par ordenado que consiga que la función de **Condición** dé como resultado **true()** se devuelve como resultado de **find-first-pair**. Recuerde que (i) si la función de **Condición** recorre los pares de argumentos dados y no consigue dar **true()** como resultado ni una vez, entonces **find-first-pair** devuelve *Sin resultados;* (ii) el resultado de **find-first-pair** siempre será un par de elementos (de cualquier tipo de datos) o ningún elemento.

▣ *Ejemplos*

- **altova:find-first-pair**`(11 to 20, 21 to 30, function($a, $b) {$a+$b = 32})` devuelve la secuencia de `xs:integers (11, 21)`
- **altova:find-first-pair**`(11 to 20, 21 to 30, function($a, $b) {$a+$b = 33})` devuelve *Sin resultados*

Observe que en los dos ejemplos anteriores el orden de los pares es: `(11, 21) (12, 22) (13, 23)...(20, 30)`. Por ese motivo el segundo ejemplo no obtiene resultados (porque ningún par ordenado consigue sumar `33`).

▼ find-first-pair-pos [altova:]

altova:find-first-pair-pos((Sec-01 *como item()**)**, (Sec-02** *como item()**)**, (Condición(Elem-Sec-01, Elem-Sec-02** *como xs:boolean*)) como **xs:integer** **XP3.1** **XQ3.1**
Esta función toma tres argumentos:

- Los dos primeros (**Sec-01** y **Sec-02**) son secuencias de uno o más elementos de cualquier tipo de datos.
- El tercero (**Condición**) es una referencia a una función XPath que toma dos argumentos (su aridad es 2) y devuelve un valor binario.

Los elementos de **Sec-01** y **Sec-02** se pasan en pares ordenados como argumentos de la función de **Condición**. Los pares se ordenan de la siguiente manera:

© 2016-2022 Altova GmbH

```
Si   Sec-01 = X1, X2, X3 ... Xn
Y  Sec-02 = Y1, Y2, Y3 ... Yn
Entonces (X1 Y1), (X2 Y2), (X3 Y3) ... (Xn Yn)
```

La posición de índice del primer par ordenado que consiga que la función de **Condición** dé como resultado **true()** se devuelve como resultado de **find-first-pair-pos**. Recuerde que si la función de **Condición** recorre los pares de argumentos dados y no da como resultado **true()** ni una sola vez, entonces **find-first-pair-pos** devuelve *Sin resultados*.

Ejemplos

- **altova:find-first-pair**(11 to 20, 21 to 30, function($a, $b) {$a+$b = 32}) devuelve 1
- **altova:find-first-pair**(11 to 20, 21 to 30, function($a, $b) {$a+$b = 33}) devuelve *Sin resultados*

Observe que en los dos ejemplos anteriores el orden de los pares es: (11, 21) (12, 22) (13, 23)...(20, 30). En el primer ejemplo el primer par consigue que la función de **Condición** dé como resultado **true()** y, por tanto, se devuelve la posición de índice que tienen en la secuencia (**1**). El segundo ejemplo, sin embargo, devuelve *Sin resultados* porque ningún par consigue sumar 33.

find-first-pos [altova:]

altova:find-first-pos((Secuencia *como item()**)**,* **(Condición(Elem-Sec** *como xs:boolean***))** como **xs:integer** **XP3.1** **XQ3.1**

Esta función toma dos argumentos. El primer argumento es una secuencia de uno o varios elementos de cualquier tipo. El segundo argumento (**Condición**) es una referencia a una función XPath que toma un argumento (su aridad es 1) y devuelve un valor binario. Cada elemento de **Secuencia** se envía a su vez a la función a la que se hace referencia en **Condición**. (Recuerde que esta función toma un solo argumento.) El primer elemento de **Secuencia** que consiga que la función de **Condición** dé como resultado **true()** devuelve la posición de índice que tiene en **Secuencia** como resultado de **find-first-pos** y la iteración se detiene.

Ejemplos

- **altova:find-first-pos**(5 to 10, function($a) {$a mod 2 = 0}) devuelve xs:integer 2

El argumento **Condición** hace referencia a la función inline XPath 3.0 **function()**, que declara una función inline llamada **$a** y después la define. Cada elemento del argumento **Sequence** de **find-first-pos** se pasa a su vez como valor de entrada de **$a**. El valor de entrada se prueba en la condición de la definición de función ($a mod 2 = 0). La posición de índice que tiene en la secuencia el primer valor de entrada que cumple la condición se devuelve como resultado de **find-first-pos** (en este caso es la posición de índice **2**, porque **6** es el primer valor (de la secuencia) que cumple la condición y su posición de índice en la secuencia es **2**).

- **altova:find-first-pos**((2 to 10), (function($a) {$a+3=7})) devuelve xs:integer 3

Más ejemplos

Si existe el archivo **C:\Temp\Customers.xml**:

- **altova:find-first-pos**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) devuelve 1

Si no existe el archivo **C:\Temp\Customers.xml** pero existe **http://www.altova.com/index.html**:

- **altova:find-first-pos**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) devuelve 2

Si no existe el archivo **C:\Temp\Customers.xml** y tampoco existe **http://www.altova.com/index.html**:

- **altova:find-first-pos**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) no devuelve ningún resultado

Notas sobre los ejemplos anteriores

- La función XPath 3.0 doc-available toma un solo argumento de cadena, que se usa como URI, y devuelve true si en el URI dado se encuentra un nodo de documento. El documento que está en el URI dado debe ser un documento XML.
- La función doc-available se puede usar para **Condición**, el segundo argumento de find-first-pos, porque solamente toma un argumento (aridad=1), porque toma un item() como entrada (una cadena que se usa como URI) y devuelve un valor binario.
- Recuerde que solamente se hace referencia a la función doc-available pero no se le llama. El sufijo #1 que se anexa a la función indica una función cuya aridad es 1. Es decir, doc-available#1 simplemente significa "*Utilizar la función doc-availabe() que tiene aridad=1, pasándole como solo argumento a su vez cada uno de los elementos de la primera secuencia.*" Como resultado, se pasarán las dos cadenas a **doc-available()**, que utiliza la cadena como URI y prueba si existe un nodo de documento en el URI. Si existe, entonces doc-available() da como resultado true() y esa cadena se devuelve como resultado de la función **find-first-pos**. Nota sobre la función doc-available(): las rutas de acceso relativas se resuelven en relación al URI base actual, que es por defecto el URI del documento XML desde el que se carga la función.

for-each-attribute-pair [altova:]

altova:for-each-attribute-pair(Seq1 *como element()?*, **Seq2** *como element()?*, **Function** *como function()*) como **item()*** **XP3.1** **XQ3.1**

Los primeros dos argumentos identifican dos elementos cuyos atributos se usan para construir pares de atributos donde uno de los atributos del par se obtiene del primer elemento y el otro atributo del segundo elemento. Los pares de atributos se seleccionan basándose en que tienen el mismo nombre y se ordenan alfabéticamente por grupos. Si un atributo no tiene un atributo correspondiente en el otro elemento, entonces el par está "desarticulado", lo que significa que tiene un solo miembro. El elemento de la función (tercer argumento Function) se aplica por separado a cada par de la secuencia de pares (articulados y desarticulados) y el resultado es una secuencia de elementos.

Ejemplos

- **altova:for-each-attribute-pair**(/Example/Test-A, /Example/Test-B, function($a, $b) {$a+b}) devuelve...

© 2016-2022 Altova GmbH

`(2, 4, 6)` si
`<Test-A att1="1" att2="2" att3="3" />`
`<Test-B att1="1" att2="2" att3="3" />`

`(2, 4, 6)` si
`<Test-A att2="2" att1="1" att3="3" />`
`<Test-B att3="3" att2="2" att1="1" />`

`(2, 6)` si
`<Test-A att4="4" att1="1" att3="3" />`
`<Test-B att3="3" att2="2" att1="1" />`

Nota: El resultado `(2, 6)` *se obtiene mediante la siguiente acción:* `(1+1, ()+2, 3+3, 4+())`. *Si uno de los operandos es la secuencia vacía, como en el caso de los elementos 2 y 4, entonces el resultado de la suma es una secuencia vacía.*

- **`altova:for-each-attribute-pair`**`(/Example/Test-A, /Example/Test-B, concat#2)` devuelve...

`(11, 22, 33)` si
`<Test-A att1="1" att2="2" att3="3" />`
`<Test-B att1="1" att2="2" att3="3" />`

`(11, 2, 33, 4)` si
`<Test-A att4="4" att1="1" att3="3" />`
`<Test-B att3="3" att2="2" att1="1" />`

for-each-combination [altova:]

`altova:for-each-combination(PrimeraSecuencia` *`como item()*`**`, SegundaSecuencia`** *`como item()*`**`, función($i,$j){$i || $j} )`** como `item()*` **XP3.1** **XQ3.1**

Los elementos de las dos secuencias en los primeros dos argumentos se combinan de forma que el primer elemento de la primera secuencia se combina, en orden, una vez con cada elemento de la segunda secuencia. La función dada como tercer argumento se aplica a cada una de las combinaciones de la secuencia resultante y da como resultado una secuencia de elementos (*véase ejemplo*).

Ejemplos

- **`altova:for-each-combination`**`( ('a', 'b', 'c'), ('1', '2', '3'), function($i, $j) {$i || $j} )` devuelve `('a1', 'a2', 'a3', 'b1', 'b2', 'b3', 'c1', 'c2', 'c3')`

for-each-matching-attribute-pair [altova:]

`altova:for-each-matching-attribute-pair(Seq1` *`como element()?`*`, `**`Seq2`** *`como element()?`*`, `**`Function`** *`como function()`*`)` como `item()*` **XP3.1** **XQ3.1**

Los primeros dos argumentos identifican dos elementos cuyos atributos se usan para construir pares de atributos donde un atributo de cada par se obtiene del primer elemento y el otro atributo del par se obtiene del segundo elemento. Los pares de elementos se seleccionan basándose en que tienen el mismo nombre y se ordenan alfabéticamente por grupos. Si un atributo no tiene un atributo correspondiente en el otro elemento, entonces no se construye ningún par. El elemento de la función (tercer argumento

Function) se aplica por separado a cada par de la secuencia de pares (articulados y desarticulados) y el resultado es una secuencia de elementos.

Ejemplos

- **altova:for-each-matching-attribute-pair**(/Example/Test-A, /Example/Test-B, function($a, $b){$a+b}) devuelve...

```
(2, 4, 6) if
<Test-A att1="1" att2="2" att3="3" />
<Test-B att1="1" att2="2" att3="3" />

(2, 4, 6) if
<Test-A att2="2" att1="1" att3="3" />
<Test-B att3="3" att2="2" att1="1" />

(2, 6) if
<Test-A att4="4" att1="1" att3="3" />
<Test-B att3="3" att2="2" att3="1" />
```

- **altova:for-each-matching-attribute-pair**(/Example/Test-A, /Example/Test-B, concat#2) devuelve...

```
(11, 22, 33) if
<Test-A att1="1" att2="2" att3="3" />
<Test-B att1="1" att2="2" att3="3" />

(11, 33) if
<Test-A att4="4" att1="1" att3="3" />
<Test-B att3="3" att2="2" att1="1" />
```

▼ substitute-empty [altova:]

altova:substitute-empty(PrimeraSecuencia *as item()**, **SegundaSecuencia** *as item()***)** como **item()*** XP3.1 XQ3.1

Si **PrimeraSecuencia** está vacío, la función devuelve **SegundaSecuencia**. Si **PrimeraSecuencia** no está vacío, la función devuelve **PrimeraSecuencia**.

Ejemplos

- **altova:substitute-empty**((1,2,3), (4,5,6)) devuelve (1,2,3)
- **altova:substitute-empty**((), (4,5,6)) devuelve (4,5,6)

30.2.1.8 Funciones XPath/XQuery: de cadena

Las funciones de extensión de Altova para trabajar con cadenas pueden utilizarse en expresiones XPath y XQuery y ofrecen funciones adicionales para el procesamiento de datos. Estas funciones se pueden usar con los motores **XPath 3.0** y **XQuery 3.0** de Altova. Están disponibles en contextos XPath/XQuery.

© 2016-2022 Altova GmbH

Nota sobre el nombre de las funciones y lenguajes

Puede utilizar todas las funciones de extensión de Altova en sus expresiones XPath/XQuery. Con ellas conseguirá funciones adicionales no disponibles en la biblioteca de funciones estándar de XPath, XQuery y XSLT. Las funciones de extensión de Altova están en el **espacio de nombres** **http://www.altova.com/xslt-extensions** y en esta sección se presentan con el prefijo, que se supone estará enlazado al espacio de nombres señalado. Tenga en cuenta que en futuras versiones del producto algunas funciones pueden dejar de ser compatibles o su comportamiento puede cambiar. Por tanto, consulte siempre la documentación del producto para conocer el funcionamiento de estas funciones en cada versión del producto.

Funciones XPath (en expresiones XPath en XSLT):	XP1 XP2 XP3.1.1
Funciones XSLT (en expresiones XPath en XSLT):	XSLT1 XSLT2 XSLT3
Funciones XQuery (en expresiones XQuery en XQuery):	XQ1 XQ3.1

camel-case [altova:]

altova:camel-case(CadenaEntrada *como xs:string***)** como **xs:string** XP3.1 XQ3.1
Devuelve la cadena de entrada **CadenaEntrada** escrita en CamelCase. La cadena se analiza usando la expresión regular **'\s'** (que es la forma abreviada del carácter espacio en blanco). El primer carácter que no sea un espacio en blanco situado después de un espacio en blanco o de una secuencia de espacios en blanco consecutivos se pondrá en mayúsculas. El primer carácter de la cadena de salida se pondrá en mayúsculas.

Ejemplos

- **altova:camel-case**("max") devuelve Max
- **altova:camel-case**("max max") devuelve Max Max
- **altova:camel-case**("file01.xml") devuelve File01.xml
- **altova:camel-case**("file01.xml file02.xml") devuelve File01.xml File02.xml
- **altova:camel-case**("file01.xml file02.xml") devuelve File01.xml File02.xml
- **altova:camel-case**("file01.xml -file02.xml") devuelve File01.xml -file02.xml

altova:camel-case(CadenaEntrada *como xs:string*, **CaracteresDivisión** *como xs:string*, **EsExpReg** *como xs:boolean***)** como **xs:string** XP3.1 XQ3.1
Devuelve la cadena de entrada **CadenaEntrada** escrita en CamelCase usando los **CaracteresDivisón** para determinar qué caracteres desencadenan el siguiente uso de mayúsculas. El argumento **CaracteresDivisón** se usa como expresión regular **cuando EsExpReg = true()** o como caracteres planos cuando r **EsExpReg = false()**. El primer carácter de la cadensa de salida se escribe con mayúsculas.

Ejemplos

- **altova:camel-case**("setname getname", "set|get", true()) devuelve setName getName
- **altova:camel-case**("altova\documents\testcases", "\", false()) devuelve Altova\Documents\Testcases

▼ char [altova:]

altova:char(Posición *as xs:integer***)** como **xs:string** XP3.1 XQ3.1
Devuelve una cadena que contiene el carácter que está en la posición indicada por el argumento **Posición** en la cadena que se obtiene al convertir el valor del elemento de contexto en `xs:string`. La cadena resultante estará vacía si en la posición indicada no existe ningún carácter.

⊟ *Ejemplos*

Si el elemento de contexto es **1234ABCD**:

- **altova:char**(2) devuelve 2
- **altova:char**(5) devuelve A
- **altova:char**(9) devuelve la cadena vacía
- **altova:char**(-2) devuelve la cadena vacía

altova:char(CadenaEntrada *as xs:string*, **Posición** *as xs:integer***)** como **xs:string** XP3.1 XQ3.1
Devuelve una cadena que contiene el carácter que está en la posición indicada por el argumento **Posición** en la cadena dada por el argumento **CadenaEntrada**. La cadena resultante estará vacía si en la posición indicada no existe ningún carácter.

⊟ *Ejemplos*

- **altova:char**("2014-01-15", 5) devuelve -
- **altova:char**("USA", 1) devuelve U
- **altova:char**("USA", 1) devuelve la cadena vacía
- **altova:char**("USA", -2) devuelve la cadena vacía

▼ create-hash-from-string[altova:]

altova:create-hash-from-string(InputString *como xs:string***)** como **xs:string** XP2 XQ1 XP3.1 XQ3.1
altova:create-hash-from-string(InputString *como xs:string*, **HashAlgo** *as xs:string***)** como **xs:string** XP2 XQ1 XP3.1 XQ3.1
Genera una cadena hash a partir de InputString usando el algoritmo de hash especificado por el argumento `HashAlgo`. Se pueden usar los siguientes algoritmos de hash (en mayúsculas o minúsculas): **MD5**, **SHA-1**, **SHA-224**, **SHA-256**, **SHA-384**, **SHA-512**. Si no se especifica el segundo argumento (*véase la primera instrucción*) se usa el algoritmo de hash **SHA-256**.

⊞ *Ejemplos*

- **altova:create-hash-from-string**('abc') devuelve una cadena hash generada usando el algoritmo de hash **SHA-256**.
- **altova:create-hash-from-string**('abc', 'md5') devuelve una cadena hash generada usando el algoritmo de hash **MD5**.
- **altova:create-hash-from-string**('abc', 'MD5') devuelve una cadena hash generada usando el algoritmo de hash **MD5**.

© 2016-2022 Altova GmbH

▼ first-chars [altova:]

altova:first-chars(X *as xs:integer*) como **xs:string** XP3.1 XQ3.1
Devuelve una cadena que contiene los **X** primeros caracteres de la cadena que se obtiene al convertir el valor del elemento de contexto en `xs:string`.

⊟ *Ejemplos*

Si el elemento de contexto es **1234ABCD**:

- **altova:first-chars**(2) devuelve 12
- **altova:first-chars**(5) devuelve 1234A
- **altova:first-chars**(9) devuelve 1234ABCD

altova:first-chars(CadenaEntrada *as xs:string*, **X** *as xs:integer*) como **xs:string** XP3.1 XQ3.1
Devuelve una cadena que contiene los **X** primeros caracteres de la cadena dada como argumento **CadenaEntrada**.

⊟ *Ejemplos*

- **altova:first-chars**("2014-01-15", 5) devuelve 2014-
- **altova:first-chars**("USA", 1) devuelve U

▼ format-string [altova:]

altova:format-string(InputString *como xs:string*, **FormatSequence** *como item()**) como **xs:string** XP3.1 XQ3.1
La cadena de entrada (primer argumento) contiene parámetros posicionales (%1, %2, etc). Cada parámetro es reemplazado por el elemento cadena ubicado en la posición correspondiente de la secuencia de formato (enviada como segundo argumento). Por tanto, el primer elemento de la secuencia de formato reemplaza al parámetro posicional %1, el segundo elemento reemplaza a %2 y así sucesivamente. La función devuelve esta secuencia con formato que contiene los elementos de reemplazo. Si no existe una cadena para alguno de los parámetros posicionales, entonces se devuelve ese mismo parámetro posicional. Esto ocurre cuando el índice de un parámetro posicional es mayor que el número de elementos de la secuencia de formato.

⊞ *Ejemplos*

- **altova:format-string**('Hello %1, %2, %3', ('Jane','John','Joe')) devuelve "Hello Jane, John, Joe"
- **altova:format-string**('Hello %1, %2, %3', ('Jane','John','Joe', 'Tom')) devuelve "Hello Jane, John, Joe"
- **altova:format-string**('Hello %1, %2, %4', ('Jane','John','Joe', 'Tom')) devuelve "Hello Jane, John, Tom"
- **altova:format-string**('Hello %1, %2, %4', ('Jane','John','Joe')) devuelve "Hello Jane, John, %4"

▼ last-chars [altova:]

altova:last-chars(X *as xs:integer*) como **xs:string** XP3.1 XQ3.1
Devuelve una cadena que contiene los X últimos caracteres de la cadena que se obtiene al convertir el valor del elemento de contexto en `xs:string`.

Ejemplos

Si el elemento de contexto es **1234ABCD**:

- **altova:last-chars**(2) devuelve CD
- **altova:last-chars**(5) devuelve 4ABCD
- **altova:last-chars**(9) devuelve 1234ABCD

altova:last-chars(CadenaEntrada *as xs:string*, **X** *as xs:integer*) como **xs:string** XP3.1 XQ3.1
Devuelve una cadena que contiene los **X** últimos caracteres de la cadena dada como argumento **CadenaEntrada**.

Ejemplos

- **altova:last-chars**("2014-01-15", 5) devuelve 01-15-
- **altova:last-chars**("USA", 10) devuelve USA

pad-string-left [altova:]

altova:pad-string-left(CadenaParaRellenar *como xs:string*, **LongitudCadena** *como xs:integer*, **CarácterRelleno** *como xs:string*) como **xs:string** XP3.1 XQ3.1
El argumento CarácterRelleno es un solo carácter. Se añade a la izquierda de la cadena para aumentar el número de caracteres de la CadenaParaRellenar, de modo que este número equivalga al valor entero del argumento LongitudCadena. El argumento LongitudCadena puede tener cualquier valor entero (positivo o negativo), pero el relleno solo se lleva a cabo si el valor de LongitudCadena es mayor que el número de caracteres de CadenaParaRellenar. Si CadenaParaRellenar tiene más caracteres que el valor de LongitudCadena, entonces CadenaParaRellenar se deja como está.

Ejemplos

- **altova:pad-string-left**('AP', 1, 'Z') devuelve 'AP'
- **altova:pad-string-left**('AP', 2, 'Z') devuelve 'AP'
- **altova:pad-string-left**('AP', 3, 'Z') devuelve 'ZAP'
- **altova:pad-string-left**('AP', 4, 'Z') devuelve 'ZZAP'
- **altova:pad-string-left**('AP', -3, 'Z') devuelve 'AP'
- **altova:pad-string-left**('AP', 3, 'YZ') devuelve un error indicando que el carácter de relleno es demasiado largo.

pad-string-right [altova:]

altova:pad-string-right(CadenaParaRellenar *como xs:string*, **LongitudCadena** *como xs:integer*, **CarácterRelleno** *como xs:string*) como **xs:string** XP3.1 XQ3.1
El argumento CarácterRelleno es un solo carácter. Se añade a la derecha de la cadena para aumentar el número de caracteres de la CadenaParaRellenar, de modo que este número equivalga al valor entero del argumento LongitudCadena. El argumento LongitudCadena puede tener cualquier valor entero (positivo o negativo), pero el relleno solo se lleva a cabo si el valor de LongitudCadena es mayor que el número de caracteres de CadenaParaRellenar. Si CadenaParaRellenar tiene más caracteres que el valor de LongitudCadena, entonces CadenaParaRellenar se deja como está.

© 2016-2022 Altova GmbH

Ejemplos

- `altova:pad-string-right('AP', 1, 'Z')` devuelve `'AP'`
- `altova:pad-string-right('AP', 2, 'Z')` devuelve `'AP'`
- `altova:pad-string-right('AP', 3, 'Z')` devuelve `'APZ'`
- `altova:pad-string-right('AP', 4, 'Z')` devuelve `'APZZ'`
- `altova:pad-string-right('AP', -3, 'Z')` devuelve `'AP'`
- `altova:pad-string-right('AP', 3, 'YZ')` devuelve un error indicando que el carácter de relleno es demasiado largo.

repeat-string [altova:]

altova:repeat-string(CadenaEntrada *as xs:string*, **Repeticiones** *as xs:integer*) como **xs:string** **XP2** **XQ1** **XP3.1** **XQ3.1**

Genera una cadena que está compuesta por el primer argumento `CadenaEntrada` repetida tantas veces como indique el argumento `Repeticiones`.

Ejemplo

- **altova:repeat-string**("Altova #", 3)
  devuelve `Altova #Altova #Altova #`"

substring-after-last [altova:]

altova:substring-after-last(CadenaPrincipal *as xs:string*, **CadenaPrueba** *as xs:string*) como **xs:string** **XP3.1** **XQ3.1**

Si **CadenaPrueba** se encuentra en **CadenaPrincipal**, la función devuelve la subcadena que aparece después de **CadenaPrueba** en **CadenaPrincipal**. Si **CadenaPrueba** no está en **CadenaPrincipal**, entonces devuelve la cadena vacía. Si **CadenaPrueba** es una cadena vacía, entonces devuelve la **CadenaPrincipal** entera. Si **CadenaPrueba** aparece varias veces en **CadenaPrincipal**, la función devuelve la subcadena que aparece después de la última **CadenaPrueba**.

Ejemplos

- `altova:substring-after-last('ABCDEFGH', 'B')` devuelve `'CDEFGH'`
- `altova:substring-after-last('ABCDEFGH', 'BC')` devuelve `'DEFGH'`
- `altova:substring-after-last('ABCDEFGH', 'BD')` devuelve `''`
- `altova:substring-after-last('ABCDEFGH', 'Z')` devuelve `''`
- `altova:substring-after-last('ABCDEFGH', '')` devuelve `'ABCDEFGH'`
- `altova:substring-after-last('ABCD-ABCD', 'B')` devuelve `'CD'`
- `altova:substring-after-last('ABCD-ABCD-ABCD', 'BCD')` devuelve `''`

substring-before-last [altova:]

altova:substring-before-last(CadenaPrincipal *as xs:string*, **CadenaPrueba** *as xs:string*) como **xs:string** **XP3.1** **XQ3.1**

Si **CadenaPrueba** se encuentra en **CadenaPrincipal**, la función devuelve la subcadena que aparece después de **CadenaPrueba** en **CadenaPrincipal**. Si **CadenaPrueba** no está en **CadenaPrincipal**, entonces devuelve la cadena vacía. Si **CadenaPrueba** es una cadena vacía, entonces devuelve la **CadenaPrincipal** entera. Si **CadenaPrueba** aparece varias veces en **CadenaPrincipal**, la función

devuelve la subcadena que aparece antes de la última **CadenaPrueba**.

Ejemplos

- **altova:substring-before-last**('ABCDEFGH', 'B') devuelve 'A'
- **altova:substring-before-last**('ABCDEFGH', 'BC') devuelve 'A'
- **altova:substring-before-last**('ABCDEFGH', 'BD') devuelve ''
- **altova:substring-before-last**('ABCDEFGH', 'Z') devuelve ''
- **altova:substring-before-last**('ABCDEFGH', '') devuelve ''
- **altova:substring-before-last**('ABCD-ABCD', 'B') devuelve 'ABCD-A'
- **altova:substring-before-last**('ABCD-ABCD-ABCD', 'ABCD') devuelve 'ABCD-ABCD-'

substring-pos [altova:]

altova:substring-pos(Cadena *as xs:string*, **CadenaBúsqueda** *as xs:string***)** como **xs:integer** **XP3.1** **XQ3.1**

Devuelve la posición de carácter de la primera instancia de **CadenaBúsqueda** en **Cadena**. La posición de carácter se devuelve como número entero. El primer carácter de **CadenaBúsqueda** tiene la posición 1. Si **CadenaBúsqueda** no aparece dentro de **Cadena**, la función devuelve el entero 0. Para buscar la segunda instancia de **CadenaBúsqueda**, etc. use la otra firma de esta función.

Ejemplos

- **altova:substring-pos**('Altova', 'to') devuelve 3
- **altova:substring-pos**('Altova', 'tov') devuelve 3
- **altova:substring-pos**('Altova', 'tv') devuelve 0
- **altova:substring-pos**('AltovaAltova', 'to') devuelve 3

altova:substring-pos(Cadena *as xs:string*, **CadenaBúsqueda** *as xs:string*, **Entero** *as xs:integer***)** como **xs:integer** **XP3.1** **XQ3.1**

Devuelve la posición de carácter de **CadenaBúsqueda** en **Cadena**. La búsqueda de **CadenaBúsqueda** empieza en la posición de carácter dada por el argumento **Entero** (es decir, no se busca en la subcadena anterior a esta posición). El entero devuelto, sin embargo, es la posición que la cadena encontrada tiene en **Cadena**. Esta firma es muy práctica si quiere buscar la segunda posición, etc. de una cadena que aparece varias veces dentro de **Cadena**. Si **CadenaBúsqueda** no aparece en **Cadena**, la función devuelve el entero 0.

Ejemplos

- **altova:substring-pos**('Altova', 'to', 1) devuelve 3
- **altova:substring-pos**('Altova', 'to', 3) devuelve 3
- **altova:substring-pos**('Altova', 'to', 4) devuelve 0
- **altova:substring-pos**('Altova-Altova', 'to', 0) devuelve 3
- **altova:substring-pos**('Altova-Altova', 'to', 4) devuelve 10

substring-pos [altova:]

altova:substring-pos(Cadena *as xs:string*, **CadenaBúsqueda** *as xs:string***)** como **xs:integer** **XP3.1** **XQ3.1**

Devuelve la posición de carácter de la primera instancia de **CadenaBúsqueda** en **Cadena**. La posición de carácter se devuelve como número entero. El primer carácter de **CadenaBúsqueda** tiene la posición 1. Si **CadenaBúsqueda** no aparece dentro de **Cadena**, la función devuelve el entero 0. Para buscar la segunda instancia de **CadenaBúsqueda**, etc. use la otra firma de esta función.

© 2016-2022 Altova GmbH

Ejemplos

- **altova:substring-pos**('Altova', 'to') devuelve 3
- **altova:substring-pos**('Altova', 'tov') devuelve 3
- **altova:substring-pos**('Altova', 'tv') devuelve 0
- **altova:substring-pos**('AltovaAltova', 'to') devuelve 3

altova:substring-pos(Cadena *as xs:string*, **CadenaBúsqueda** *as xs:string*, **Entero** *as xs:integer*) como **xs:integer** XP3.1 XQ3.1
Devuelve la posición de carácter de **CadenaBúsqueda** en **Cadena**. La búsqueda de **CadenaBúsqueda** empieza en la posición de carácter dada por el argumento **Entero** (es decir, no se busca en la subcadena anterior a esta posición). El entero devuelto, sin embargo, es la posición que la cadena encontrada tiene en **Cadena**. Esta firma es muy práctica si quiere buscar la segunda posición, etc. de una cadena que aparece varias veces dentro de **Cadena**. Si **CadenaBúsqueda** no aparece en **Cadena**, la función devuelve el entero 0.

Ejemplos

- **altova:substring-pos**('Altova', 'to', 1) devuelve 3
- **altova:substring-pos**('Altova', 'to', 3) devuelve 3
- **altova:substring-pos**('Altova', 'to', 4) devuelve 0
- **altova:substring-pos**('Altova-Altova', 'to', 0) devuelve 3
- **altova:substring-pos**('Altova-Altova', 'to', 4) devuelve 10

trim-string [altova:]

altova:trim-string(CadenaEntrada *as xs:string*) como **xs:string** XP3.1 XQ3.1
Esta función toma un argumento xs:string, quita los espacios en blanco iniciales y finales y devuelve un xs:string "recortado".

Ejemplos

- **altova:trim-string**(" Hello World ")) devuelve "Hello World"
- **altova:trim-string**("Hello World ")) devuelve "Hello World"
- **altova:trim-string**(" Hello World")) devuelve "Hello World"
- **altova:trim-string**("Hello World")) devuelve "Hello World"
- **altova:trim-string**("Hello World")) devuelve "Hello World"

trim-string-left [altova:]

altova:trim-string-left(CadenaEntrada *as xs:string*) como **xs:string** XP3.1 XQ3.1
Esta función toma un argumento xs:string, quita los espacios en blanco iniciales y devuelve un xs:string recortado por la izquierda.

Ejemplos

- **altova:trim-string-left**(" Hello World ")) devuelve "Hello World "
- **altova:trim-string-left**("Hello World ")) devuelve "Hello World "
- **altova:trim-string-left**(" Hello World")) devuelve "Hello World"
- **altova:trim-string-left**("Hello World")) devuelve "Hello World"
- **altova:trim-string-left**("Hello World")) devuelve "Hello World"

▼ trim-string-right [altova:]

altova:trim-string-right(CadenaEntrada *as xs:string***)** como **xs:string** XP3.1 XQ3.1
Esta función toma un argumento `xs:string`, quita los espacios en blanco finales y devuelve una cadena `xs:string` recortada por la derecha.

⊟ *Ejemplos*

- **altova:trim-string-right**(" Hello World ")) devuelve " Hello World"
- **altova:trim-string-right**("Hello World ")) devuelve "Hello World"
- **altova:trim-string-right**(" Hello World")) devuelve " Hello World"
- **altova:trim-string-right**("Hello World")) devuelve "Hello World"
- **altova:trim-string-right**("Hello World")) devuelve "Hello World"

30.2.1.9 Funciones XPath/XQuery varias

Estas funciones de extensión XPath/XQuery generales son compatibles con la versión actual de XMLSpy y se pueden usar en (i) expresiones XPath en contextos XSLT o (ii) en expresiones XQuery en documentos XQuery.

Nota sobre el nombre de las funciones y lenguajes

Puede utilizar todas las funciones de extensión de Altova en sus expresiones XPath/XQuery. Con ellas conseguirá funciones adicionales no disponibles en la biblioteca de funciones estándar de XPath, XQuery y XSLT. Las funciones de extensión de Altova están en el **espacio de nombres** **http://www.altova.com/xslt-extensions** y en esta sección se presentan con el prefijo, que se supone estará enlazado al espacio de nombres señalado. Tenga en cuenta que en futuras versiones del producto algunas funciones pueden dejar de ser compatibles o su comportamiento puede cambiar. Por tanto, consulte siempre la documentación del producto para conocer el funcionamiento de estas funciones en cada versión del producto.

Funciones XPath (en expresiones XPath en XSLT):	XP1 XP2 XP3.1.1
Funciones XSLT (en expresiones XPath en XSLT):	XSLT1 XSLT2 XSLT3
Funciones XQuery (en expresiones XQuery en XQuery):	XQ1 XQ3.1

▼ Clic para expandir/contraer

altova:decode-string(Input *as xs:base64Binary***) como xs:string** XP3.1 XQ3.1
altova:decode-string(Input *as xs:base64Binary***, Encoding como** *xs:string***) como xs:string** XP3.1 XQ3.1
Descifra la entrada en base64Binary en una cadena con el cifrado que se indique. Si no se indica ninguno se usa UTF-8. Estos son los cifrados compatibles: `US-ASCII, ISO-8859-1, UTF-16, UTF-16LE, UTF-16BE, ISO-10646-UCS2, UTF-32, UTF-32LE, UTF-32BE, ISO-10646-UCS4`

⊟ *Ejemplos*

- **altova:decode-string**($XML1/MailData/Meta/b64B) devuelve la entrada en base64Binary

© 2016-2022 Altova GmbH

como cadena de texto cifrada en UTF-8

- `altova:decode-string($XML1/MailData/Meta/b64B, "UTF-8")` devuelve la entrada en base64Binary como cadena de texto cifrada en UTF-8
- `altova:decode-string($XML1/MailData/Meta/b64B, "ISO-8859-1")` devuelve la entrada en base64Binary como una cadena de texto cifrada en ISO-8859-1

encode-string [altova:]

altova:encode-string(InputString *como xs:string***) como xs:base64Binaryinteger** XP3.1 XQ3.1
altova:encode-string(InputString *como xs:string*, **Encoding** *como xs:string***) como xs:base64Binaryinteger** XP3.1 XQ3.1
Cifra una cadena de texto usando el cifrado que se indique. Si no se indica ninguno, entonces se usa UTF-8. La cadena cifrada se convierte en caracteres base64Binary y se devuelve el valor base64Binary convertido. De momento se admite UTF-8, pero ampliaremos la compatibilidad a: `US-ASCII, ISO-8859-1, UTF-16, UTF-16LE, UTF-16BE, ISO-10646-UCS2, UTF-32, UTF-32LE, UTF-32BE, ISO-10646-UCS4`

Ejemplos

- `altova:encode-string("Altova")` devuelve el equivalente en base64Binary de la cadena de texto cifrada en UTF-8 `"Altova"`
- `altova:encode-string("Altova", "UTF-8")` devuelve el equivalente en base64Binary de la cadena de texto cifrada en UTF-8 `"Altova"`

get-temp-folder [altova:]

altova:get-temp-folder() como `xs:string` XP2 XQ1 XP3.1 XQ3.1
Esta función no toma ningún argumento. Devuelve la ruta de acceso de la carpeta temporal del usuario actual.

Ejemplo

- `altova:get-temp-folder()`
  en un equipo Windows devuelve (más o menos) `C:\Usuarios\<Usuario>\AppData\Local\Temp\` como valor de tipo `xs:string`.

generate-guid [altova:]

altova:generate-guid() asxs:string XP2 XQ1 XP3.1 XQ3.1
Genera una cadena única de la interfaz gráfica del usuario.

Ejemplo

- `altova:generate-guid()` devuelve (por ejemplo) `85F971DA-17F3-4E4E-994E-99137873ACCD`

high-res-timer [altova:]

altova:high-res-timer() como **xs:double** XP3.1 XQ3.1
Devuelve un valor de temporizador de alta resolución en segundos. La presencia de un temporizador de alta resolución en un sistema permite hacer mediciones de alta precisión si es necesario (por ejemplo, en

animaciones y para precisar de forma exacta horas de ejecución de código). Esta función ofrece la resolución del temporizador de alta resolución del sistema.

Ejemplos

- `altova:high-res-timer()` devuelve algo como `'1.16766146154566E6'`

parse-html [altova:]

`altova:parse-html(HTMLText` *`as xs:string`***`)`** como **`node()`** XP3.1 XQ3.1
El argumento `HTMLText` es una cadena que contiene el texto de un documento HTML. La función crea una estructura HTML a partir de la cadena. La cadena enviada puede contener o no el elemento HTML. En ambos casos el elemento raíz de la estructura es un elemento llamado **`HTML`**. Asegúrese de que el código HTML de la cadena enviada es válido.

Ejemplos

- `altova:parse-html("<html><head/><body><h1>Header</h1></body></html>")` crea una estructura HTML a partir de la cadena enviada

sleep[altova:]

`altova:sleep(Millisecs` *`como xs:integer`***`)`** como **`empty-sequence()`** XP2 XQ1 XP3.1 XQ3.1
Suspende la ejecución de la operación actual durante el número de milisegundos dado por el argumento `Millisecs`.

Ejemplos

- `altova:sleep(1000)` suspende la ejecución de la operación actual durante 1000 millisegundos.

[Subir 2198]

30.2.1.10 Funciones de extensión para gráficos

Las funciones para gráficos que aparecen a continuación sirven para crear, generar y guardar gráficos como imágenes. Esta funciones son compatibles con la versión actual de su producto de Altova. No obstante, tenga en cuenta que en futuras versiones del producto algunas funciones pueden dejar de ser compatibles o su comportamiento puede cambiar. Por tanto, consulte siempre la documentación del producto para conocer el funcionamiento de estas funciones en cada versión del producto.

Nota: las funciones para gráficos solamente son compatibles con los **productos servidor** de Altova y con las ediciones **Enterprise Edition** de los productos de Altova.

Nota: en los productos servidor de Altova los formatos de imagen compatibles para gráficos son `jpg`, `png` y `bmp`. La opción más recomendable es `png` porque no pierde información y es un formato comprimido. En las ediciones **Enterprise Edition** de las herramientas de escritorio de Altova, los formatos compatibles son `jpg`, `png`, `bmp` y `gif`.

 © 2016-2022 Altova GmbH

Funciones para generar y guardar gráficos

Estas funciones toman el objeto del gráfico (obtenido con las funciones de creación de gráficos) y generan una imagen o guardan una imagen en un archivo.

altova:generate-chart-image ($chart, $width, $height, $encoding) as atomic

donde

- $chart es el componente de extensión de gráficos obtenido con la función create-chart
- $width y $height deben especificarse con una unidad de longitud
- $encoding puede ser x-binarytobase64 o x-binarytobase16

La función devuelve la imagen del gráfico en la codificación elegida.

altova:generate-chart-image ($chart, $width, $height, $encoding, $imagetype) as atomic

donde

- $chart es el componente de extensión de gráficos obtenido con la función create-chart
- $width y $height deben especificarse con una unidad de longitud
- $encoding puede ser x-binarytobase64 o x-binarytobase16
- $imagetype puede ser uno de estos formatos de imagen: png, gif, bmp, jpg, jpeg. Recuerde que el formato gif no es compatible con los productos servidor de Altova (*ver nota al principio de este apartado*)

La función devuelve la imagen del gráfico en la codificación y formato de imagen elegidos.

altova:save-chart-image ($chart, $filename, $width, $height) as empty() ***(solo en Windows)***

donde

- $chart es el componente de extensión de gráficos obtenido con la función create-chart
- $filename es la ruta de acceso del archivo y el nombre de archivo donde se debe guardar la imagen del gráfico
- $width y $height deben especificarse con una unidad de longitud

La función guarda la imagen del gráfico en el archivo especificado en $filename. También puede usar la función xsl:result-document con encoding="x-base64tobinary", donde el contenido de imagen-datos se obtiene con las funciones generate-chart-image() o chart().

altova:save-chart-image ($chart, $filename, $width, $height, $imagetype) as empty() ***(solo en Windows)***

donde

- $chart es el componente de extensión de gráficos obtenido con la función create-chart
- $filename es la ruta de acceso del archivo y el nombre de archivo donde se debe guardar la imagen del gráfico

- `$width` y `$height` deben especificarse con una unidad de longitud
- `$imagetype` puede ser uno de estos formatos de imagen: `png`, `gif`, `bmp`, `jpg`, `jpeg`. Recuerde que el formato `gif` no es compatible con los productos servidor de Altova (*ver nota al principio de este apartado*)

La función guarda la imagen del gráfico en el archivo especificado en `$filename` en el formato de imagen elegido. También puede usar la función `xsl:result-document` con `encoding="x-base64tobinary"`, donde el contenido de imagen-datos se obtiene con las funciones `generate-chart-image()` o `chart()`.

Funciones para crear gráficos

Puede usar estas funciones para crear gráficos.

`altova:create-chart``($chart-config, $chart-data-series*) como componente de extensión de gráficos`

donde

- `$chart-config` es el componente de extensión chart-config obtenido con la función `create-chart-config` o con la función `create-chart-config-from-xml`
- `$chart-data-series` es el componente de extensión chart-data-series obtenido con la función `create-chart-data-series` o con la función `create-chart-data-series-from-rows`

La función devuelve un componente de extensión de gráficos, que se crea a partir de los datos suministrados con los argumentos.

`altova:chart``($chart-config, $chart-data-series*) como componente de extensión de gráficos`

donde

- `$chart-config` es el componente de extensión de gráficos. Se trata de una serie no ordenada de cuatro pares clave:valor en las que las claves son: `"width"`, `"height"`, `"title"`, and `"kind"`. Los valores de `width` y `height` son números enteros. El valor de `kind` es uno de estos: `Pie`, `Pie3d`, `BarChart`, `BarChart3d`, `BarChart3dGrouped`, `LineChart`, `ValueLineChart`, `RoundGauge`, `BarGauge`.
- Cada `$chart-data-series` es una matriz de tamaño 3 donde cada matriz define una serie chart-data. Cada matriz se compone de: (i) el nombre de la serie de datos, (ii) los valores del eje X y (iii) los valores del eje Y. Se pueden enviar varias series de datos; en el ejemplo siguiente, por ejemplo, las dos matrices dan datos, respectivamente, de las temperaturas mensuales máxima y mínima.

La función devuelve un elemento de tipo `xs:base64Binary` que contiene la imagen del gráfico. Esta imagen se crea a partir de los datos suministrados con los argumentos de la función. Recuerde que, al usar esta función matrices y asignaciones, solo se puede usar en XPath 3.1, XQuery 3.1 o XSLT 3.0.

Ejemplo:
```
altova:chart( map{'width':800, 'height':600, "kind":"LineChart", "title":"Monthly Temperatures"}, (['Min', $temps/Month, $temps/Month/@min], ['Max', $temps/Month, $temps/Month/@max]) )
```

`altova:create-chart-config``($type-name, $title) como componente de extensión de gráficos`

© 2016-2022 Altova GmbH

donde

- `$type-name` indica el gráfico que se va a crear: `Pie, Pie3d, BarChart, BarChart3d, BarChart3dGrouped, LineChart, ValueLineChart, RoundGauge, BarGauge`
- `$title` es el nombre del gráfico

La función devuelve un componente de extensión chart-config que contiene los datos de configuración del gráfico.

altova:create-chart-config-from-xml`($xml-struct) como componente de extensión de gráficos chart-config`

donde

- `$xml-struct` es la estructura XML que contiene los datos de configuración del gráfico

La función devuelve un componente de extensión chart-config que contiene los datos de configuración del gráfico. Estos datos se suministran en un fragmento de código XML[2205].

altova:create-chart-data-series`($series-name?, $x-values*, $y-values*) como componente de extensión de gráficos chart-data-series`

donde

- `$series-name` especifica el nombre de la serie
- `$x-values` presenta la lista de valores del eje X
- `$y-values` presenta la lista de valores del eje Y

La función devuelve un componente de extensión chart-data-series que contiene los datos necesarios para generar el gráfico: es decir, el nombre de las series y los datos de los ejes.

altova:create-chart-data-row`(x, y1, y2, y3, ...) como componente de extensión de gráficos chart-data-x-Ny-row`

donde

- `x` es el valor de la columna del eje X de la fila de datos del gráfico
- `yN` son los valores de las columnas del eje Y

La función devuelve un componente de extensión chart-data-x-Ny-row, que contiene los datos para la columna del eje X y las columnas del eje Y de una sola serie.

altova:create-chart-data-series-from-rows`($series-names as xs:string*, $row*) como componente de extensión de gráficos chart-data-series`

donde

- `$series-name` es el nombre de la series que se debe crear
- `$row` es el componente de extensión chart-data-x-Ny-row que se debe crear como serie

La función devuelve un componente de extensión chart-data-series, que contiene los datos para el eje X y el eje Y de la serie.

altova:create-chart-layer(`$chart-config, $chart-data-series*`) como componente de extensión de gráficos chart-layer

donde

- `$chart-config` es el componente de extensión chart-config obtenido con la función `create-chart-config` o con la función `create-chart-config-from-xml`
- `$chart-data-series` es el componente de extensión chart-data-series que se obtiene con la función `create-chart-data-series` o con la función `create-chart-data-series-from-rows`

La función devuelve un componente de extensión chart-layer, que contiene los datos de la capa de gráfico.

altova:create-multi-layer-chart(`$chart-config, $chart-data-series*, $chart-layer*`)

donde

- `$chart-config` es el componente de extensión chart-config obtenido con la función `create-chart-config` o con la función `create-chart-config-from-xml`
- `$chart-data-series` es el componente de extensión chart-data-series obtenido con la función `create-chart-data-series` o con la función `create-chart-data-series-from-rows`
- `$chart-layer` es el componente de extensión de gráficos multicapas obtenido con la función `create-chart-layer`

La función devuelve un componente de extensión de gráficos multicapas.

altova:create-multi-layer-chart(`$chart-config, $chart-data-series*, $chart-layer*, xs:boolean $mergecategoryvalues`)

donde

- `$chart-config` es el componente de extensión chart-config obtenido con la función `create-chart-config` o con la función `create-chart-config-from-xml`
- `$chart-data-series` es el componente de extensión chart-data-series obtenido con la función `create-chart-data-series` o con la función `create-chart-data-series-from-rows`
- `$chart-layer` es el componente de extensión de gráficos multicapa obtenido con la función `create-chart-layer`
- `$mergecategoryvalues` combina los valores de varias series de datos si `true` y no los combina si `false`

La función devuelve un componente de extensión de gráficos multicapa.

© 2016-2022 Altova GmbH

30.2.1.10.1 Estructura XML de los datos de gráficos

A continuación puede ver un fragmento de código XML con datos de gráfico, tal y como aparecería para las funciones de extensión para gráficos de Altova[2200]. Esto afecta al aspecto del gráfico. No todos los elementos se utilizan para todos los tipos de gráfico. Por ejemplo, el elemento `<Pie>` se omite en los gráficos de barras.

Nota: las funciones para gráficos son compatibles solamente con las ediciones **Enterprise** y **Server**.

```
<chart-config>
        <General
        SettingsVersion="1"  debe darse
        ChartKind="BarChart"  Pie, Pie3d, BarChart, StackedBarChart, BarChart3d, BarChart3dGrouped,
LineChart, ValueLineChart, AreaChart, StackedAreaChart, RoundGauge, BarGauge, CandleStick
        BKColor="#ffffff"  Color
        BKColorGradientEnd="#ffffff"  Color. En caso de degradado, BKColor y BKColorGradientEnd
definen el color del degradado
        BKMode="#ffffff"  Solid, HorzGradient, VertGradient
        BKFile="Ruta+NombreArchivo"  Cadena. Si el archivo existe, su contenido se dibuja sobre el fondo.
        BKFileMode="Stretch"  Stretch, ZoomToFit, Center, Tile
        ShowBorder="1"  Bool
        PlotBorderColor="#000000"  Color
        PlotBKColor="#ffffff"  Color
        Title=""  Cadena de texto
        ShowLegend="1"  Bool
        OutsideMargin="3.%"  PorcentajeOPíxel
        TitleToPlotMargin="3.%"  PorcentajeOPíxel
        LegendToPlotMargin="3.%"  PorcentajeOPíxel
        Orientation="vert"  Enums. Los valores posibles son: vert, horz
        >
        <TitleFont
           Color="#000000"  Color
           Name="Tahoma"  Cadena de texto
           Bold="1"  Bool
           Italic="0"  Bool
           Underline="0"  Bool
           MinFontHeight="10.pt"  TamañoFuente (solo valores pt)
           Size="8.%"  TamañoFuente />
        <LegendFont
           Color="#000000"
           Name="Tahoma"
           Bold="0"
           Italic="0"
           Underline="0"
           MinFontHeight="10.pt"
           Size="3.5%" />
        <AxisLabelFont
           Color="#000000"
           Name="Tahoma"
           Bold="1"
           Italic="0"
           Underline="0"
```

```
      MinFontHeight="10.pt"
      Size="5.%" />
</General>

<Line
      ConnectionShapeSize="1.%"  PorcentajeOPixel
      DrawFilledConnectionShapes="1"  Bool
      DrawOutlineConnectionShapes="0"  Bool
      DrawSlashConnectionShapes="0"  Bool
      DrawBackslashConnectionShapes="0"  Bool
/>

<Bar
      ShowShadow="1"  Bool
      ShadowColor="#a0a0a0"  Color
      OutlineColor="#000000"  Color
      ShowOutline="1"  Bool
/>

<Area
      Transparency="0"  UINT ( 0-255 ) 255 es totalmente transparente y 0 es opaco
      OutlineColor="#000000"  Color
      ShowOutline="1"  Bool
/>

<CandleStick
      FillHighClose="0"  Bool. Si es 0, el cuerpo está vacío. Si es 1, FillColorHighClose se usa
para el cuerpo de la vela
      FillColorHighClose="#ffffff"  Color. Para el cuerpo de la vela cuando el valor de close
> que el valor de open
      FillHighOpenWithSeriesColor="1"  Bool. Si es true, el color de la serie se usa para
rellenar el cuerpo de la vela cuando el valor de open > que el valor de close
      FillColorHighOpen="#000000"  Color. Para el cuerpo de la vela cuando el valor de open >
que el valor de close y FillHighOpenWithSeriesColor es false
/>

<Colors  Combinación de colores definida por el usuario. Este elemento está vacío por defecto,
excepto el estilo, y no tiene atributos Color
      UseSubsequentColors ="1"  Booleano. Si es 0, entonces se superpone el color. Si es 1, se
usan los colores siguientes de la capa de gráfico anterior
      Style="User"  Valores posibles: "Default", "Grayscale", "Colorful", "Pastel", "User"
      Colors="#52aca0"  Color: solamente se añade para el conjunto de colores definido por el
usuario
      Colors1="#d3c15d"  Color: solamente se añade para el conjunto de colores definido por el
usuario
      Colors2="#8971d8"  Color: solamente se añade para el conjunto de colores definido por el
usuario
      ...
      ColorsN=""  Cada conjunto de colores puede tener un máximo de diez colores, de Colors a
Colors9
</Colors>

<Pie
```


© 2016-2022 Altova GmbH

```
            ShowLabels="1"  Bool
            OutlineColor="#404040"  Color
            ShowOutline="1"  Bool
            StartAngle="0."  Double
            Clockwise="1"  Bool
            Draw2dHighlights="1"  Bool
            Transparency="0"  Int (De 0 a 255: 0 es opaco, 255 es totalmente transparente)
            DropShadowColor="#c0c0c0"  Color
            DropShadowSize="5.%"  PorcentajeOPíxel
            PieHeight="10.%"  PorcentajeOPíxel. Los valores de píxel pueden ser diferente en el
resultado debido a la inclinación 3D
            Tilt="40.0"  Double (De 10 a 90: la inclinación 3D en grados de un gráfico circular 3D)
            ShowDropShadow="1"  Bool
            ChartToLabelMargin="10.%"  PorcentajeOPíxel
            AddValueToLabel="0"  Bool
            AddPercentToLabel="0"  Bool
            AddPercentToLabels_DecimalDigits="0"  UINT ( 0 – 2 )
            >
            <LabelFont
                  Color="#000000"
                  Name="Arial"
                  Bold="0"
                  Italic="0"
                  Underline="0"
                  MinFontHeight="10.pt"
                  Size="4.%"    />
      </Pie>

      <XY>
            <XAxis  Axis
                  AutoRange="1"  Bool
                  AutoRangeIncludesZero="1"  Bool
                  RangeFrom="0."  Double: intervalo manual
                  RangeTill="1."  Double : intervalo manual
                  LabelToAxisMargin="3.%"  PorcentajeOPíxel
                  AxisLabel=""  Cadena de texto
                  AxisColor="#000000"  Color
                  AxisGridColor="#e6e6e6"  Color
                  ShowGrid="1"  Bool
                  UseAutoTick="1"  Bool
                  ManualTickInterval="1."  Double
                  AxisToChartMargin="0.px"  PorcentajeOPíxel
                  TickSize="3.px"  PorcentajeOPíxel
                  ShowTicks="1"  Bool
                  ShowValues="1"  Bool
                  AxisPosition="LeftOrBottom"  Enums: "LeftOrBottom", "RightOrTop", "AtValue"
                  AxisPositionAtValue = "0"  Double
                  >
                  <ValueFont
                        Color="#000000"
                        Name="Tahoma"
                        Bold="0"
                        Italic="0"
                        Underline="0"
```

```
                MinFontHeight="10.pt"
                Size="3.%"    />
        </XAxis>
        <YAxis  Eje (igual que XAxis)
                AutoRange="1"
                AutoRangeIncludesZero="1"
                RangeFrom="0."
                RangeTill="1."
                LabelToAxisMargin="3.%"
                AxisLabel=""
                AxisColor="#000000"
                AxisGridColor="#e6e6e6"
                ShowGrid="1"
                UseAutoTick="1"
                ManualTickInterval="1."
                AxisToChartMargin="0.px"
                TickSize="3.px"
                ShowTicks="1"  Bool
                ShowValues="1"  Bool
                AxisPosition="LeftOrBottom"  Enums: "LeftOrBottom", "RightOrTop", "AtValue"
                AxisPositionAtValue = "0"  Double
                >
                <ValueFont
                        Color="#000000"
                        Name="Tahoma"
                        Bold="0"
                        Italic="0"
                        Underline="0"
                        MinFontHeight="10.pt"
                        Size="3.%"/>
        </YAxis>
</XY>

<XY3d
        AxisAutoSize="1"  Bool: Si es false, XSize y YSize definen la relación de aspecto de los
ejes x e y. Si es true, la relación de aspecto es igual a la ventana del gráfico
        XSize="100.%"  PorcentajeOPíxel. Los valores en píxel pueden ser diferentes en el resultado
debido a la inclinación 3D y a la opción de ajustar al tamaño
        YSize="100.%"  PorcentajeOPíxel. Los valores en píxel pueden ser diferentes en el resultado
debido a la inclinación 3D y a la opción de ajustar al tamaño
        SeriesMargin="30.%"  PorcentajeOPíxel. Los valores en píxel pueden ser diferentes en el
resultado debido a la inclinación 3D y a la opción de ajustar al tamaño
        Tilt="20."  Double. De -90 a +90 grados
        Rot="20."  Double. De -359 a +359 grados
        FoV="50.">  Double. Campo de visión: de 1 a 120 grados
        >
        <ZAxis
                AutoRange="1"
                AutoRangeIncludesZero="1"
                RangeFrom="0."
                RangeTill="1."
                LabelToAxisMargin="3.%"
                AxisLabel=""
                AxisColor="#000000"
                AxisGridColor="#e6e6e6"
```


© 2016-2022 Altova GmbH

```
                ShowGrid="1"
                UseAutoTick="1"
                ManualTickInterval="1."
                AxisToChartMargin="0.px"
                TickSize="3.px" >
                <ValueFont
                        Color="#000000"
                        Name="Tahoma"
                        Bold="0"
                        Italic="0"
                        Underline="0"
                        MinFontHeight="10.pt"
                        Size="3.%"/>
        </ZAxis>
</XY3d>

<Gauge
        MinVal="0."  Double
        MaxVal="100."  Double
        MinAngle="225"  UINT: -359-359
        SweepAngle="270"  UINT: 1-359
        BorderToTick="1.%"  PorcentajeOPíxel
        MajorTickWidth="3.px"  PorcentajeOPíxel
        MajorTickLength="4.%"  PorcentajeOPíxel
        MinorTickWidth="1.px"  PorcentajeOPíxel
        MinorTickLength="3.%"  PorcentajeOPíxel
        BorderColor="#a0a0a0"  Color
        FillColor="#303535"  Color
        MajorTickColor="#a0c0b0"  Color
        MinorTickColor="#a0c0b0"  Color
        BorderWidth="2.%"  PorcentajeOPíxel
        NeedleBaseWidth="1.5%"  PorcentajeOPíxel
        NeedleBaseRadius="5.%"  PorcentajeOPíxel
        NeedleColor="#f00000"  Color
        NeedleBaseColor="#141414"  Color
        TickToTickValueMargin="5.%"  PorcentajeOPíxel
        MajorTickStep="10."  Double
        MinorTickStep="5."  Double
        RoundGaugeBorderToColorRange="0.%"  PorcentajeOPíxel
        RoundGaugeColorRangeWidth ="6.%"  PorcentajeOPíxel
        BarGaugeRadius="5.%"  PorcentajeOPíxel
        BarGaugeMaxHeight="20.%"  PorcentajeOPíxel
        RoundGaugeNeedleLength="45.%"  PorcentajeOPíxel
        BarGaugeNeedleLength="3.%"  PorcentajeOPíxel
        >
        <TicksFont
                Color="#a0c0b0"
                Name="Tahoma"
                Bold="0"
                Italic="0"
                Underline="0"
                MinFontHeight="10.pt"
                Size="4.%"
        />
```

```
            <ColorRanges>   Intervalos de color definidos por el usuario. Está vacío por defecto y no
tiene elementos secundarios
                  <Entry
                        From="50. "   Double
                        FillWithColor="1"   Bool
                        Color="#00ff00"   Color
                  />
                  <Entry
                        From="50.0"
                        FillWithColor="1"
                        Color="#ff0000"
                  />
                  ...
            </ColorRanges>
      </Gauge>
</chart-config>
```

30.2.1.10.2 Ejemplo: funciones para gráficos

A continuación puede ver un ejemplo de documento XSLT que muestra cómo usar las funciones de extensión para gráficos de Altova[2200]. Después ofrecemos un documento XML y una captura de pantalla de la imagen generada cuando el documento XML se procesa con el documento XSLT usando el motor XSLT 2.0.

Nota: las funciones para gráficos solamente son compatibles con las ediciones **Enterprise Edition** y **Server** de los productos de Altova.

Nota: para más información sobre las tablas de datos del gráfico, consulte la documentación de las herramientas XMLSpy y StyleVision de Altova.

Documento XSLT

Este documento XSLT usa las funciones de extensión para gráficos de Altova para generar un gráfico circular. Este XSLT se puede usar para procesar el documento XML que aparece más abajo.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:altovaext="http://www.altova.com/xslt-extensions"
   exclude-result-prefixes="#all">
   <xsl:output version="4.0" method="html" indent="yes" encoding="UTF-8"/>
   <xsl:template match="/">
      <html>
         <head>
            <title>
               <xsl:text>HTML Page with Embedded Chart</xsl:text>
            </title>
         </head>
         <body>
            <xsl:for-each select="/Data/Region[1]">
```

```
                <xsl:variable name="extChartConfig" as="item()*">
                    <xsl:variable name="ext-chart-settings" as="item()*">
                        <chart-config>
                            <General
                                SettingsVersion="1"
                                ChartKind="Pie3d"
                                BKColor="#ffffff"
                                ShowBorder="1"
                                PlotBorderColor="#000000"
                                PlotBKColor="#ffffff"
                                Title="{@id}"
                                ShowLegend="1"
                                OutsideMargin="3.2%"
                                TitleToPlotMargin="3.%"
                                LegendToPlotMargin="6.%"
                                >
                                <TitleFont
                                    Color="#023d7d"
                                    Name="Tahoma"
                                    Bold="1"
                                    Italic="0"
                                    Underline="0"
                                    MinFontHeight="10.pt"
                                    Size="8.%" />
                            </General>
                        </chart-config>
                    </xsl:variable>
                    <xsl:sequence select="altovaext:create-chart-config-from-xml( $ext-
chart-settings )"/>
                </xsl:variable>
                <xsl:variable name="chartDataSeries" as="item()*">
                    <xsl:variable name="chartDataRows" as="item()*">
                        <xsl:for-each select="(Year)">
                            <xsl:sequence select="altovaext:create-chart-data-row( (@id),
( .) )"/>
                        </xsl:for-each>
                    </xsl:variable>
                    <xsl:variable name="chartDataSeriesNames" as="xs:string*"
select=" ( (&amp;quot;Series 1&amp;quot;), &amp;apos;&amp;apos; )[1]"/>
                    <xsl:sequence
                        select="altovaext:create-chart-data-series-from-
rows( $chartDataSeriesNames, $chartDataRows)"/>
                </xsl:variable>
                <xsl:variable name="ChartObj" select="altovaext:create-
chart( $extChartConfig, ( $chartDataSeries), false() )"/>
                <xsl:variable name="sChartFileName" select="'mychart1.png'"/>
                <img src="{$sChartFileName, altovaext:save-chart-image( $ChartObj,
$sChartFileName, 400, 400 ) }"/>
            </xsl:for-each>
        </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Documento XML

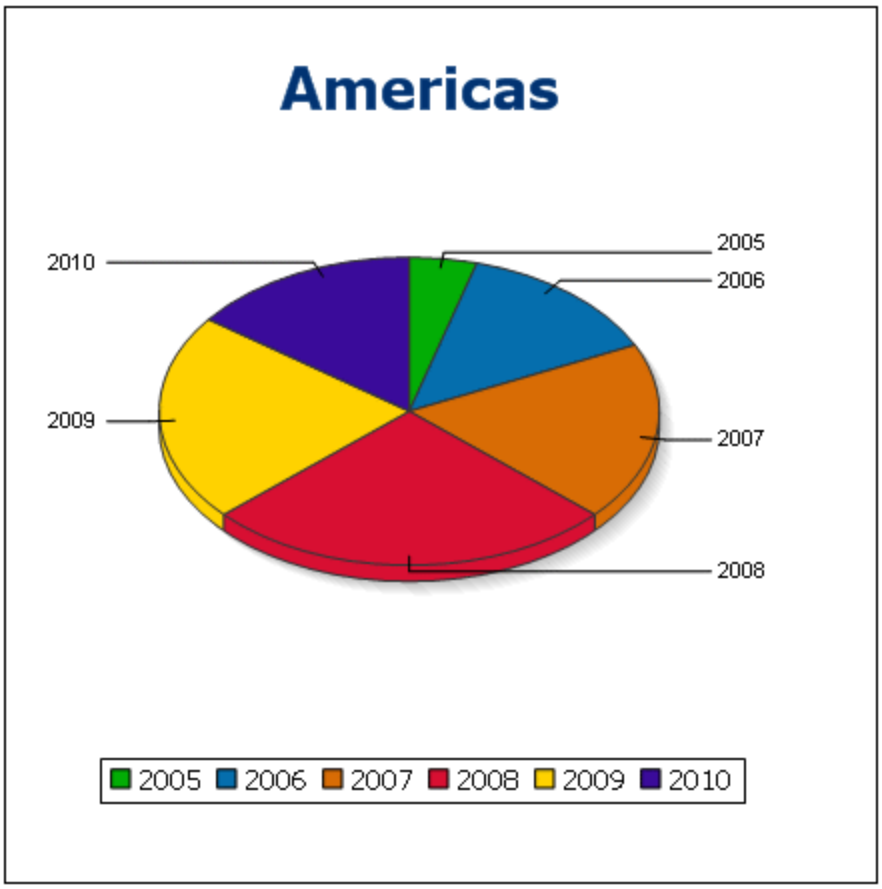
Este documento XML se puede procesar con el XSLT que aparece arriba. Los datos del documento XML se usan para generar el gráfico circular de la imagen que aparece más abajo.

```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
   <ChartType>Pie Chart 2D</ChartType>
   <Region id="Americas">
      <Year id="2005">30000</Year>
      <Year id="2006">90000</Year>
      <Year id="2007">120000</Year>
      <Year id="2008">180000</Year>
      <Year id="2009">140000</Year>
      <Year id="2010">100000</Year>
   </Region>
   <Region id="Europe">
      <Year id="2005">50000</Year>
      <Year id="2006">60000</Year>
      <Year id="2007">80000</Year>
      <Year id="2008">100000</Year>
      <Year id="2009">95000</Year>
      <Year id="2010">80000</Year>
   </Region>
   <Region id="Asia">
      <Year id="2005">10000</Year>
      <Year id="2006">25000</Year>
      <Year id="2007">70000</Year>
      <Year id="2008">110000</Year>
      <Year id="2009">125000</Year>
      <Year id="2010">150000</Year>
   </Region>
</Data>
```

Imagen de salida

Este gráfico circular se generó al procesar el documento XML anterior con el archivo XSLT que aparece al principio de este apartado.

© 2016-2022 Altova GmbH



30.2.2 Funciones de extensión varias

Los lenguajes de programación como Java y C# ofrecen varias funciones predefinidas que no están disponibles como funciones XQuery/XPath ni XSLT. Un ejemplo son las funciones matemáticas de Java `sin()` y `cos()`. Si los diseñadores de hojas de estilos XSLT y consultas XQuery tuvieran acceso a estas funciones, el área de aplicación de sus hojas de estilos y consultas aumentaría y su trabajo sería un poco más sencillo.

Los motores XSLT y XQuery de los productos de Altova admiten el uso de funciones de extensión en Java[2214] y .NET[2223], así como scripts MSXSL para XSLT[2229].

Esta sección describe cómo usar funciones de extensión y scripts MSXSL en hojas de estilos XSLT y documentos XQuery. Las funciones de extensión pueden organizarse en varios grupos:

- Funciones de extensión Java[2214]
- Funciones de extensión .NET[2223]
- Scripts MSXSL para XSLT[2229]

En los apartados de esta sección nos ocupamos de tres aspectos fundamentales: (i) cómo se llaman las funciones en sus respectivas bibliotecas, (ii) qué reglas deben seguirse para convertir los argumentos de una llamada a función en el formato de entrada necesario de la función y (iii) qué reglas deben seguirse para la conversión del tipo devuelto.

Requisitos

Para que estas funciones de extensión funcionen es necesario tener Java Runtime Environment (para las funciones Java) y .NET Framework 2.0 o superior (para las funciones .NET) instalado en el equipo que ejecuta la transformación XSLT o XQuery.

30.2.2.1 Funciones de extensión Java

Puede usar una función de extensión Java dentro de una expresión XPath o XQuery para invocar un constructor Java o llamar a un método Java (estático o de instancia).

Un campo de una clase Java se trata como un método sin argumentos. Un campo puede ser estático o de instancia. Más adelante describimos cómo se accede a los campos estáticos y de instancia.

Este apartado tiene varias partes:

- Constructores Java 2220
- Métodos estáticos y campos estáticos 2220
- Métodos de instancia y campos de instancia 2221
- Tipos de datos: conversión de XPath/XQuery en Java 2222
- Tipos de datos: conversión de Java en XPath/XQuery 2223

Tenga en cuenta que:

- si está usando un producto de escritorio de Altova, la aplicación intentará detectar automáticamente la ruta de acceso al equipo virtual Java; para ello leerá (en este orden): (i) el registro de Windows y (ii) la variable de entorno **JAVA_HOME**. También puede añadir una ruta personal en el cuadro de diálogo "Opciones" de la aplicación; esta ruta tendrá prioridad frente a cualquier otra ruta de acceso a un equipo virtual Java que se detecte automáticamente.
- si está usando un producto servidor de Altova en un equipo Windows, la ruta de acceso al equipo virtual Java se leerá primero desde el registro de Windows; si esto no ocurre se usa la variable de entorno **JAVA_HOME**.
- si está usando un producto servidor de Altova en un equipo Linux o macOS, entonces asegúrese de que la variable de entorno **JAVA_HOME** está definida correctamente y la biblioteca Java de equipos virtuales (en Windows, el archivo **jvm.dll**) se encuentra en uno de estos directorios: **\bin\server** o **\bin\client**.

Formato de la función de extensión

La función de extensión de la expresión XPath/XQuery debe tener este formato `prefijo:nombreFunción()`.

- La parte `prefijo:` identifica la función de extensión como función Java. Lo hace asociando la función de extensión con una declaración de espacio de nombres del ámbito, cuyo URI debe empezar por `java:` (*ver ejemplos más abajo*). La declaración de espacio de nombres debe identificar una clase Java, por ejemplo: `xmlns:myns="java:java.lang.Math"`. Sin embargo, también puede ser

 © 2016-2022 Altova GmbH

simplemente: `xmlns:myns="java"` (sin los dos puntos), dejando la identificación de la clase Java a la parte `nombreFunción()` de la función de extensión.

- La parte `nombreFunción()` identifica el método Java al que se llama y presenta los argumentos para el método (*ver ejemplos más abajo*). Sin embargo, si el URI de espacio de nombres identificado por la parte `prefijo:` no identifica una clase Java (*ver punto anterior*), entonces la clase Java debe identificarse en la parte `nombreFunción()`, antes de la clase y separada de la clase por un punto (*ver el segundo ejemplo XSLT que aparece más abajo*).

Nota: la clase a la que se llama debe estar en la ruta de acceso de clase del equipo.

Ejemplo de código XSLT

Aquí ofrecemos dos ejemplos de cómo se puede llamar a un método estático. En el primer ejemplo, el nombre de la clase (`java.lang.Math`) se incluye en el URI de espacio de nombres y, por tanto, no puede estar en la parte `nombreFunción()`. En el segundo ejemplo, la parte `prefijo:` presenta el prefijo `java:` mientras que la parte `nombreFunción()` identifica la clase y el método.

```
<xsl:value-of xmlns:jMath="java:java.lang.Math"
              select="jMath:cos(3.14)" />

<xsl:value-of xmlns:jmath="java"
              select="jmath:java.lang.Math.cos(3.14)" />
```

El método nombrado en la función de extensión (`cos()`) debe coincidir con el nombre de un método estático público de la clase Java nombrada (`java.lang.Math`).

Ejemplo de código XQuery

Aquí puede ver un ejemplo de código XQuery similar al código XSLT anterior:

```
<cosine xmlns:jMath="java:java.lang.Math">
   {jMath:cos(3.14)}
</cosine>
```

Clases Java definidas por el usuario

Si creó sus propias clases Java, a los métodos de estas clases se les llama de otra manera, dependiendo de: (i) si a las clases se accede por medio de un archivo JAR o de un archivo de clases y (ii) si estos archivos están en el directorio actual (el directorio del documento XSLT o XQuery). Para más información consulte los apartados Archivos de clases definidos por el usuario 2215 y Archivos Jar definidos por el usuario 2219. Recuerde que debe especificar las rutas de acceso de los archivos de clases que no están en el directorio actual y de todos los archivos JAR.

30.2.2.1.1 Archivos de clases definidos por el usuario

Si se accede a las clases por medio de un archivo de clases, entonces hay cuatro posibilidades:

- El archivo de clases está en un paquete. El archivo XSLT/XQuery está en la misma carpeta que el paquete Java. (*ver ejemplo*[2216].)
- El archivo de clases no está en un paquete. El archivo XSLT/XQuery está en la misma carpeta que el archivo de clases. (*ver ejemplo*[2217].)
- El archivo de clases está en un paquete. El archivo XSLT/XQuery está en una carpeta cualquiera. (*ver ejemplo*[2217].)
- El archivo de clases no está en un paquete. El archivo XSLT/XQuery está una carpeta cualquiera. (*ver ejemplo*[2218].)

Imaginemos que tenemos un archivo de clases que no está en un paquete y que está en la misma carpeta que el documento XSLT/XQuery. En este caso, puesto que en la carpeta se encuentran todas las clases, no es necesario especificar la ubicación del archivo. La sintaxis que se utiliza para identificar una clase es esta:

```
java:nombreClase
```

donde

`java`: indica que se está llamando a una función definida por el usuario (por defecto se cargan las clases Java del directorio actual)

`nombreClase` es el nombre de la clase del método elegido

La clase se identifica en un URI de espacio de nombres y el espacio de nombres se usa como prefijo para la llamada al método.

El archivo de clases está en un paquete. El archivo XSLT/XQuery está en la misma carpeta que el paquete Java

El código que aparece a continuación llama al método `getVehicleType()` de la clase `Car` del paquete `com.altova.extfunc`. El paquete `com.altova.extfunc` está en la carpeta `JavaProject`. El archivo XSLT también está en la carpeta `JavaProject`.

```
<xsl:stylesheet version="2.0"
       xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
       xmlns:xs="http://www.w3.org/2001/XMLSchema"
       xmlns:fn="http://www.w3.org/2005/xpath-functions"
       xmlns:car="java:com.altova.extfunc.Car" >
<xsl:output exclude-result-prefixes="fn car xsl fo xs"/>

<xsl:template match="/">
   <a>
   <xsl:value-of select="car:getVehicleType()"/>
   </a>
</xsl:template>

</xsl:stylesheet>
```


© 2016-2022 Altova GmbH

El archivo de clases está referenciado. El archivo XSLT/XQuery está en la misma carpeta que el archivo de clases

El código que aparece a continuación llama al método `getVehicleType()` de la clase `Car`. Digamos que: (i) el archivo de clases `Car` está en esta carpeta: `JavaProject/com/altova/extfunc` y que (ii) esa carpeta es la del ejemplo siguiente. El archivo XSLT también está en la carpeta `JavaProject/com/altova/extfunc`.

```
<xsl:stylesheet version="2.0"
       xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
       xmlns:xs="http://www.w3.org/2001/XMLSchema"
       xmlns:fn="http://www.w3.org/2005/xpath-functions"
       xmlns:car="java:Car" >
<xsl:output exclude-result-prefixes="fn car xsl fo xs"/>

<xsl:template match="/">
   <a>
   <xsl:value-of select="car:getVehicleType()"/>
   </a>
</xsl:template>

</xsl:stylesheet>
```

El archivo de clases está en un paquete. El archivo XSLT/XQuery está en una carpeta cualquiera

El código que aparece a continuación llama al método `getCarColor()` de la clase `Car` del paquete `com.altova.extfunc`. El paquete `com.altova.extfunc` está en la carpeta `JavaProject`. El archivo XSLT está en otra carpeta cualquiera. En este caso debe especificarse la ubicación del paquete dentro del URI como una cadena de consulta. La sintaxis es esta:

```
java:nombreClase[?ruta=uri-del-paquete]
```

donde

`java`: indica que se está llamando a una función Java definida por el usuario
`uri-del-paquete` es el URI del paquete Java
`nombreClase` es el nombre de la clase del método elegido

La clase se identifica en un URI de espacio de nombres y el espacio de nombres se usa como prefijo para la llamada al método. El ejemplo de código que aparece a continuación explica cómo se accede a un archivo de clases que está ubicado en un directorio que no es el directorio actual.

```
<xsl:stylesheet version="2.0"
       xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
       xmlns:xs="http://www.w3.org/2001/XMLSchema"
       xmlns:fn="http://www.w3.org/2005/xpath-functions"
       xmlns:car="java:com.altova.extfunc.Car?path=file:///C:/JavaProject/" >

<xsl:output exclude-result-prefixes="fn car xsl xs"/>
```

```
<xsl:template match="/">
   <xsl:variable name="myCar" select="car:new('red')" />
   <a><xsl:value-of select="car:getCarColor($myCar)"/></a>
</xsl:template>

</xsl:stylesheet>
```

El archivo de clases no está en un paquete. El archivo XSLT/XQuery está una carpeta cualquiera

El código que aparece a continuación llama al método `getCarColor()` de la clase `Car`. Digamos que el archivo de clases `Car` está en la carpeta `C:/JavaProject/com/altova/extfunc` y que el archivo XSLT está en otra carpeta cualquiera. En este caso debe especificarse la ubicación del paquete dentro del URI como una cadena de consulta. La sintaxis es esta:

```
java:nombreClase[?ruta=<uri-del-archivoClases>]
```

donde

`java`: indica que se está llamando a una función Java definida por el usuario
`uri-del-archivoClases` es el URI de la carpeta donde se ubica el archivo de clases
`nombreClase` es el nombre de la clase del método elegido

La clase se identifica en un URI de espacio de nombres y el espacio de nombres se usa como prefijo para la llamada al método. El ejemplo de código que aparece a continuación explica cómo se accede a un archivo de clases que está ubicado en un directorio que no es el directorio actual.

```
<xsl:stylesheet version="2.0"
       xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
       xmlns:xs="http://www.w3.org/2001/XMLSchema"
       xmlns:fn="http://www.w3.org/2005/xpath-functions"
       xmlns:car="java:Car?path=file:///C:/JavaProject/com/altova/extfunc/" >

<xsl:output exclude-result-prefixes="fn car xsl xs"/>

<xsl:template match="/">
   <xsl:variable name="myCar" select="car:new('red')" />
   <a><xsl:value-of select="car:getCarColor($myCar)"/></a>
</xsl:template>

</xsl:stylesheet>
```

Nota: cuando se presenta una ruta de acceso por medio de una función de extensión, la ruta de acceso se añade al ClassLoader.

© 2016-2022 Altova GmbH

30.2.2.1.2 Archivos JAR definidos por el usuario

Si se accede a las clases por medio de un archivo JAR, entonces se debe especificar el URI del archivo JAR usando esta sintaxis:

```
xmlns:claseEspacioNombres="java:nombreClase?ruta=jar:uri-del-archivoJar!/"
```

Para la llamada al método se usa el prefijo del URI de espacio de nombres que identifica la clase: `claseEspacioNombres:método()`

En la sintaxis anterior:

`java:` indica que se está llamando a una función de Java
`nombreClase` es el nombre de la clase definida por el usuario
`?` es el separador entre el nombre de la clase y la ruta de acceso
`ruta=jar:` indica que se ofrece una ruta de acceso a un archivo JAR
`uri-del-archivoJar` es el URI del archivo JAR
`!/` es el delimitador final de la ruta de acceso
`claseEspacioNombres:método()` es la llamada al método

Otra opción es dar el nombre de la clase con la llamada al método. Por ejemplo:

```
xmlns:ns1="java:docx.layout.pages?
path=jar:file:///c:/projects/docs/docx.jar!/"
ns1:main()

xmlns:ns2="java?path=jar:file:///c:/projects/docs/docx.jar!/"
ns2:docx.layout.pages.main()
```

Y aquí puede ver un ejemplo de XSLT que usa un archivo JAR para llamar a una función de extensión Java:

```
<xsl:stylesheet version="2.0"
	xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
	xmlns:xs="http://www.w3.org/2001/XMLSchema"
	xmlns:fn="http://www.w3.org/2005/xpath-functions"
	xmlns:car="java?path=jar:file:///C:/test/Car1.jar!/" >
<xsl:output exclude-result-prefixes="fn car xsl xs"/>

<xsl:template match="/">
	<xsl:variable name="myCar" select="car:Car1.new('red')" />
	<a><xsl:value-of select="car:Car1.getCarColor($myCar)"/></a>
</xsl:template>

<xsl:template match="car"/>

</xsl:stylesheet>
```

Nota: cuando se presenta una ruta de acceso por medio de una función de extensión, la ruta de acceso se añade al ClassLoader.

30.2.2.1.3 Constructores

Una función de extensión se puede usar para llamar a un constructor Java. A todos los constructores se les llama con la pseudofunción `new()`.

Si el resultado de una llamada a un constructor Java se puede convertir de manera implícita a tipos de datos XPath/XQuery[2223], entonces la llamada a la función de extensión Java devuelve una secuencia que es un tipo de datos XPath/XQuery. Si el resultado de una llamada a un constructor Java no se puede convertir a un tipo de datos XPath/XQuery adecuado, entonces el constructor crea un objeto Java contenido con un tipo que es el nombre de la clase que devuelve ese objeto Java. Por ejemplo, si se llama a un constructor para la clase `java.util.Date` (`java.util.Date.new()`), entonces se devuelve un objeto que tiene el tipo `java.util.Date`. Puede que el formato léxico del objeto devuelto no coincida con el formato léxico de un tipo de datos XPath y, por tanto, su valor debe convertirse al formato léxico del tipo de datos XPath pertinente y después al tipo de datos XPath.

Puede hacer dos cosas con el objeto Java creado por un constructor:

- Puede asignar el objeto a una variable:
  ```
  <xsl:variable name="currentdate" select="date:new()"
  xmlns:date="java:java.util.Date" />
  ```
- Puede pasar el objeto a una función de extensión (*ver métodos de instancia y campos de instancia*[2221]):
  ```
  <xsl:value-of select="date:toString(date:new())" xmlns:date="java:java.util.Date" />
  ```

30.2.2.1.4 Métodos estáticos y campos estáticos

La llamada a un método estático la hace directamente su nombre Java y se hace presentando los argumentos para el método. A los campos estáticos (es decir, los métodos que no toman argumentos), como los campos de valor constante `E` y `PI`, se accede sin especificar ningún argumento.

Ejemplos de código XSLT

Aquí puede ver varios ejemplos de cómo se llama a métodos y campos estáticos:

```
<xsl:value-of xmlns:jMath="java:java.lang.Math"
              select="jMath:cos(3.14)" />

<xsl:value-of xmlns:jMath="java:java.lang.Math"
              select="jMath:cos( jMath:PI() )" />

<xsl:value-of xmlns:jMath="java:java.lang.Math"
              select="jMath:E() * jMath:cos(3.14)" />
```

Observe que las funciones de extensión anteriores tienen el formato `prefijo:nombreFunción()`. En los tres ejemplos anteriores, el prefijo es `jMath:`, que está asociado al URI de espacio de nombres `java:java.lang.Math`. (El URI de espacio de nombres debe empezar por `java:`. En los ejemplos anteriores se extiende para contener el nombre de la clase (`java.lang.Math`).) La parte `nombreFunción()` de las funciones de extensión debe coincidir con el nombre de una clase pública (p. ej. `java.lang.Math`) seguido del

© 2016-2022 Altova GmbH

nombre de un método estático público con sus argumentos (como `cos(3.14)`) o de un campo estático público (como `PI()`).

En los tres ejemplos anteriores, el nombre de la clase se incluyó en el URI de espacio de nombres. Si no estuviera en el URI de espacio de nombres, se incluiría en la parte `nombreFunción()` de la función de extensión. Por ejemplo:

```
<xsl:value-of xmlns:java="java:"
              select="java:java.lang.Math.cos(3.14)" />
```

Ejemplo de XQuery

Un ejemplo de XQuery similar sería:

```
<cosine xmlns:jMath="java:java.lang.Math">
   {jMath:cos(3.14)}
</cosine>
```

30.2.2.1.5 Métodos de instancia y campos de instancia

A un método de instancia se le pasa un objeto Java como primer argumento de la llamada a método. Dicho objeto Java suele crearse usando una función de extensión (por ejemplo, una llamada a un constructor) o un parámetro o una variable de hoja de estilos. Un ejemplo de código XSLT de este tipo sería:

```
<xsl:stylesheet version="1.0" exclude-result-prefixes="date"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:date="java:java.util.Date"
  xmlns:jlang="java:java.lang">
  <xsl:param name="CurrentDate" select="date:new()"/>
  <xsl:template match="/">
      <enrollment institution-id="Altova School"
                date="{date:toString($CurrentDate)}"
                type="{jlang:Object.toString(jlang:Object.getClass( date:new() ))}">
      </enrollment>
  </xsl:template>
</xsl:stylesheet>
```

En el ejemplo anterior el valor del nodo `enrollment/@type` se crea de la siguiente manera:

1. Se crea un objeto con un constructor para la clase `java.util.Date` (con el constructor `date:new()`).
2. Este objeto Java se pasa como argumento del método `jlang.Object.getClass`.
3. El objeto que obtiene el método `getClass` se pasa como argumento al método `jlang.Object.toString`.

El resultado (el valor de `@type`) será una cadena con este valor: `java.util.Date`.

En teoría, un campo de instancia es diferente de un método de instancia porque al campo de instancia no se pasa como argumento un objeto Java propiamente dicho. En su lugar se pasa como argumento un parámetro o variable. Sin embargo, el parámetro o la variable puede contener el valor devuelto por un objeto Java. Por ejemplo, el parámetro `CurrentDate` toma el valor que devolvió un constructor para la clase `java.util.Date`.

Este valor se pasa después como argumento al método de instancia `date:toString` a fin de suministrar el valor de `/enrollment/@date`.

30.2.2.1.6 Tipos de datos: conversión de XPath/XQuery en Java

Cuando se llama a una función Java desde dentro de una expresión XPath/XQuery, el tipo de datos de los argumentos de la función es importante a la hora de determinar a cuál de las clases Java que tienen el mismo nombre se llama.

En Java se siguen estas reglas:

- Si hay más de un método Java con el mismo nombre, pero cada método tiene un número diferente de argumentos, entonces se selecciona el método Java que mejor se ajusta al número de argumentos de la llamada a función.
- Los tipos de datos de cadena, numéricos y booleanos de XPath/XQuery (*ver lista más abajo*) se convierten de forma implícita en el tipo de datos Java correspondiente. Si el tipo XPath/XQuery suministrado se puede convertir a más de un tipo Java (p. ej. `xs:integer`), entonces se selecciona el tipo Java que se declaró para el método seleccionado. Por ejemplo, si el método Java al que se llama es `fx(decimal)` y el tipo de datos XPath/XQuery suministrado es `xs:integer`, entonces `xs:integer` se convierte en el tipo de datos Java `decimal`.

La tabla que aparece a continuación enumera las conversiones implícitas de los tipos de cadena, numéricos y booleanos XPath/XQuery en tipos de datos Java.

`xs:string`	`java.lang.String`
`xs:boolean`	`boolean` (primitivo), `java.lang.Boolean`
`xs:integer`	int, long, short, byte, float, double y sus clases contenedoras, como `java.lang.Integer`
`xs:float`	`float` (primitivo), `java.lang.Float`, `double` (primitivo)
`xs:double`	`double` (primitivo), `java.lang.Double`
`xs:decimal`	`float` (primitivo), `java.lang.Float`, `double`(primitivo), `java.lang.Double`

Los subtipos de los tipos de datos XML Schema de la tabla anterior (que se usan en XPath y XQuery) también se convierten en los tipos Java correspondientes al tipo antecesor del subtipo.

En algunos casos quizás no sea posible seleccionar el método Java correcto usando la información dada. Por ejemplo, imagine que:

- El argumento presentado es un valor `xs:untypedAtomic` de 10 y está destinado al método `mimétodo(float)`.
- Sin embargo, hay otro método en la clase que toma un argumento de otro tipo de datos: `mimétodo(double)`.

© 2016-2022 Altova GmbH

- Puesto que los métodos tienen el mismo nombre y el tipo suministrado (`xs:untypedAtomic`) se puede convertir correctamente tanto en `float` como en `double`, es posible que `xs:untypedAtomic` se convierta en `double` en lugar de en `float`.
- Por consiguiente, el método seleccionado no será el método necesario y quizás no produzca el resultado esperado. Una solución es crear un método definido por el usuario con un nombre diferente y usar ese método.

Los tipos que no aparecen en la lista anterior (p. ej. `xs:date`) no se convertirán y generarán un error. No obstante, tenga en cuenta que en algunos casos, es posible crear el tipo Java necesario usando un constructor Java.

30.2.2.1.7 Tipos de datos: conversión de Java en XPath/XQuery

Cuando un método Java devuelve un valor y el tipo de datos del valor es un tipo de cadena, numérico o booleano, entonces se convierte en el tipo de datos XPath/XQuery correspondiente. Por ejemplo, los tipos de datos Java `java.lang.Boolean` y `boolean` se convierten en `xsd:boolean`.

Las matrices unidimensionales devueltas por las funciones se extienden en una secuencia. Las matrices multidimensionales no se convierten y, por tanto, deberían ser contenidas.

Cuando se devuelve un objeto Java contenido o un tipo de datos que no es de cadena, numérico ni booleano, puede garantizar la conversión del tipo XPath/XQuery necesario usando primero un método Java (p. ej. `toString`) para convertir el objeto Java en una cadena. En XPath/XQuery la cadena se puede modificar para ajustarse a la representación léxica del tipo necesario y convertirse después en dicho tipo (usando la expresión `cast as`, por ejemplo).

30.2.2.2 Funciones de extensión .NET

Si trabaja en la plataforma .NET desde un equipo Windows, puede usar funciones de extensión escritas en cualquier lenguaje .NET (p. ej. C#). Una función de extensión .NET se puede usar dentro de una expresión XPath/XQuery para invocar un constructor, una propiedad o un método (estático o de instancia) de una clase .NET.

A una propiedad de una clase .NET se le llama usando la sintaxis `get_NombrePropiedad()`.

Este apartado tiene varias partes:

- Constructores 2226
- Métodos estáticos y campos estáticos 2226
- Métodos de instancia y campos de instancia 2227
- Tipos de datos: conversión de XPath/XQuery en .NET 2228
- Tipos de datos: conversión de .NET en XPath/XQuery 2229

Formato de la función de extensión

La función de extensión de la expresión XPath/XQuery debe tener este formato `prefijo:nombreFunción()`.

- La parte `prefijo:` está asociada a un URI que identifica la clase .NET.
- La parte `nombreFunción()` identifica el constructor, la propiedad o el método (estático o de instancia) dentro de la clase .NET y, si es necesario, suministra los argumentos.
- El URI debe empezar por `clitype:` (que identifica la función como función de extensión .NET).
- El formato `prefijo:nombreFunción()` de la función de extensión se puede usar con clases del sistema y con clases de un ensamblado cargado. No obstante, si se tiene que cargar una clase, será necesario suministrar parámetros que contengan la información necesaria.

Parámetros

Para cargar un ensamblado se usan estos parámetros:

`asm`	El nombre del ensamblado que se debe cargar.
`ver`	El número de versión (máximo cuatro enteros separados por puntos).
`sn`	El símbolo de clave del nombre seguro del ensamblado (16 dígitos hexadecimales).
`from`	Un URI que da la ubicación del ensamblado (DLL) que se debe cargar. Si el URI es relativo, es relativo al archivo XSLT o XQuery. Si está presente este parámetro, se ignoran los demás parámetros.
`partialname`	El nombre parcial del ensamblado. Se suministra a `Assembly.LoadWith.PartialName()`, que intentará cargar el ensamblado. Si está presente el parámetro `partialname`, se ignoran los demás parámetros.
`loc`	La configuración regional, por ejemplo, `en-US`. La configuración predeterminada es `neutral`.

Si el ensamblado se debe cargar desde un archivo DLL, use el parámetro `from` y omita el parámetro `sn`. Si el ensamblado se debe cargar desde el caché general de ensamblados (GAC), use el parámetro `sn` y omita el parámetro `from`.

Debe insertar un signo de interrogación final antes del primer parámetro y los parámetros deben separarse con un punto y coma (;). El nombre de parámetro da su valor con un signo igual (=), como en el ejemplo que aparece más abajo.

Ejemplos de declaraciones de espacios de nombres

Esto es un ejemplo de una declaración de espacio de nombres en XSLT que identifica la clase del sistema `System.Environment`:

```
xmlns:myns="clitype:System.Environment"
```

Esto es un ejemplo de una declaración de espacio de nombres en XSLT que identifica la clase que se debe cargar como `Trade.Forward.Scrip`:

```
xmlns:myns="clitype:Trade.Forward.Scrip?asm=forward;version=10.6.2.1"
```


© 2016-2022 Altova GmbH

Esto es un ejemplo de una declaración de espacio de nombres en XQuery que identifica la clase del sistema `MyManagedDLL.testClass:`. Existen dos tipos de clases:

1. Cuando el ensamblado se carga desde el GAC:
   ```
   declare namespace cs="clitype:MyManagedDLL.testClass?asm=MyManagedDLL;
          ver=1.2.3.4;loc=neutral;sn=b9f091b72dccfba8";
   ```

2. Cuando el ensamblado se carga desde el archivo DLL (ver las referencias parciales y completas):
   ```
          declare namespace cs="clitype:MyManagedDLL.testClass?from=file:///C:/Altova
          Projects/extFunctions/MyManagedDLL.dll;

   declare namespace cs="clitype:MyManagedDLL.testClass?from=MyManagedDLL.dll;
   ```

Ejemplo de código XSLT

Aquí puede ver un ejemplo de código XSLT que llama a funciones de la clase del sistema `System.Math`:

```
<xsl:stylesheet version="2.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:fn="http://www.w3.org/2005/xpath-functions">
   <xsl:output method="xml" omit-xml-declaration="yes" />
   <xsl:template match="/">
      <math xmlns:math="clitype:System.Math">
         <sqrt><xsl:value-of select="math:Sqrt(9)"/></sqrt>
         <pi><xsl:value-of select="math:PI()"/></pi>
         <e><xsl:value-of select="math:E()"/></e>
         <pow><xsl:value-of select="math:Pow(math:PI(), math:E())"/></pow>
      </math>
   </xsl:template>
</xsl:stylesheet>
```

La declaración de espacio de nombres del elemento `math` asocia el prefijo `math:` al URI `clitype:System.Math`. La parte inicial `clitype:` del URI indica que lo que sigue identifica una clase del sistema o una clase cargada. El prefijo `math:` de las expresiones XPath asocia las funciones de extensión al URI (y, por extensión, a la clase) `System.Math`. Las funciones de extensión identifican métodos en la clase `System.Math` y presenta argumentos cuando es necesario.

Ejemplo de código XQuery

Aquí puede ver un fragmento de código XQuery similar al ejemplo anterior:

```
<math xmlns:math="clitype:System.Math">
   {math:Sqrt(9)}
</math>
```

Tal y como ocurre con el código XSLT anterior, la declaración de espacio de nombres identifica la clase .NET, en este caso una clase del sistema. La expresión XQuery identifica el método al que se debe llamar y presenta el argumento.

30.2.2.2.1 Constructores

Una función de extensión se puede usar para llamar a un constructor .NET. A todos los constructores se les llama con la pseudofunción `new()`. Si hay más de un constructor para una clase, entonces se selecciona el constructor que más se ajusta al número de argumentos suministrados. Si no se encuentra ningún constructor que coincida con los argumentos suministrados, entonces se genera el error "`No constructor found`".

Constructores que devuelven tipos de datos XPath/XQuery

Si el resultado de una llamada a un constructor .NET se puede convertir de forma implícita en tipos de datos XPath/XQuery 2223, entonces la función de extensión .NET devuelve una secuencia que es un tipo de datos XPath/XQuery.

Constructores que devuelven objetos .NET

Si el resultado de una llamada a un constructor .NET no se puede convertir a un tipo de datos XPath/XQuery adecuado, entonces el constructr crea un objeto .NET contenido con un tipo que es el nombre de la clase que devuelve dicho objeto. Por ejemplo, si se llama al constructor para la clase `System.DateTime` (con `System.DateTime.new()`), entonces se devuelve un objeto que tiene un tipo `System.DateTime`.

Puede que el formato léxico del objeto devuelto no coincida con el formato léxico de un tipo de datos XPath. En estos casos, el valor devuelto (i) debe convertirse al formato léxico del tipo de datos XPath pertinente y (ii) debe convertirse en el tipo de datos XPath necesario.

Se pueden hacer tres cosas con un objeto .NET creado con un constructor:

- Se puede usar dentro de una variable:
```
<xsl:variable name="currentdate" select="date:new(2008, 4, 29)"
xmlns:date="clitype:System.DateTime" />
```
- Se puede pasar a una función de extensión (ver *Métodos de instancia y campos de instancia* 2221):
```
<xsl:value-of select="date:ToString(date:new(2008, 4, 29))"
xmlns:date="clitype:System.DateTime" />
```
- Se puede convertir en un tipo de cadena, numérico o booleano:
```
<xsl:value-of select="xs:integer(date:get_Month(date:new(2008, 4, 29)))"
xmlns:date="clitype:System.DateTime" />
```

30.2.2.2.2 Metodos estáticos y campos estáticos

La llamada a un método estático la hace directamente su nombre y se hace presentando los argumentos para el método. El nombre usado en la llamada debe ser el mismo que un método estático público de la clase especificada. Si el nombre del método y el número de argumentos que se dio en la llamada a función coincide con algún método de la clase, entonces los tipos de los argumentos presentados se evalúan para encontrar el resultado ideal. Si no se encuentra ninguna coincidencia, se emite un error.

Nota: un campo de una clase .NET se trata como si fuera un método sin argumentos. Para llamar a una propiedad se usa la sintaxis `get_nombrePropiedad()`.

© 2016-2022 Altova GmbH

Ejemplos

Este ejemplo de código XSLT muestra una llamada a un método con un argumento (`System.Math.Sin(arg)`):

```
<xsl:value-of select="math:Sin(30)" xmlns:math="clitype:System.Math"/>
```

Este ejemplo de código XSLT muestra una llamada a un campo (que se trata como si fuera un método sin argumentos) (`System.Double.MaxValue()`):

```
<xsl:value-of select="double:MaxValue()" xmlns:double="clitype:System.Double"/>
```

Este ejemplo de código XSLT muestra una llamada a una propiedad (la sintaxis es `get_nombrePropiedad()`) (`System.String()`):

```
<xsl:value-of select="string:get_Length('my string')"
xmlns:string="clitype:System.String"/>
```

Este ejemplo de código XQuery muestra una llamada a un método con un argumento (`System.Math.Sin(arg)`):

```
<sin xmlns:math="clitype:System.Math">
   { math:Sin(30) }
</sin>
```

30.2.2.2.3 Métodos de instancia y campos de instancia

Un método de instancia es un método al que se le pasa un objeto .NET como primer argumento de la llamada al método. Este objeto .NET se suele crear usando una función de extensión (por ejemplo, una llamada a un constructor) o un parámetro o una variable de una hoja de estilos. Un ejemplo de código XSLT para este tipo de método sería:

```
<xsl:stylesheet version="2.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:fn="http://www.w3.org/2005/xpath-functions">
   <xsl:output method="xml" omit-xml-declaration="yes"/>
   <xsl:template match="/">
      <xsl:variable name="releasedate"
         select="date:new(2008, 4, 29)"
         xmlns:date="clitype:System.DateTime"/>
      <doc>
         <date>
            <xsl:value-of select="date:ToString(date:new(2008, 4, 29))"
               xmlns:date="clitype:System.DateTime"/>
         </date>
         <date>
            <xsl:value-of select="date:ToString($releasedate)"
```

```
                xmlns:date="clitype:System.DateTime"/>
            </date>
        </doc>
    </xsl:template>
</xsl:stylesheet>
```

En el ejemplo anterior, se usó un constructor `System.DateTime` (`new(2008, 4, 29)`) para crear un objeto .NET de tipo `System.DateTime`. Este objeto se creó dos veces, una vez como valor de la variable `releasedate`, y otra vez como primer y único argumento del método `System.DateTime.ToString()`. Al método de instancia `System.DateTime.ToString()` se le llama dos veces, ambas con el constructor `System.DateTime` (`new(2008, 4, 29)`) como primer y único argumento. En una de estas instancias, se usó la variable `releasedate` para obtener el objeto .NET.

Métodos de instancia y campos de instancia

La diferencia entre un método de instancia y un campo de instancia es solo teórica. En un método de instancia, se pasa directamente un objeto .NET como argumento. En un campo de instancia, se pasa un parámetro o una variable (aunque el parámetro o la variable puede contener un objeto .NET). Por ejemplo, en el código del ejemplo anterior, la variable `releasedate` contiene un objeto .NET y esta es la variable que se pasa como argumento de `ToString()` en el segundo constructor de elemento `date`. Por tanto, la instancia `ToString()` del primer elemento `date` es un método de instancia, mientras que la segunda se considera un campo de instancia. El resultado es el mismo en ambos casos.

30.2.2.2.4 Tipos de datos: conversión de XPath/XQuery en .NET

Cuando se usa una función de extensión .NET dentro de una expresión XPath/XQuery, los tipos de datos de los argumentos de la función son importantes para determinar a cuál de los métodos .NET que tienen el mismo nombre se está llamando.

En .NET se siguen estas normas:

- Si en una clase hay varios métodos que tienen el mismo nombre, solamente se pueden seleccionar los métodos que tienen el mismo número de argumentos que la llamada a función.
- Los tipos de datos de cadena, numéricos y booleanos XPath/XQuery (*ver lista más abajo*) se convierten de forma implícita en el tipo de datos .NET correspondiente. Si el tipo XPath/XQuery suministrado se puede convertir en más de un tipo .NET (p. ej. `xs:integer`), entonces se selecciona el tipo .NET que se declaró para el método seleccionado. Por ejemplo, si el método .NET al que se está llamando es `fx(double)` y el tipo de datos XPath/XQuery suministrado es `xs:integer`, entonces se convierte `xs:integer` en el tipo de datos .NET `double`.

La tabla que aparece a continuación enumera las conversiones implícitas de los tipos de cadena, numéricos y booleanos XPath/XQuery en tipos de datos .NET.

`xs:string`	`StringValue, string`
`xs:boolean`	`BooleanValue, bool`
`xs:integer`	`IntegerValue, decimal, long, integer, short, byte, double, float`

© 2016-2022 Altova GmbH

xs:float	FloatValue, float, double
xs:double	DoubleValue, double
xs:decimal	DecimalValue, decimal, double, float

Los subtipos de los tipos de datos XML Schema de la tabla anterior (que se usan en XPath y XQuery) también se convierten en los tipos .NET correspondientes al tipo antecesor del subtipo.

En algunos casos quizás no sea posible seleccionar el método .NET correcto usando la información dada. Por ejemplo, imagine que:

- El argumento presentado es un valor `xs:untypedAtomic` de 10 y está destinado al método `mimétodo(float)`.
- Sin embargo, hay otro método en la clase que toma un argumento de otro tipo de datos: `mimétodo(double)`.
- Puesto que los métodos tienen el mismo nombre y el tipo suministrado (`xs:untypedAtomic`) se puede convertir correctamente tanto en `float` como en `double`, es posible que `xs:untypedAtomic` se convierta en `double` en lugar de en `float`.
- Por consiguiente, el método seleccionado no será el método necesario y puede que no produzca el resultado esperado. Una solución es crear un método definido por el usuario con un nombre diferente y usar ese método.

Los tipos que no aparecen en la lista anterior (p. ej. `xs:date`) no se convertirán y generarán un error.

30.2.2.2.5 Tipos de datos: conversión de .NET en XPath/XQuery

Cuando un método .NET devuelve un valor y el tipo de datos del valor es un tipo de cadena, numérico o booleano, entonces se convierte en el tipo de datos XPath/XQuery correspondiente. Por ejemplo, el tipo de datos .NET `decimal` se convierte en `xsd:decimal`.

Cuando se devuelve un objeto .NET o un tipo de datos que no es de cadena, numérico ni booleano, puede garantizar la conversión del tipo XPath/XQuery necesario usando primero un método .NET (p. ej. `System.DateTime.ToString()`) para convertir el objeto .NET en una cadena. En XPath/XQuery la cadena se puede modificar para ajustarse a la representación léxica del tipo necesario y convertirse después en dicho tipo (usando la expresión `cast as`, por ejemplo).

30.2.2.3 Scripts MSXSL para XSLT

El elemento `<msxsl:script>` contiene funciones y variables definidas por el usuario a las que se puede llamar desde dentro de expresiones XPath en la hoja de estilos XSLT. El elemento `<msxsl:script>` es un elemento de nivel superior, es decir, debe ser un elemento secundario de `<xsl:stylesheet>` o `<xsl:transform>`.

El elemento `<msxsl:script>` debe estar en el espacio de nombres `urn:schemas-microsoft-com:xslt` (*ver ejemplo más abajo*).

Lenguaje de scripting y espacio de nombres

El lenguaje de scripting utilizado dentro del bloque se especifica en el atributo `language` del elemento `<msxsl:script>` y el espacio de nombres que se debe usar para las llamadas a función desde expresiones XPath se identifica con el atributo `implements-prefix`:

```
<msxsl:script language="lenguaje-de-scripting" implements-prefix="prefijo-espacioNombres-usuario">

   función-1 o variable-1
   ...
   función-n o variable-n

</msxsl:script>
```

El elemento `<msxsl:script>` interactúa con Windows Scripting Runtime, de modo que dentro del elemento `<msxsl:script>` solamente se pueden usar lenguajes que estén instalados en el equipo. Para poder usar scripts MSXSL es necesario tener **instalada la plataforma .NET Framework 2.0 (o superior)**. Por tanto, los lenguajes de scripting .NET se pueden usar dentro del elemento `<msxsl:script>`.

El atributo `language` admite los mismos valores que el atributo `language` del elemento HTML `<script>`. Si no se especifica el atributo `language`, entonce se asume Microsoft JScript por defecto.

El atributo `implements-prefix` toma un valor que es un prefijo de un espacio de nombres declarado dentro del ámbito. Este espacio de nombres suele ser un espacio de nombres de usuario que se reservó para una biblioteca de funciones. Todas las funciones y variables definidas dentro del elemento `<msxsl:script>` están en el espacio de nombres identificado por el prefijo indicado en el atributo `implements-prefix`. Cuando se llama a una función desde dentro de una expresión XPath, el nombre de función completo debe estar en el mismo espacio de nombres que la definición de función.

Ejemplo

Aquí puede ver un ejemplo de una hoja de estilos XSLT que usa una función definida dentro de un elemento `<msxsl:script>`.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:fn="http://www.w3.org/2005/xpath-functions"
   xmlns:msxsl="urn:schemas-microsoft-com:xslt"
   xmlns:user="http://mycompany.com/mynamespace">

  <msxsl:script language="VBScript" implements-prefix="user">
    <![CDATA[
    ' Input: A currency value: the wholesale price
    ' Returns: The retail price: the input value plus 20% margin,
    ' rounded to the nearest cent
    dim a as integer  = 13
    Function AddMargin(WholesalePrice) as integer
```


© 2016-2022 Altova GmbH

```
      AddMargin = WholesalePrice * 1.2 + a
    End Function
  ]]>
  </msxsl:script>

  <xsl:template match="/">
    <html>
      <body>
        <p>
          <b>Total Retail Price =
            $<xsl:value-of select="user:AddMargin(50)"/>
          </b>
          <br/>
          <b>Total Wholesale Price =
            $<xsl:value-of select="50"/>
          </b>
        </p>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Tipos de datos

Los valores de los parámetros que se pasan dentro y fuera del bloque de script solamente pueden ser tipos de datos XPath. Esta restricción no afecta a los datos que se pasan las funciones y variables situadas dentro del bloque de script.

Ensamblados

Puede importar un ensamblado al script usando el elemento `msxsl:assembly`. El ensamblado se identifica con un nombre o un URI. El ensamblado se importa cuando se compila la hoja de estilos. Aquí puede ver cómo se usa el elemento `msxsl:assembly`:

```
<msxsl:script>
       <msxsl:assembly name="miEnsamblado.nombreEnsamblado" />
       <msxsl:assembly href="rutaDelEnsamblado" />

       ...

</msxsl:script>
```

El nombre de ensamblado puede ser un nombre completo, como:

```
"system.Math, Version=3.1.4500.1 Culture=neutral PublicKeyToken=a46b3f648229c514"
```

o un nombre abreviado, como `"miEnsamblado.Draw"`.

Espacios de nombres

Puede declarar espacios de nombres con el elemento `msxsl:using`. Esto permite escribir las clases del ensamblado en el script sin sus espacios de nombres, lo cual le permitirá ahorrar mucho tiempo. Aquí puede ver cómo se usa el elemento `msxsl:using` para declarar espacios de nombres.

```
<msxsl:script>
       <msxsl:using namespace="ENmiEnsamblado.NombreEspaciodenombres" />

       ...

</msxsl:script>
```

El valor del atributo `namespace` es el nombre del espacio de nombres.

© 2016-2022 Altova GmbH

30.3 Tipos de datos en esquemas XML generados a partir de BD

Cuando se genera un esquema XML a partir de una base de datos (BD), los tipos de datos propios de esa BD se convierten en tipos de datos XML Schema. En este anexo describimos las correspondencias entre tipos de datos de BD y tipos de datos XML Schema para las bases de datos más utilizadas:

- ADO 2233
- MS Access 2234
- MS SQL Server 2235
- MySQL 2236
- ODBC 2236
- Oracle 2237
- Sybase 2238

30.3.1 ADO

Cuando se genera un esquema XML a partir de una base de datos ADO, los tipos de datos ADO se convierten en tipos de datos XML Schema tal y como muestra la tabla que aparece a continuación:

Tipo de datos ADO	Tipo de datos XML Schema
adGUID	xs:ID
adChar	xs:string
adWChar	xs:string
adVarChar	xs:string
adWVarChar	xs:string
adLongVarChar	xs:string
adWLongVarChar	xs:string
adVarWChar	xs:string
adBoolean	xs:boolean
adSingle	xs:float
adDouble	xs:double
adNumeric	xs:decimal
adCurrency	xs:decimal
adDBTimeStamp	xs:dateTime
adDate	xs:date
adBinary	xs:base64Binary

adVarBinary	xs:base64Binary
adLongVarBinary	xs:base64Binary
adInteger	xs:Integer
adUnsignedInt	xs:unsignedInt
adSmallInt	xs:short
adUnsignedSmallInt	xs:unsignedShort
adBigInt	xs:long
adUnsignedBigInt	xs:unsignedLong
adTinyInt	xs:byte
adUnsignedTinyInt	xs:unsignedByte

30.3.2 MS Access

Cuando se genera un esquema XML a partir de una base de datos MS Access, los tipos de datos MS Access se convierten en tipos de datos XML Schema tal y como muestra la tabla que aparece a continuación:

Tipo de datos MS Access	Tipo de datos XML Schema
GUID	xs:ID
char	xs:string
varchar	xs:string
memo	xs:string
bit	xs:boolean
Number(single)	xs:float
Number(double)	xs:double
Decimal	xs:decimal
Currency	xs:decimal
Date/Time	xs:dateTime
Number(Entero largo)	xs:integer
Number(Entero)	xs:short
Number(Byte)	xs:byte
OLE Object	xs:base64Binary

© 2016-2022 Altova GmbH

30.3.3 MS SQL Server

Cuando se genera un esquema XML a partir de una base de datos MS SQL Server, los tipos de datos MS SQL Server se convierten en tipos de datos XML Schema tal y como muestra la tabla que aparece a continuación:

Tipo de datos MS SQL Server	Tipo de datos XML Schema
uniqueidentifier	xs:ID
char	xs:string
nchar	xs:string
varchar	xs:string
nvarchar	xs:string
text	xs:string
ntext	xs:string
sysname	xs:string
bit	xs:boolean
real	xs:float
float	xs:double
decimal	xs:decimal
money	xs:decimal
smallmoney	xs:decimal
datetime	xs:dateTime
smalldatetime	xs:dateTime
binary	xs:base64Binary
varbinary	xs:base64Binary
image	xs:base64Binary
integer	xs:integer
smallint	xs:short
bigint	xs:long
tinyint	xs:byte

30.3.4 MySQL

Cuando se genera un esquema XML a partir de una base de datos MySQL, los tipos de datos MySQL se convierten en tipos de datos XML Schema tal y como muestra la tabla que aparece a continuación:

Tipo de datos MySQL	Tipo de datos XML Schema
char	xs:string
varchar	xs:string
text	xs:string
tinytext	xs:string
mediumtext	xs:string
longtext	xs:string
tinyint(1)	xs:boolean
float	xs:float
double	xs:double
decimal	xs:decimal
datetime	xs:dateTime
blob	xs:base64Binary
tinyblob	xs:base64Binary
mediumblob	xs:base64Binary
longblob	xs:base64Binary
smallint	xs:short
bigint	xs:long
tinyint	xs:byte

30.3.5 ODBC

Cuando se genera un esquema XML a partir de una base de datos ODBC, los tipos de datos ODBC se convierten en tipos de datos XML Schema tal y como muestra la tabla que aparece a continuación:

Tipo de datos ODBC	Tipo de datos XML Schema
SQL_GUID	xs:ID
SQL_CHAR	xs:string

© 2016-2022 Altova GmbH

SQL_VARCHAR	xs:string
SQL_LONGVARCHAR	xs:string
SQL_BIT	xs:boolean
SQL_REAL	xs:float
SQL_DOUBLE	xs:double
SQL_DECIMAL	xs:decimal
SQL_TIMESTAMP	xs:dateTime
SQL_DATE	xs:date
SQL_BINARY	xs:base64Binary
SQL_VARBINARY	xs:base64Binary
SQL_LONGVARBINARY	xs:base64Binary
SQL_INTEGER	xs:integer
SQL_SMALLINT	xs:short
SQL_BIGINT	xs:long
SQL_TINYINT	xs:byte

30.3.6 Oracle

Cuando se genera un esquema XML a partir de una base de datos Oracle, los tipos de datos Oracle se convierten en tipos de datos XML Schema tal y como muestra la tabla que aparece a continuación:

Tipo de datos Oracle	Tipo de datos XML Schema
ROWID	xs:ID
CHAR	xs:string
NCHAR	xs:string
VARCHAR2	xs:string
NVARCHAR2	xs:string
CLOB	xs:string
NCLOB	xs:string
NUMBER (cuando tiene aplicada una restricción de comprobación)*	xs:boolean
NUMBER	xs:decimal

FLOAT	xs:double
DATE	xs:dateTime
INTERVAL YEAR TO MONTH	xs:gYearMonth
BLOB	xs:base64Binary

* Si se aplica una restricción de comprobación a una columna de tipo de datos NUMBER y la restricción de comprobación busca los valores 0 o 1, entonces el tipo de datos NUMBER para esta columna se convierte en el tipo de datos XML Schema xs:boolean. Este mecanismo es muy práctico a la hora de generar un tipo de datos xs:boolean en el esquema XML generado.

30.3.7 Sybase

Cuando se genera un esquema XML a partir de una base de datos Sybase, los tipos de datos Sybase se convierten en tipos de datos XML Schema tal y como muestra la tabla que aparece a continuación:

Tipo de datos Sybase	Tipo de datos XML Schema
char	xs:string
nchar	xs:string
varchar	xs:string
nvarchar	xs:string
text	xs:string
sysname-varchar(30)	xs:string
bit	xs:boolean
real	xs:float
float	xs:float
double	xs:double
decimal	xs:decimal
money	xs:decimal
smallmoney	xs:decimal
datetime	xs:dateTime
smalldatetime	xs:dateTime
timestamp	xs:dateTime
binary<=255	xs:base64Binary
varbinary<=255	xs:base64Binary

© 2016-2022 Altova GmbH

image	xs:base64Binary
integer	xs:integer
smallint	xs:short
tinyint	xs:byte

30.4 Tipos de datos en BD generadas a partir de esquemas XML

Cuando se genera una estructura de BD a partir de un esquema XML, los tipos de datos propios de esa BD se generan a partir de los tipos de datos del esquema XML. En este anexo describimos las correspondencias entre tipos de datos XML Schema y tipos de datos de BD para las bases de datos más utilizadas:

- MS Access 2240
- MS SQL Server 2242
- MySQL 2244
- Oracle 2246

30.4.1 MS Access

Cuando se crea una base de datos MS Access a partir de un esquema XML, los tipos de datos XML Schema se convierten en tipos de datos MS Access tal y como muestra la tabla que aparece a continuación:

Tipo de datos XML Schema	Tipo de datos MS Access
xs:ID	GUID
xs:string	If no facets varchar (255)
	Size = either length or maxLength
	If Size <= 255 varchar (n)
	else memo
xs:normalizedString	Igual que xs:string
xs:token	Igual que xs:string
xs:Name	Igual que xs:string
xs:NCName	Igual que xs:string
xs:anyURI	Igual que xs:string
xs:QName	Igual que xs:string
xs:NOTATION	Igual que xs:string
xs:boolean	bit
xs:float	Number (single)
xs:double	Number (double)
xs:decimal	Decimal

© 2016-2022 Altova GmbH

xs:duration	Date/Time
xs:dateTime	Date/Time
xs:time	Date/Time
xs:date	Date/Time
xs:gYearMonth	Date/Time
xs:gYear	Date/Time
xs:gMonthDay	Date/Time
xs:gDay	Date/Time
xs:gMonth	Date/Time
xs:hexBinary	If no facets varbinary (255)
	Size = either length or maxLength
	If Size <= 8000 varbinary
	else image (OLE Object)
xs:base64Binary	Igual que xs:hexBinary
xs:integer	Number (Entero largo)
xs:int	Number (Entero largo)
xs:negativeInteger	Number (Entero largo); restricción de valor
xs:positiveInteger	Number (Entero largo); restricción de valor
xs:nonNegativeInteger	Number (Entero largo); restricción de valor
xs:nonPositiveInteger	Number (Entero largo); restricción de valor
xs:unsignedInt	Number (Entero largo)
xs:short	-- sin equivalente --
xs:unsignedShort	-- sin equivalente --
xs:long	-- sin equivalente --
xs:unsignedLong	-- sin equivalente --
xs:byte	Number (Byte)
xs:unsignedByte	Number (Byte)

30.4.2 MS SQL Server

Cuando se crea una base de datos MS SQL Server a partir de un esquema XML, los tipos de datos XML Schema se convierten en tipos de datos MS SQL Server tal y como muestra la tabla que aparece a continuación:

Tipo de datos XML Schema	Tipo de datos MS SQL Server
ID	uniqueidentifier
xs:string	If no facets
	{ if UNICODE nvarchar (255)
	else varchar (255) }
	else
	{ if UNICODE
	(Size = either length or maxLength)
	If Size <= 4000
	if FacetLengthIsSet then nChar
	else nVarChar
	if Size <= 1073741823 then nText }
	else
	{ if NON-UNICODE
	(Size = either length or maxLength)
	If Size <= 8000
	if FacetLengthIsSet then char
	else varchar
	if Size <= 2147483647 then text }
xs:normalizedString	Igual que xs:string
xs:token	Igual que xs:string
xs:Name	Igual que xs:string
xs:NCName	Igual que xs:string
xs:anyURi	Igual que xs:string
xs:QName	Igual que xs:string
xs:NOTATION	Igual que xs:string
xs:boolean	bit
xs:float	real

© 2016-2022 Altova GmbH

xs:double	float
xs:decimal	decimal
xs:duration	datetime
xs:dateTime	datetime
xs:time	datetime
xs:date	datetime
xs:gYearMonth	datetime
xs:gYear	datetime
xs:gMonthDay	datetime
xs:gDay	datetime
xs:gMonth	datetime
xs:hexBinary	If no facets varbinary (255)
	(Size = either length or maxLength
	If Size <= 8000
	if FacetLengthIsSet then binary
	else varbinary
	if Size <= 2147483647 then image
xs:base64Binary	Igual que xs:hexBinary
xs:integer	int
xs:int	int
xs:negativeInteger	Int (limitado a {...,-2,-1})
xs:positiveInteger	Int (limitado a {1,2,...})
xs:nonNegativeInteger	int (limitado a {0,1,2,...})
xs:nonPositiveInteger	int (limitado a {...,-2,-1,0})
xs:unsignedInt	int (restricciones adicionales)
xs:short	smallint
xs:unsignedShort	smallint (restricciones adicionales)
xs:long	bigint
xs:unsignedLong	bigint (restricciones adicionales)
xs:byte	tinyint
xs:unsignedByte	tinyint (restricciones adicionales)

30.4.3 MySQL

Cuando se crea una base de datos MySQL a partir de un esquema XML, los tipos de datos XML Schema se convierten en tipos de datos MySQL tal y como muestra la tabla que aparece a continuación:

Tipo de datos XML Schema	Tipo de datos MySQL
xs:ID	varchar(255)
xs:string	If no facets then varchar (255)
	else if facet length is set and <= 255 then char
	else if facet maxLength set and <= 255 then varchar
	else if maxLength is set and <= 65545 then text
	else if maxlength is set and <= 16777215 then mediumtext
	else if maxlength is set and <= 429496295 then longtext
xs:normalizedString	Igual que xs:string
xs:token	Igual que xs:string
xs:Name	Igual que xs:string
xs:NCName	Igual que xs:string
xs:anyURI	Igual que xs:string
xs:QName	Igual que xs:string
xs:NOTATION	Igual que xs:string
xs:boolean	tinyint(1)
xs:float	float
xs:double	double
xs:decimal	decimal
xs:duration	timestamp
xs:dateTime	datetime
xs:time	time
xs:date	date
xs:gYearMonth	timestamp(4)
xs:gYear	year(4)

© 2016-2022 Altova GmbH

xs:gMonthDay	timestamp(8); constraints to check month, day
xs:gDay	timestamp(8); constraints to check day
xs:gMonth	timestamp(8); constraints to check month
xs:hexBinary	If no facets then blob (255)
	else if facet length is set and <= 255 then blob
	else if facet maxLength is set and <= 255 then tinyblob
	else if maxlength is set and <= 65545 then blob
	else if maxlength is set and <= 16777215 then mediumblob
	else if maxlength is set and <= 429496295 then longblob
xs:base64Binary	Igual que xs:hexBinary
xs:integer	Integer
xs:int	int
xs:negativeInteger	Integer (limitado a {...,-2,-1})
xs:positiveInteger	Integer (limitado a {1,2,...})
xs:nonNegativeInteger	Integer (limitado a {0,1,2,...})
xs:nonPositiveInteger	Integer (limitado a {...,-2,-1,0})
xs:unsignedInt	Int (restricciones adicionales)
xs:short	Smallint
xs:unsignedShort	Smallint (restricciones adicionales)
xs:long	Bigint
xs:unsignedLong	Bigint (restricciones adicionales)
xs:byte	Tinyint
xs:unsignedByte	Tinyint (restricciones adicionales)

30.4.4 Oracle

Cuando se crea una base de datos Oracle a partir de un esquema XML, los tipos de datos XML Schema se convierten en tipos de datos Oracle tal y como muestra la tabla que aparece a continuación:

Tipo de datos XML Schema	Tipo de datos Oracle
xs:ID	ROWID
xs:string	If no facets
	if UNICODE then NVARCHAR2 (255)
	else VARCHAR2 (255)
	else if UNICODE
	(Size = either length or maxLength)
	If Size <= 2000 then NCHAR
	if Size <= 4000 then NVARHCAR2
	if Size <= 4 Gigabytes then NCLOB
	else if NON-UNICODE
	(Size = either length or maxLength)
	If Size <= 2000 then CHAR
	if Size <= 4000 then VARCHAR2
	if Size <= 4 Gigabytes then CLOB
xs:normalizedString	Igual que xs:string
xs:token	Igual que xs:string
xs:Name	Igual que xs:string
xs:NCName	Igual que xs:string
xs:anyURI	Igual que xs:string
xs:QName	Igual que xs:string
xs:NOTATION	Igual que xs:string
xs:boolean	NUMBER with constraint Boolean
xs:float	FLOAT
xs:double	FLOAT
xs:decimal	NUMBER
xs:duration	TIMESTAMP
xs:dateTime	TIMESTAMP

© 2016-2022 Altova GmbH

xs:time	DATE
xs:date	DATE
xs:gYearMonth	INTERVAL YEAR TO MONTH
xs:gYear	DATE
xs:gMonthDay	DATE
xs:gDay	DATE
xs:gMonth	DATE
xs:hexBinary	if no facets then RAW (255)
	(Size = either length or maxLength)
	If Size <= 2000 then RAW (X)
	else Size <= 2 Gigabytes then LONG RAW (X)
	if Size <= 4 Gigabytes then BLOB (X)
xs:base64Binary	BLOB
xs:integer	NUMBER
xs:int	NUMBER
xs:negativeInteger	NUMBER (limitado a {...,-2,-1})
xs:positiveInteger	NUMBER (limitado a {1,2,...})
xs:nonNegativeInteger	NUMBER (limitado a {0,1,2,...})
xs:nonPositiveInteger	NUMBER (limitado a {...,-2,-1,0})
xs:unsignedInt	NUMBER (restricciones adicionales)
xs:short	NUMBER
xs:unsignedShort	NUMBER (restricciones adicionales)
xs:long	NUMBER
xs:unsignedLong	NUMBER (restricciones adicionales)
xs:byte	BLOB
xs:unsignedByte	BLOB (restricciones adicionales)

30.5 Datos técnicos

Esta sección incluye información sobre algunos de aspectos técnicos de su software. La información está organizada en varios apartados:

- Requisitos de OS y memoria 2248
- Motores de Altova 2248
- Compatibilidad con Unicode 2249
- Uso de Internet 2249

30.5.1 Requisitos de SO y memoria

Sistema operativo

Las aplicaciones de software de Altova están disponibles en estas plataformas:

- Windows 7 SP1 con actualización de la plataforma, Windows 8, Windows 10, Windows 11
- Windows Server 2008 R2 SP1 con actualización de la plataforma o superior

Memoria

Puesto que el software está escrito en C++ no necesita tanto espacio como un JRE y suele necesitar menos memoria que otras aplicaciones similares basadas en Java. No obstante, todos los documentos se cargan en memoria por completo, para poder analizarlos completamente y mejorar la velocidad de visualización y edición. Los requisitos de memoria, por consiguiente, aumentan en función del tamaño del documento.

Los requisitos de memoria también vienen dados por el historial de operaciones Deshacer. Cuando se cortan y pegan secciones grandes de documentos de gran tamaño, la memoria disponible se puede agotar rápidamente.

30.5.2 Motores XSLT y XQuery de Altova

Validador XML

Al abrir un documento XML, la aplicación usa su validador XML integrado para comprobar si el formato es correcto, para validar el documento en relación a un esquema (si se ha especificado uno) y para generar estructuras y conjuntos de información (infosets). El validador XML también se usa para proporcionar ayuda de edición inteligente cuando usted modifique documentos y para mostrar de forma dinámica cualquier error de validación que ocurra.

El validador XML integrado implementa la recomendación final para las especificaciones 1.0 y 1.1 del esquema de XML del W3C. Altova incorpora continuamente las recomendaciones más recientes del Grupo de Trabajo del esquema de XML del W3C a su validador XML para que los productos de Altova siempre ofrezcan el entorno de desarrollo más avanzado.

© 2016-2022 Altova GmbH

Motores XSLT y XQuery

Los productos de Altova usan los motores XSLT 1.0, XSLT 2.0, XSLT 3.0, XQuery 1.0 y XQuery 3.1 de Altova. Si alguno de estos motores está incluido en el producto, encontrará documentación específica sobre el comportamiento de cada motor en la implementación en los anexos.

Nota: Altova MapForce genera código con los motores XSLT 1.0, XSLT 2.0 y XQuery 1.0.

30.5.3 Compatibilidad con Unicode

Los productos XML de Altova son completamente compatibles con Unicode. Para editar un documento XML también necesitará una fuente compatible con los caracteres Unicode utilizados por el documento.

Tenga en cuenta que la mayoría de las fuentes contienen solamente un subconjunto muy concreto de caracteres Unicode y, por tanto, están destinadas a un sistema de escritura concreto. Si algunos caracteres aparecen desfigurados, el motivo puede ser que la fuente seleccionada no contiene los glifos necesarios. Por tanto, es recomendable tener una fuente que abarque todos los caracteres Unicode. Sobre todo si edita documentos XML en varios idiomas o sistemas de escritura. Una fuente Unicode que suele venir con los equipos Windows es la fuente Arial Unicode MS.

En la carpeta `/Examples` de la carpeta de su aplicación puede encontrar un archivo XHTML llamado `UnicodeUTF-8.html` que incluye esta frase en gran número de idiomas y sistemas de escritura diferentes:

- *When the world wants to talk, it speaks Unicode*
- *Cuando el mundo quiere conversar, habla Unicode*
- *Wenn die Welt miteinander spricht, spricht sie Unicode*
- 世界的に話すなら、Unicode です。

Abra este archivo XHTML y observe el potencial de Unicode.

30.5.4 Uso de Internet

Las aplicaciones de Altova inician conexiones a Internet en estos casos:

- Si hace clic en el botón **Solicitar una clave de evaluación GRATUITA** del cuadro de diálogo "Activación del software" (**Ayuda | Activación del software**), los campos del cuadro de diálogo de activación del software se transfieren a nuestro servidor web por medio de una conexión HTTP corriente (puerto 80) y le enviamos el código de evaluación gratuito por correo electrónico.
- En algunos productos de Altova puede abrir un archivo por Internet (**Archivo | Abrir | Cambiar a URL**). En este caso, el documento se recupera usando uno de estos protocolos y conexiones: HTTP (normalmente por el puerto 80), FTP (normalmente por el puerto 20/21) o HTTPS (normalmente por el puerto 443). También puede ejecutar un servidor HTTP en el puerto 8080. (En el cuadro de diálogo "Abrir URL", después del nombre de servidor escriba dos puntos y el número de puerto.)
- Si abre un documento XML que hace referencia a un documento DTD o esquema XML y el documento se especifica a través de una URL, el documento de esquema al que se hace referencia también se recupera a través de una conexión HTTP (puerto 80) o cualquier otro protocolo (ver punto anterior). El documento de esquema también se recupera para validar el archivo XML. Recuerde que la validación

puede realizarse automáticamente nada más abrir el documento, si seleccionó esta opción en la sección *Archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).

- En las aplicaciones de Altova que trabajen con WSDL y SOAP, las conexiones a servicios web son definidas por documentos WSDL.
- Si usa el comando **Archivo | Enviar por correo electrónico** de XMLSpy, el texto seleccionado actualmente o el archivo se envía con el programa de correo electrónico instalado en el equipo.
- Durante la activación del software y la búsqueda de actualizaciones, tal y como se describe en el contrato de licencia de software de Altova.

© 2016-2022 Altova GmbH

30.6 Información sobre licencias

En esta sección encontrará información sobre:

- la distribución de este producto de software
- la activación del software y medición de licencias
- el contrato de licencia para el usuario final que rige el uso de este producto de software

Los términos del contrato de licencia que aceptó al instalar el producto de software son vinculantes, por lo que rogamos lea atentamente toda esta información.

Para leer los términos y condiciones de cualquiera de las licencias de Altova, consulte la página de información legal de Altova en el sitio web de Altova.

30.6.1 Distribución electrónica de software

Este producto está disponible por distribución electrónica de software, un método de distribución que ofrece ventajas únicas:

- Puede evaluar el software de forma totalmente gratuita durante 30 días antes de decidir si compra el producto (*Nota: la licencia para Altova Mobile Together Designer es gratuita*).
- Si decide comprarlo, puede hacer un pedido en línea en el sitio web de Altova y conseguir en pocos minutos el software con licencia.
- Si realiza el pedido en línea, siempre recibirá la versión más reciente de nuestro software.
- El paquete de instalación del producto incluye un sistema de ayuda en pantalla al que se puede acceder desde la interfaz de la aplicación. La versión más reciente del manual del usuario está disponible en www.altova.com (i) en formato HTML y (ii) en formato PDF para descargar e imprimir si lo desea.

Período de evaluación de 30 días

Después de descargar el producto de software, puede probarlo de forma totalmente gratuita durante un plazo de 30 días. Pasados unos 20 días, el software empieza a recordarle que no tiene una licencia. El mensaje de aviso aparece una sola vez cada vez que se inicie la aplicación. Para seguir utilizando el programa una vez pasado el plazo de 30 días, deberá comprar una licencia permanente, que se entrega en forma de código clave. Para desbloquear el producto debe introducir ese código clave en el cuadro de diálogo "Activación del software".

Las licencias de los productos pueden comprarse directamente en la tienda en línea del sitio web de Altova.

Distribuir la versión de evaluación a otros usuarios de su organización

Si desea distribuir la versión de evaluación en la red de su compañía o si desea usarlo en un PC que no está conectado a Internet, solamente puede distribuir los programas de instalación (siempre y cuando no se modifiquen de forma alguna). Todo usuario que acceda al instalador debe solicitar su propio código clave de

evaluación (de 30 días). Una vez pasado este plazo de 30 días, todos los usuarios deben comprar también una licencia para poder seguir usando el producto.

30.6.2 Activación del software y medición de licencias

Durante el proceso de activación del software de Altova, puede que la aplicación utilice su red interna y su conexión a Internet para transmitir datos relacionados con la licencia durante la instalación, registro, uso o actualización del software a un servidor de licencias operado por Altova y para validar la autenticidad de los datos relacionados con la licencia y proteger a Altova de un uso ilegítimo del software y mejorar el servicio a los clientes. La activación es posible gracias al intercambio de datos de la licencia (como el sistema operativo, la dirección IP, la fecha y hora, la versión del software, el nombre del equipo, etc.) entre su equipo y el servidor de licencias de Altova.

Su producto incluye un módulo integrado de medición de licencias que le ayudará a evitar infracciones del contrato de licencia para el usuario final. Puede comprar una licencia de un solo usuario o de varios usuarios para el producto de software y el módulo de medición de licencias se asegura de que no se utiliza un número de licencias mayor al permitido.

Esta tecnología de medición de licencias usa su red de área local (LAN) para comunicarse con las instancias de la aplicación que se ejecutan en equipos diferentes.

Licencia de un solo usuario

Cuando se inicia la aplicación, se inicia el proceso de medición de licencias y el software envía un breve datagrama de multidifusión para averiguar si hay otras instancias del producto activas en otros equipos del mismo segmento de red al mismo tiempo. Si no recibe ninguna respuesta, la aplicación abre un puerto para escuchar a otras instancias de la aplicación.

Licencia de varios usuarios

Si se usa más de una instancia de la aplicación dentro de la misma red LAN, estas instancias se comunicarán entre ellas al iniciarse. Estas instancias intercambian códigos claves para que ayudarle a no sobrepasar por error el número máximo de licencias concurrentes. Se trata de la misma tecnología de medición de licencias que suele utilizarse en Unix y en otras herramientas de desarrollo de bases de datos. Gracias a ella puede comprar licencias de varios usuarios de uso concurrente a un precio razonable.

Las aplicaciones se diseñaron de tal modo que envían pocos paquetes pequeños de red y no cargan demasiado su red. Los puertos TCP/IP (2799) utilizados por su producto de Altova están registrados oficialmente en la IANA (*para más información consulte el sitio web de la IANA www.iana.org*) y nuestro módulo de medición de licencias es una tecnología probada y eficaz.

Si usa un servidor de seguridad, puede notar las comunicaciones del puerto 2799 entre los equipos que ejecutan los productos de Altova. Si quiere, puede bloquear ese tráfico, siempre y cuando esto no resulte en una infracción del contrato de licencia.

© 2016-2022 Altova GmbH

Nota sobre los certificados

Su aplicación de Altova contacta con el servidor de licencias de Altova (`link.altova.com`) vía HTTPS. Para esta comunicación, Altova usa un certificado SSL registrado. Si se reemplaza este certificado (por ejemplo, si lo reemplaza su departamento de informática o un organismo externo), entonces su aplicación de Altova le advertirá de que la conexión puede no ser segura. Si usa el certificado sustitutivo para iniciar la aplicación, lo hace por su cuenta y riesgo. Si ve un mensaje de advertencia de que la conexión puede no ser segura, compruebe el origen del certificado y consulte con su equipo técnico (que decidirán si se debe continuar con el reemplazo del certificado de Altova).

Si su organización necesita usar su propio certificado (por ejemplo, para monitorizar la comunicación hacia y desde equipos cliente), entonces recomendamos que instale en su red Altova LicenseServer, el software gratuito de gestión de licencias de Altova. Así, sus equipos cliente pueden seguir usando los certificados de su organización y AltovaLicenseServer puede usar el certificado de Altova cuando necesite comunicarse con Altova.

30.6.3 Contrato de licencia para el usuario final

- Encontrará el Contrato de licencia de Altova para el usuario final en: https://www.altova.com/legal/eula
- Encontrará la Política de privacidad de Altova en: https://www.altova.com/privacy

Índice

.

.docx, 281, 938
.NET,
diferencias con la versión independiente de XMLSpy, 1125
integración de XMLSpy con, 1123
y los depuradores de XMLSpy, 1128
.pptx, 281, 938
.xslx, 281, 938

2

2005R3,
modo de compatibilidad, 1239

A

Abrir,
archivo, 1262
Acceso rápido, 1569
asignar/eliminar, 1569
Acciones de la línea de comandos, 1638
Activar el software, 1633
ActiveX,
integration at application level, 2055
integration at document level, 2057
integration prerequisites, 2052
ActiveX controls,
adding to the Visual Studio Toolbox, 2053
Add Child command,
in Grid View, 1333
Adelante,
en la vista Esquema, 250
Administrador del control de código fuente, 1316
Administrar espacios en blanco, 337
ADO,
como interfaz de conexión de datos, 952
configurar una conexión, 958
conversión de tipos de datos en esquemas XML generados a partir de BD, 2233
ADO.NET,
configurar una conexión, 964
Ajuste automático de línea,
habilitar/deshabilitar, 1488
Ajuste automático de línea en la vista Texto, 146
Ajuste de línea,
habilitar/deshabilitar ajuste automático de línea, 1488
Alias,
ver Recursos globales, 1042
Almacenes de certificados, 419
Analizador,
XSLT, 1612
Ancho de celda máximo, 1597
Ancho de tabla óptimo, 1488, 1597
Anexar,
fila (en la vista Authentic), 1420
Anotaciones en la vista Esquema, 182
API,
documentation, 1686
JAVA, 2001
JAVA Classpath, 2001
overview, 1687
API COM,
en el editor de scripting, 1651
Aplicaciones externas,
abrir archivos en, 1568
Aplicar, 1583
Append command,
in Grid View, 1333
Application,
ActiveDocument, 1728
AddMacroMenuItem, 1728
AddXSLT_XQParameter, 1728
Application, 1728
ClearMacroMenu, 1729
CurrentProject, 1729
Dialogs, 1730
Documents, 1730
GetDatabaseImportElementList, 1731
GetDatabaseSettings, 1731
GetDatabaseTables, 1732
GetExportSettings, 1732
GetTextImportElementList, 1733
GetTextImportExportSettings, 1733
GetXSLT_XQParameterCount, 1734
GetXSLT_XQParameterName, 1734

© 2016-2022 Altova GmbH

Application,
GetXSLT_XQParameterXPath, 1734
ImportFromDatabase, 1735
ImportFromSchema, 1736
ImportFromText, 1736
ImportFromWord, 1737
NewProject, 1738
OnBeforeOpenDocument, 1725
OnBeforeOpenProject, 1726
OnDocumentOpened, 1727
OnProjectOpened, 1727
OpenProject, 1738
Parent, 1739
Quit, 1739
ReloadSettings, 1740
RemoveXSLT_XQParameter, 1740
RunMacro, 1740
ScriptingEnvironment, 1740
ShowApplication, 1741
ShowForm, 1742
URLDelete, 1742
URLMakeDirectory, 1742
WarningNumber, 1743
WarningText, 1743
Archivo, 1584
abrir, 1262
cerrar, 1268
codificación predeterminada, 1589
codificar, 1268
crear nuevo, 1258
guardar, 1269
guardar por correo electrónico, 1274
opciones de impresión, 1275
opciones para abrir, 1584
sección, 1584
Archivo CSV,
importar como XML, 1452
Archivo de metadatos, 314, 316, 696
Archivo de metadatos JSON para filtros y fórmulas, 696
Archivo de metadatos XML para filtros, 314
Archivo de metadatos XML para fórmulas, 316
Archivo de texto,
exportación de datos XML a, 1471
importar como XML, 1452
Archivo nuevo,
crear, 1258
Archivo PXF,
generar documentos de salida desde la vista Authentic, 1421
Archivo SPS,
asignar al archivo XML, 1413
Archivo XML,
generar a partir de la DTD o el esquema XML, 1365
Archivo XML de plantilla,
en la vista Authentic, 600
Archivo XML de recursos globales, 1043
Archivos,
agregarlos al control de código fuente, 1309
comparar dos, 1547
opciones de comparación, 1555
usados recientemente, 1278
Archivos binarios Avro, 773
Archivos de plantilla,
para documentos nuevos, 1258
Archivos DTD y catálogos, 456
Archivos EPUB, 945
Archivos no XML, 1587
Archivos usados recientemente,
lista de, 1278
Archivos WSDL,
Validación, 1343
Archivos XQuery,
definir extensiones de archivo en XMLSpy, 507
Archivos ZIP, 281, 938
crear en la vista Archivo, 943
edición en la vista Authentic, 943
Arrastrar y colocar en la vista Cuadrícula JSON, 689
Arrastrar y colocar en la vista Cuadrícula XML, 306
Aserciones en la vista Esquema, 212, 216
Asignación de tipos condicionales, 205
Asignar,
tecla de acceso rápido a un comando, 1569
asignar al documento XML,
agregar elementos con, 59
Los documentos DTD se pueden editar en la vista Texto y en la vista Cuadrícula., 442
Asignar hoja de estilos StyleVision Power Stylesheet al archivo XML, 1413
Asistente para crear taxonomías nuevas, 838
Atajos de teclado, 283
ATL,
archivos de muestra para complementos, 1673
Atrás,
en la vista Esquema, 250
Atributo, 77
activar o desactivar atributos en la vista del modelo de contenido, 77

© 2016-2022 Altova GmbH

Atributo, 77
en definiciones de esquema, 77
Atributos, 182
Atributos (ayudante de entrada),
en la vista Authentic, 619
Atributos en la vista Esquema, 212, 213
Atributos globales, 182
AttributeFormDefault,
configuración del esquema, 1370
Authentic XML, 597
AuthenticDataTransfer,
dropEffect, 1746
getData, 1746
ownDrag, 1746
type, 1747
AuthenticRange,
AppendRow, 1752
Application, 1752
CanPerformAction, 1753
CanPerformActionWith, 1753
Close, 1754
CollapsToBegin, 1754
CollapsToEnd, 1754
Copy, 1754
Cut, 1755
Delete, 1755
DeleteRow, 1755
DuplicateRow, 1756
ExpandTo, 1757
FirstTextPosition, 1757
FirstXMLData, 1758
FirstXMLDataOffset, 1759
GetElementAttributeNames, 1760
GetElementAttributeValue, 1760
GetElementHierarchy, 1760
GetEntityNames, 1762
Goto, 1762
GotoNext, 1763
GotoNextCursorPosition, 1763
GotoPrevious, 1764
GotoPreviousCursorPosition, 1764
HasElementAttribute, 1765
InsertEntity, 1765
InsertRow, 1766
IsCopyEnabled, 1766
IsCutEnabled, 1766
IsDeleteEnabled, 1767
IsEmpty, 1767
IsEqual, 1767
IsFirstRow, 1767
IsInDynamicTable, 1768
IsLastRow, 1768
IsPasteEnabled, 1768
IsTextStateApplied, 1769
LastTextPosition, 1769
LastXMLData, 1770
LastXMLDataOffset, 1770
MoveBegin, 1772
MoveEnd, 1772
MoveRowDown, 1773
MoveRowUp, 1772
Parent, 1773
Paste, 1773
PerformAction, 1773
Select, 1774
SelectNext, 1775
SelectPrevious, 1775
SetElementAttributeValue, 1776
SetFromRange, 1777
Text, 1778
AuthenticView, 1796
Application, 1789
AsXMLString, 1789
DocumentBegin, 1790
DocumentEnd, 1791
Event, 1792
Goto, 1793
IsRedoEnabled, 1793
IsUndoEnabled, 1794
MarkupVisibility, 1794
OnBeforeCopy, 1780
OnBeforeCut, 1780
OnBeforeDelete, 1781
OnBeforeDrop, 1781
OnBeforePaste, 1782
OnDragOver, 1783
OnKeyboardEvent, 1784
OnMouseEvent, 1785
OnSelectionChanged, 1786
Parent, 1794
Print, 1794
Redo, 1795
Selection, 1795
Undo, 1796
WholeDocument, 1797
XMLDataRoot, 1797

© 2016-2022 Altova GmbH

Avro,
estructuras de datos en binarios, 773
estructuras de datos en JSON, 772
información general, 767
tipos de archivo, 767
y RaptorXML, 767
Ayuda,
contenido, 1631
mapa de teclado, 1632
Ayudante de entrada,
Detalles, 54
en la vista Cuadrícula, 102
Ayudante de entrada Componentes, 229
Ayudante de entrada Detalles, 54, 232
Ayudante de entrada Facetas, 234
Ayudantes de entrada, 124
para documentos XML, 333
para XQuery, 509
Ayudantes de entrada de la vista Esquema, 228
Ayudantes de entrada en la vista Texto, 164

B

Barra de estado, 134
Barra de herramientas, 134, 1566
activar/desactivar, 1566
agregar comando a, 1564
agregar macro a, 1575
crear nueva, 1566
mostrar iconos grandes, 1579
restablecer barra de herramientas y comandos de menú, 1566
Barra de menú, 134
Barra gris, 1487
Base de datos,
crear BD basada en el esquema, 1467
editar los registros, 1436
exportación de datos XML a, 1474
importar datos como XML, 1455
importar estructura como esquema XML, 1461
Oracle XML DB, 1447
Bases de datos,
compatibles, 1040, 1425
compatibles con XMLSpy, 1040
edición en la vista Authentic, 1412
ver también BD, 639
y recursos globales, 1057
Bases de datos en XMLSpy, 950
Bases de enlaces,
hacer referencia a, 1521
Bases de enlaces en taxonomías, 845
BD, 639, 640
consultas en la vista Authentic, 639
conversión de tipos de datos para XML Schema, 2233
crear consultas, 640
editar en la vista Authentic, 639, 644
filtrar opciones de visualización en la vista Authentic, 640
navegar por tablas en la vista Authentic, 639
parámetros en consultas de BD, 640
tipos de datos de esquemas XML, 2240
BD XML,
administrar esquemas XML para IBM DB2, 1439
asignar esquemas XML para IBM DB2, 1443
cargar filas de datos XML nuevas, 639
cargar nueva fila de datos en la vista Authentic, 1414
Biblioteca, 1253
Big-endian, 1589
BSON en MongoDB, 742
Buscar, 1293
texto en el documento, 1289
y reemplazar texto en el documento, 1293
Buscar cambios, 1584
Buscar en archivos (comando), 1294
Buscar en archivos (ventana), 129
Buscar en esquemas,
ver Búsqueda en esquemas, 475
Buscar en esquemas (ventana), 131
Buscar en XBRL,
término de búsqueda, 930
ver Búsqueda en XBRL, 930
Buscar en XBRL (ventana), 132
Búsqueda de esquemas,
configurar el ámbito de, 483
Búsqueda en esquemas, 475
componentes globales, 487
ejecutar comando Buscar, 933
ejecutar los comandos Buscar y Reemplazar, 484
reemplazar término, 477
renombrar componentes globales, 487
restringir la búsqueda a componentes, 478
restringir la búsqueda a las propiedades y a los valores de las propiedades, 480
resultados, 486
término de búsqueda, 477
ventana, 486

© 2016-2022 Altova GmbH

Búsqueda en XBRL, 930
resultados, 935
ventana, 935

C

C#,
código, 1138
configuración, 1239
integration of XMLSpy, 2060
C++,
código, 1138
configuración, 1239
Cambiar de vista,
a la vista Authentic, 614
Capítulos, 1631
Carácter,
posición, 1488
Catálogo,
Oasis XML, 1337
Catálogos, 456
Catálogos en XMLSpy, 457
Catálogos y edición inteligente, 459
Catalogs and envirnment variables, 460
Centro de soporte técnico, 1636
Certificados, 419
Class,
JAVA, 2001
ClassPath statement, 2001
Clave de evaluación,
para su software de Altova, 1633
Clave privada de certificados, 419
Clave pública de certificados, 419
CodeGeneratorDlg,
Application, 1798
CPPSettings_DOMType, 1799
CPPSettings_LibraryType, 1800
CPPSettings_UseMFC, 1800
CSharpSettings_ProjectType, 1801
OutputPath, 1801
OutputPathDialogAction, 1801
OutputResultDialogAction, 1802
Parent, 1802
ProgrammingLanguage, 1802
PropertySheetDialogAction, 1803
TemplateFileName, 1803
Codificación,
de archivos, 1268
predeterminada, 1589
Código,
SPL, 1241
tipos integrados, 1252
Códigos clave,
para su software de Altova, 1633
Color, 1603, 1606
Color de sintaxis, 1587, 1597
para XQuery, 510
Comando, 1573
agregar a menú/barra de herramientas, 1564
eliminar de menú, 1573
menú contextual, 1573
restablecer menú, 1573
Comandos,
listado del mapa de teclado, 1632
Comandos de menú, 1257
COM-API,
documentation, 1686
Comentarios, 179
Comentarios en la vista Esquema, 182
Comentarios en los scripts SQL, 1432
Comentarios globales,
presentación de la línea de, 179
Command reference, 2072
Comparación XML,
comparar archivos, 1547, 1555
comparar directorios, 1551
Comparaciones,
de archivos, 1094
de archivos y directorios, 1093
de directorios, 1095
Comparaciones de archivos, 1093, 1094
Comparaciones de directorios, 1093, 1095
Comparar archivos, 1547
opciones, 1555
Comparar directorios, 1551
Compatibilidad con Unicode,
de los productos de Altova, 2249
Compatibilidad del código,
v2005R3, 1239
Compilador,
configuración, 1239
Complemento,
archivos de muestra ATL, 1673
configuración de la interfaz del usuario, 1670

© 2016-2022 Altova GmbH

Complemento,
registro, 1669
XMLSPY, 1669
Complemento de XMLSpy para Eclipse,
instalar, 1130
Complemento de XMLSpy para VS .NET,
instalar, 1124
Componentes globales del esquema,
buscar y renombrar, 487
Compositor,
para secuencias, 54
Compositor de secuencia,
usar, 54
Compositores de la vista Esquema, 192
Comprar software de Altova, 1633
Comprobación de corrección XML,
para documento XML, 98
Comprobación de formato de documentos JSON, 674
Comprobación de formato en documentos JSON, 679
Comprobación de formato XML, 1335
Comprobación de sintaxis en documentos JSON, 679
Conectarse a SchemaAgent Server, 464
Conectarse al servidor de SchemaAgent, 1387
Conexión de base de datos,
configurar, 952
ejemplo de configuración, 985
iniciar el asistente, 953
reutilizar desde recursos globales, 984
Configuración, 136, 1583
C++ y C#, 1239
características de los gráficos circulares, 384
características de los gráficos de áreas, 384
características de los gráficos de barras, 384
características de los gráficos de líneas, 384
características de los gráficos de velas, 384
características de los medidores, 384
colores del gráfico, 388
cuadrícula del gráfico, 390, 393, 394
depurador XSLT, 1410
eje X del gráfico, 390
eje Y del gráfico, 393
eje Z del gráfico, 394
fondo del gráfico, 382
fuentes del gráfico, 398
gráficos 3D, 395
leyenda del gráfico, 382
script, 1625
tamaños del gráfico, 396
título del gráfico, 382
Configuración activa,
para recursos globales, 1558
Configuración de Java, 1618
Configuración de la comparación de archivos, 1555
Configuración del programa, 1583
Configuraciones,
de un recurso global, 1043
Configuraciones en recursos globales, 1058
Configurar,
depuración XSL, 1410
interfaz de usuario de XMLSpy, 1670
Configurar impresión, 1277
Configurar la vista,
para la vista del modelo de contenido, 1378
Configurar la vista Cuadrícula (XML), 326
Configurar la vista Texto (cuadro de diálogo), 1489
Configurar vista Cuadrícula (JSON), 709
Conformidad con XML, 1587
Consulta,
ver Consulta de base de datos (ventana), 1423
ver Consultar la base de datos (comando), 1423
ver XQuery, 1423
Consulta de base de datos,
Conectarse a una BD para consultarla, 1425
crear la consulta, 1435
editar los resultados, 1436
panel de mensajes, 1436
Panel Explorador de la ventana de consulta de BD, 1428
resultados, 1436
Consulta de base de datos (ventana), 1423
Consultar la base de datos (comando), 1423
Consultas,
para visualizar BD en la vista Authentic, 640
Contenido, 1631
Contenido JSON externo,
arrastrar y colocar en la vista Cuadrícula JSON, 689
copiar en la vista Cuadrícula JSON, 689
Contenido XML externo,
arrastrar y colocar en la vista Cuadrícula XML, 306
copiar en la vista Cuadrícula XML, 306
Contraer,
elementos no seleccionados, 1487
Contrato de licencia para el usuario final, 2251, 2253
Control de código fuente, 1097, 1625
abrir proyecto, 1303
actualizar estado, 1316
agregar al control de código fuente, 1309

© 2016-2022 Altova GmbH

Control de código fuente, 1097, 1625
anular desprotección, 1309
cambiar de proveedor, 1316
compartir desde, 1310
desproteger, 1306
habilitar, deshabilitar, 1304
mostrar diferencias, 1313
mostrar historial, 1312
obtener archivos, 1304
obtener la versión más reciente, 1304
propiedades, 1315
proveedores compatibles, 1302
quitar del, 1310
Controladores de BD,
resumen, 955
Controladores ODBC,
comprobar disponibilidad, 971
Controles ActiveX,
compatibilidad, 1670
conversión de tipos de datos en esquemas XML generados a partir de BD, 2236
Convertir,
archivo de texto en XML, 1452
datos de BD en XML, 1455
datos de MS Word en XML, 1460
esquema de BD en esquema XML, 1461
esquema en estructura de BD, 1467
Oracle XML DB, 1447
Convertir en OIM xBRL-CSV, 1483
Convertir en OIM xBRL-JSON, 1482
Convertir en OIM xBRL-XML, 1482
Convertir texto XML en comentario o eliminar comentario, 1297
Copia de seguridad automática, 143
Copiar,
como texto XML, 1281
XML como texto estructurado, 1282
Copiar (comando), 1280
Copiar XPath, 1283
Copiar XPointer, 1283
Copias de seguridad, 143
Copy image in Grid View, 1282
CoreCatalog.xml, 457
Corrección rápida de esquemas XML, 238
Correo electrónico,
enviar archivos por, 1274
Cortar (comando), 1280
CPU,
carga - acelerar actualización de estado en segundo plano, 1097
CR & LF, 1584
Creación de documento XML,
tutorial, 88
Crear,
BD basada en el esquema, 1467
CSS, 656
color de sintaxis, 659
esquema del documento, 659
finalización automática, 659
propiedades, 659
ventana de información, 659
CustomCatalog, 1337
CustomCatalog.xml, 457
CVS, 1097

D

DatabaseConnection,
ADOConnection, 1804
AsAttributes, 1804
CreateMissingTables, 1805
CreateNew, 1805
DatabaseKind, 1805
ExcludeKeys, 1806
File, 1806
IncludeEmptyElements, 1807
NumberDateTimeFormat, 1807
ODBCConnection, 1808
SQLSelect, 1809
TextFieldLen, 1809
Datos JSON,
ejemplo, 668
matrices, 668
objetos, 668
tipos, 668
Datos XML,
exportar a archivo de texto, 1471
exportar a base de datos, 1474
db2-fn:sqlquery, 526
db2-fn:xmlcolumn, 526
defaultOpenContent, 182
Definición de componente, 67
ampliar la definición, 67
en definiciones de esquema, 67

© 2016-2022 Altova GmbH

Definición de componente, 67
reutilizar, 67
Definiciones de esquema XML,
avanzadas, 67
Definir,
características de los gráficos circulares, 384
características de los gráficos de áreas, 384
características de los gráficos de barras, 384
características de los gráficos de líneas, 384
características de los gráficos de velas, 384
características de los medidores, 384
color de los gráficos, 388
colores del gráfico, 388
fuentes del gráfico, 398
fuentes en los gráficos, 398
líneas de la cuadrícula, 390, 393, 394
opciones de configuración 3D, 395
opciones de configuración del eje X, 390
opciones de configuración del eje Y, 393
opciones de configuración del eje Z, 394
tamaños del gráfico, 396
título del gráfico, 382
Degradado,
fondo, 1239
Delete,
Application.URLDelete, 1742
Depuración XSLT/XQuery,
archivos usados, 531
Depurador,
configuración, 1410
cuadro de diálogo puntos de interrupción/seguimiento, 1408
detener el depurador XSLT, 1405
ejemplo de comunicación SOAP, 790
finalizar la sesión, 1406
habilitar o deshabilitar punto de interrupción, 1407
habilitar o deshabilitar punto de seguimiento, 1408
iniciar el depurador XSLT, 1405
insertar o quitar punto de interrupción, 1407
insertar o quitar punto de seguimiento, 1407
mostrar nodos en ejecución, 1407
opciones para SOAP, 1513
para SOAP, 1511
paso a paso para salir, 1406
paso a paso por instrucciones, 1406
paso a paso por procedimientos, 1406
reiniciar el depurador XSLT, 1405
ventanas del depurador, 1409
Depurador SOAP,
ejemplo de uso, 790
en Visual Studio .NET, 1128
establecer puntos de interrupción, 802
Depurador XQuery,
ver depurador XSLT/XQuery, 531
Depurador XSLT,
configuración, 1410
cuadro de diálogo puntos de interrupción/seguimiento, 1408
detener el depurador, 1405
en Visual Studio .NET, 1128
finalizar la sesión del depurador, 1406
habilitar o deshabilitar punto de interrupción, 1407
habilitar o deshabilitar punto de seguimiento, 1408
iniciar el depurador, 1405
insertar o quitar punto de interrupción, 1407
insertar o quitar punto de seguimiento, 1407
mostrar nodos en ejecución, 1407
paso a paso para salir, 1406
paso a paso por instrucciones, 1406
paso a paso por procedimientos, 1406
reiniciar el depurador, 1405
ventanas del depurador, 1409
Depurador XSLT/XQuery, 532
configuración, 537
descripción de la interfaz, 532
descripción de su funcionamiento, 532
funciones y uso, 531
iconos de la barra de herramientas, 534
iniciar una sesión, 539
uso de los puntos de interrupción, 548
uso de los puntos de seguimiento, 551
ventana Contexto, 542
ventana Información, 546
ventana Inspección XPath, 543
ventana Mensajes, 544
Ventana Pila de llamadas, 544
ventana Plantillas, 545
ventana Seguimiento, 546
ventana Variables, 542
ventanas de información, 541
Depurador XSTL/XQuery,
teclas de acceso rápido, 557
Depurar expresiones XPath/XQuery, 582
Derivaciones de tipo simple, 232
Derivar un tipo del esquema, 244
Desconectar XMLSpy de SchemaAgent, 1388
Descripción de la IGU, 119
Descripción de la interfaz gráfica, 119

© 2016-2022 Altova GmbH

Deshabilitar,
punto de interrupción - depurador XSLT, 1407
punto de seguimiento (depurador XSLT), 1408
Deshabilitar validación automática, 1587
Deshacer (comando), 1280
Detener,
depurador XSLT, 1405
Dialogs,
Application, 1810
CodeGeneratorDlg, 1810
DTDSchemaGeneratorDlg, 1812
FileSelectionDlg, 1810
GenerateSampleXMLDlg, 1811
Parent, 1811
SchemaDocumentationDlg, 1811
Diccionario, 1539
añadir diccionario personalizado, 1539
modificar, 1539
revisión ortográfica, 1539
Diccionario personalizado, 1539
directories,
creating with Application.URLMakeDirectory, 1742
Directorios,
comparar dos, 1551
Distribución,
de productos de software de Altova, 2251
Dll,
configuración del compilador, 1239
DocEditEvent (obsolete),
altKey (obsolete), 1951
altLeft (obsolete), 1952
button (obsolete), 1953
cancelBubble (obsolete), 1954
clientX (obsolete), 1955
clientY (obsolete), 1955
ctrlKey (obsolete), 1956
ctrlLeft (obsolete), 1957
dataTransfer (obsolete), 1957
fromElement (obsolete), 1958
keyCode (obsolete), 1959
propertyName (obsolete), 1959
repeat (obsolete), 1960
returnValue (obsolete), 1960
shiftKey (obsolete), 1960
shiftLeft (obsolete), 1961
srcElement (obsolete), 1962
type (obsolete), 1962
DocEditView (obsolete),
ApplyTextState (obsolete), 1966
CurrentSelection (obsolete), 1967
EditClear (obsolete), 1968
EditCopy (obsolete), 1968
EditCut (obsolete), 1969
EditPaste (obsolete), 1969
EditRedo (obsolete), 1970
EditSelectAll (obsolete), 1970
EditUndo (obsolete), 1970
event (obsolete), 1971
GetAllowedElements (obsolete), 1971
GetNextVisible (obsolete), 1973
GetPreviousVisible (obsolete), 1974
IsEditClearEnabled (obsolete), 1974
IsEditCopyEnabled (obsolete), 1975
IsEditCutEnabled (obsolete), 1975
IsEditPasteEnabled (obsolete), 1976
IsEditRedoEnabled (obsolete), 1976
IsEditUndoEnabled (obsolete), 1977
IsRowAppendEnabled (obsolete), 1977
IsRowDeleteEnabled (obsolete), 1978
IsRowDuplicateEnabled (obsolete), 1978
IsRowInsertEnabled (obsolete), 1979
IsRowMoveDOwnEnabled (obsolete), 1979
IsRowMoveUpEnabled (obsolete), 1980
IsTextStateApplied (obsolete), 1980
IsTextStateEnabled (obsolete), 1981
LoadXML (obsolete), 1981
RowAppend (obsolete), 1982
RowDelete (obsolete), 1983
RowDuplicate (obsolete), 1983
RowInsert (obsolete), 1984
RowMoveDown (obsolete), 1984
RowMoveUp (obsolete), 1985
SaveXML (obsolete), 1985
SelectionMoveTabOrder (obsolete), 1986
SelectionSet (obsolete), 1986
XMLRoot (obsolete), 1987
Document, 1825
Application, 1818
AssignDTD, 1818
AssignSchema, 1818
AssignXSL, 1818
AssignXSLFO, 1819
AuthenticView, 1819
Close, 1820
ConvertDTDOrSchema, 1820
CreateChild, 1822

© 2016-2022 Altova GmbH

Document, 1825
CreateSchemaDiagram, 1823
CurrentViewMode, 1823
DataRoot, 1823
DocEditView, 1824
Encoding, 1824
EndChanges, 1824
ExecuteXQuery, 1825
ExportToDatabase, 1825
ExportToText, 1826
FullName, 1828
GenerateDTDOrSchema, 1828
GenerateProgramCode, 1829
GenerateSampleXML, 1829
GenerateSchemaDocumentation, 1829
GetExportElementList, 1831
GetPathName, 1832
GridView, 1832
IsModified, 1833
IsValid, 1833
IsWellFormed, 1835
Name, 1835
OnBeforeCloseDocument, 1816
OnBeforeSaveDocument, 1815
OnBeforeValidate, 1816
OnCloseDocument, 1817
OnViewActivation, 1817
Path, 1836
RootElement, 1836
Save, 1837
SaveAs, 1837
Saved, 1837
SaveInString, 1837
SaveToURL, 1838
SetActiveDocument, 1838
SetEncoding, 1838
SetExternalIsValid, 1839
SetPathName, 1839
StartChanges, 1840
SwitchViewMode, 1840
Title, 1841
TransformXSL, 1841
TransformXSLFO, 1842
UpdateViews, 1842
UpdateXMLData, 1842
XQuery, 1825
Documentación,
de archivos WSDL, 1499
de esquemas XML y esquemas JSON, 1373
de taxonomías XBRL, 1523
para esquemas, 83
Document-level,
examples of integration of XMLSpy, 2060
Documento,
examinar Oracle XML DB, 1450
revisión ortográfica, 1539
Documento de salida HTML,
generar en la vista Authentic a partir del archivo PXF, 1421
Documento de salida PDF,
generar en la vista Authentic a partir del archivo PXF, 1421
Documento de salida RTF,
generar en la vista Authentic a partir del archivo PXF, 1421
Documento de salida Word 2007+,
generar en la vista Authentic a partir del archivo PXF, 1421
Documento Word,
importar como XML, 1460
Documento XML,
abrir en la vista Authentic, 600
asignar a la hoja de estilos XSLT, 1404
crear nuevo, 88
edición en la vista Texto, 93
examinar Oracle XML DB, 1450
Documento XML nuevo,
crear, 88
documentos DTD,
convertir en esquemas XML (edición Enterprise y Professional), 442
edición en la vista Cuadrícula (ediciones Enterprise y Professional), 442
edición en la vista Texto, 442
generar documento XML a partir de, 442
Documentos en la ventana principal, 120
Documentos HTML,
edición, 657
ventana de información, 657
Documentos JSON,
abrir en XMLSpy, 665
conversión en XML y viceversa, 674, 766
crear nuevo, 665
edición en la vista Texto, 674
editar en la vista Cuadrícula, 679
expresiones XQuery para, 593
validación, 754
Documentos XML, 285
abrir, 286
asignar esquemas (incluidos archivos DTD), 288

© 2016-2022 Altova GmbH

Documentos XML, 285
automatizar ejecuciones XQuery de, 341
automatizar transformaciones XSLT de, 341
buscar y reemplazar en, 423
codificación de, 423
comprobar formato XML, 288
comprobar validez, 98
edición en la vista Texto, 331
editar en la vista Cuadrícula, 293
evaluar expresiones XPath en, 423
funciones de edición de la vista Texto para, 290
generar esquemas a partir de, 423
guardar, 286
importar y exportar texto, 423
insertar rutas de acceso de archivos en, 290
insertar XInclude, 290
transformación con XSLT, 341
validación, 288
validación automática, 286
vistas predeterminadas de, 286
y comentarios en la vista Texto, 290
y expresión XPath de un nodo, 290
y líneas vacías en la vista Texto, 290
y XQuery, 341
Documentos XQuery,
analizar tiempo de ejecución de, 558, 1403
Documentos XSLT,
edición y administración con ayuda del esquema XSL, 496
y la ventana de información, 500
Documentos XULE,
compatibilidad para editar, 922
finalización automática, 922
validación de, 922
y proyectos de XMLSpy, 922
y taxonomías para finalización automática, 922
y tipos de archivos conforme a esquemas, 922
Documents,
Count, 1844
Item, 1844
NewAuthenticFile, 1844
NewFile, 1845
NewFileFromText, 1845
OpenAuthenticFile, 1845
OpenFile, 1846
OpenURL, 1846
OpenURLDialog, 1847
DSN de archivo,
configurar, 971
DSN de sistema,
configurar, 971
DSN de usuario,
configurar, 971
DTD, 425, 1584, 1587
asignar a un documento XML, 1355
comandos de menú relacionados con, 1355
convertir en esquema XML, 1360
convertir en UML, 1363
generar a partir de esquema XML (ediciones Enterprise y Professional), 444
generar archivo XML de muestra a partir de, 1365
generar código a partir de, 1369
generar DTD a partir del documento XML, 1358
incluir entidades, 1356
ir a la definición en la DTD desde el documento XML, 1357
ir a la DTD desde el documento XML, 1357
DTDSchemaGeneratorDlg,
Application, 1848
AttributeTypeDefinition, 1848
DTDSchemaFormat, 1848
FrequentElements, 1848
GlobalAttributes, 1849
MaxEnumLength, 1849
MergeAllEqualNamed, 1849
OnlyStringEnums, 1850
OutputPath, 1850
OutputPathDialogAction, 1850
Parent, 1850
ResolveEntities, 1851
TypeDetection, 1851
ValueList, 1851
Duplicar,
fila (en la vista Authentic), 1420

E

Edición de esquemas,
modelos de contenido, 190
Edición de esquemas BSON, 742
Edición en la vista Texto, 151
Edición inteligente, 1590
Edit as Raw Text,
in Grid View, 1334
Editado con XMLSPY, 1584
Editar,

© 2016-2022 Altova GmbH

Editar,
botón de macro, 1580
Editar registros de base de datos, 1436
Editor de definiciones de tabla XBRL, 889
Editor de fórmulas XBRL, 864
Editor de script,
iniciar, 1546
Editor de scripting,
vista general, 1643
Editor predeterminado, 1587
Editor SQL,
crear una consulta en, 1435
descripción, 1432
en la ventana de consulta de base de datos, 1432
Ejecutar XQuery, 516
Ejecutar XQuery/XQuery Update, 1400
Ejecutar XULE, 1534
cómo, 929
opciones, 929
ElementFormDefault,
configuración del esquema, 1370
ElementList,
Count, 1852
Item, 1852
RemoveElement, 1852
ElementListItem,
ElementKind, 1853
FieldCount, 1853
Name, 1853
RecordCount, 1853
Elemento,
convertir en opcional, 62
Element, 62
limitar el contenido, 62
Elemento global,
uso en esquema XML, 75
Elemento opcional,
convertir en, 62
Elementos (ayudante de entrada),
en la vista Authentic, 619
Elementos globales, 182
Elementos no seleccionados, 1487
Elementos Notation en la vista Esquema, 182
Elementos repetidos, 1590
Elementos vacíos, 1587
Eliminar, 1564
acceso rápido, 1569
barra de herramientas, 1566
comando de la barra de herramientas, 1564
comando del menú contextual, 1573
fila (en la vista Authentic), 1420
icono de la barra de herramientas, 1564
Eliminar (comando), 1280
Eliminar espacios en blanco, 1289
En paralelo, 1597
Entidades,
definir en la vista Authentic, 627, 648
en documentos XML basados en esquemas XML, 288
insertar en la vista Authentic, 610, 627
Entidades (ayudante de entrada),
en la vista Authentic, 619
Entidades externas analizadas, 1587
Entorno de scripting, 1641
Entry Helpers,
updating, 1344
Enumeración,
definir para atributos, 77
Enumerations,
in XMLSpyControl, 2111
SPYAttributeTypeDefinition, 1988
SPYAuthenticActions, 1988
SPYAuthenticDocumentPosition, 1988
SpyAuthenticElementActions, 1989
SPYAuthenticElementKind, 1989
SPYAuthenticMarkupVisibility, 1989
SPYDatabaseKind, 1990
SPYDialogAction, 1990
SPYDOMType, 1990
SPYDTDSchemaFormat, 1991
SPYEncodingByteOrder, 1991
SPYExportNamespace, 1991
SPYFrequentElements, 1992
SPYKeyEvent, 1992
SPYLibType, 1993
SPYLoading, 1993
SPYMouseEvent, 1993
SPYNumberDateTimeFormat, 1994
SPYProgrammingLanguage, 1994
SPYProjectItemTypes, 1994
SPYProjectType, 1995
SPYSampleXMLGenerationOptimization, 1995
SPYSampleXMLGenerationSchemaOrDTDAssignment, 1996
SPYSchemaDefKind, 1996
SPYSchemaDocumentationFormat, 1997
SPYTextDelimiters, 1998

© 2016-2022 Altova GmbH

Enumerations,
SPYTextEnclosing, 1998
SPYTypeDetection, 1998
SPYURLTapes, 1998
SPYViewModes, 1999
SPYVirtualKeyMask, 2000
SPYXMLDataKind, 2000
Environnment variables, 460
Equipo vrtual Java,
configurar ruta de acceso, 1618
Espacio de nombres,
en esquemas, 53
Espacio en blanco,
eliminar, 1289
Espacios de nombres,
configuración del esquema, 1370
en documentos WSDL, 776
en taxonomías XBRL, 843
Espacios de nombres de destino en XBRL, 1519
Espacios de nombres en XBRL, 1519
Espacios en blanco en XBRL inline, 1620
Esquema,
abrir esquema WSDL, 783
administrar para IBM DB2, 1439
configuración, 1584
convertir en UML, 1363
crear BD basada en el esquema, 1467
documentación, 83
fuentes de la documentación, 1608
generador de código, 1138
gestión y asignación en bases de datos IBM DB2, 1439
ver esquema XML, 51
ver también Esquema XML, 1355
Vista Diseño, 1485
Esquema Avro, 665
Esquema JSON, 665, 673
agregar definiciones globales, 717
condicionales, 748
cualquiera, 740, 746
definiciones de cadena, 738
definiciones numéricas, 738
descripción, 671
generar a partir de instancia JSON, 762
generar documentación de, 1373
if-then-else, 748
libre, 740
matrices, 736
múltiple, 740
no, 746
objetos, 725, 729
objetos y dependencias, 732
operadores, 746
prohibido, 740
propiedades de los objetos, 725
propiedades sin especificar, 729
selectores de tipo (cualquiera, múltiple), 740
selectores de tipo (libre, prohibido), 740
terminología, 671
tipos atómicos, 738
tipos primitivos, 738
tipos simples, 738
todos, 746
uno, 746
Esquema XML, 51, 1355, 1370
<alternative>, 205
administrar para IBM DB2, 1439
Archivos DTD, 442
asignar al documento XML, 1356
comandos de menú relacionados con, 1355
configuración de espacio de nombres en la vista Diseño de esquemas, 1370
configuración del esquema, 1370
configuración del esquema de MS SQL Server, 1385, 1386
configuración del esquema de Oracle, 1384
configurar la vista del modelo de contenido, 1378
convertir en DTD, 1360
corrección rápida, 238
crear un esquema básico, 51
definir espacios de nombres en, 53
diagrama del modelo de contenido, 1373
editar modelos de contenido, 190
extensiones de MS SQL Server, 1384
extensiones de Oracle, 1383
generar a partir de esquema XML (ediciones Enterprise y Professional), 444
generar archivo XML de muestra a partir de, 1365
generar código a partir de, 1369
generar documentación de, 1373
generar esquema a partir del documento XML, 1358
gestión y asignación en bases de datos IBM DB2, 1439
ir a la definición en el esquema desde el documento XML, 1357
ir al esquema desde el documento XML, 1357
modificar esquema XML mientras se edita el documento XML, 108
navegación en la vista del diseñador, 81
tutorial, 51

© 2016-2022 Altova GmbH

Esquema XML, 51, 1355, 1370
validación, 238
ver también Esquema, 1355
Esquema XSL, 496
Esquemas,
asignar a una BD XML, 1443
buscar en los catálogos, 459
en memoria, 1369
Esquemas Avro,
descripción, 769
terminología, 769
Esquemas XML, 425
asignar a una BD XML, 1443
convertir en DTD (ediciones Enterprise y Professional), 444
dividirlos en subconjuntos, 1389
edición en la vista Cuadrícula (ediciones Enterprise y Professional), 444
edición en la vista Esquema (ediciones Enterprise y Professional), 444
edición en la vista Texto, 444
generar documento XML a partir de, 444
incluir otros esquemas en, 445
nivelar esquemas incluidos, 445
nivelar los esquemas incluidos, 1390
y archivos DTD, 288
y recursos globales, 288
Esquemas y catálogos, 456
Esquemas, buscar en,
ver Búsqueda en esquemas, 475
Estilo, 1603, 1606
Estructura del documento XSLT, 128
Evaluador de XPath Evaluator,
uso, 1335
Evaluar XPath, 127
Event, 1725, 1726, 1727, 1780, 1781, 1782, 1783, 1784, 1785, 1786, 1815, 1816, 1817, 1875, 1876, 1877
Events, 1693
Examinar,
Oracle XML DB, 1450
Excel 2007, 281, 938
Excepción,
memoria insuficiente, 1238
Expandir,
todo, 1487
Explorador, 1587, 1597
Vista, 1486
Explorador (menú), 1491
Exportar datos XML a archivos de texto, 1471
Exportar datos XML a una base de datos, 1474
ExportSettings,
CreateKeys, 1854
ElementList, 1854
EntitiesToText, 1854
ExportAllElements, 1854
FromAttributes, 1855
FromSingleSubElements, 1855
FromTextValues, 1855
IndependentPrimaryKey, 1855
Namespace, 1855
SubLevelLimit, 1856
Expresiones regulares, 1294
en la cadena de búsqueda, 1289
Expresiones XPath/XQuery,
construir en la ventana XPath/XQuery, 590
depurar, 582
en la ventana XPath/XQuery, 595
evaluar, 576
evaluar y depurar, 573
para documentos JSON, 593
Expresiones XQuery en la vista Cuadrícula JSON, 696
Expresiones XQuery en la vista Cuadrícula XML, 314, 316
Expresiones XQuery para JSON, 760
Extensiones de Altova,
funciones para gráficos, 2121
Extensiones de archivo,
para archivos XQuery, 507
personalizar, 1337

F

Facetas assertion de tipos simples, 234
Facetas de tipos simples, 234
Facetas enumeration de tipos simples, 234
Facetas pattern de tipos simples, 234
Favoritos, 1631
Fechas,
cambiar manualmente, 647
Fila,
anexar (en la vista Authentic), 1420
duplicar (en la vista Authentic), 1420
eliminar (en la vista Authentic), 1420
insertar (en la vista Authentic), 1420
subir/bajar, 1421

© 2016-2022 Altova GmbH

FileSelectionDlg,
Application, 1856
DialogAction, 1857
FullName, 1857
Parent, 1857
Filtros de la vista Cuadrícula JSON,
configurar, 709
Filtros en la vista Cuadrícula JSON, 696
Filtros en la vista Cuadrícula XML, 314
configurar, 326
Finalización automática,
habilitar/deshabilitar en la vista Texto, 1590
Finalización automática en los scripts SQL, 1432
Finalizar,
sesión del depurador, 1406
Firebird,
conectarse por JDBC, 985
conectarse por ODBC, 986
Firma,
ver Firmas XML, 1344
Firma XML, 1415
Firmas,
ver Firmas XML, 411
Firmas XML, 411, 650
crear, 413, 1344
verificar, 416, 1347
Fondo,
degradado, 1239
FOP,
fuentes, 343
Formato de salida, 1584
Formato en la vista Texto, 146
Formato XML correcto en documentos XML, 288
Formatos de imagen,
en la vista Authentic, 651
Formatos gráficos,
en la vista Authentic, 651
Formatos xBRL OIM, 936
Fórmulas,
generar a partir de la vista previa del diseño de tabla, 918
Fórmulas en la vista Cuadrícula JSON, 696, 697
configurar, 709
Fórmulas en la vista Cuadrícula XML, 316
configurar, 326
Fuentes, 1608
documentación del esquema, 1608
vista Esquema, 1608
Fuentes de la vista Cuadrícula, 1606
Fuentes de la vista Esquema, 1608
Fuentes de la vista WSDL, 1611
Fuentes de la vista XBRL, 1612
Fuentes en la vista Texto, 146
Fuentes PDF, 343
Función de zoom,
en la vista Diseño de esquemas, 1381
Funciones de arco en XBRL, 1515
Funciones de extensión .NET,
campos de instancia, 2227
campos estáticos, 2226
constructores, 2226
conversiones de tipos de datos, 2228, 2229
métodos de instancia, 2227
métodos estáticos, 2226
para XSLT y XQuery, 2223
resumen, 2223
tipos de datos .NET en XPath/XQuery, 2229
tipos de datos XPath/XQuery en .NET, 2228
Funciones de extensión .NET para XSLT y XQuery,
ver Funciones de extensión .NET, 2223
Funciones de extensión en scripts MSXSL,
msxsl:script, 2229
Funciones de extensión Java,
archivos de clases definidos por el usuario, 2215
archivos JAR definidos por el usuario, 2219
campos de instancia, 2221
campos estáticos, 2220
constructores, 2220
conversiones de tipos de datos, 2222, 2223
métodos de instancia, 2221
métodos estáticos, 2220
para XSLT y XQuery, 2214
resumen, 2214
tipos de datos Java en XPath/XQuery, 2223
tipos de datos XPath/XQuery en Java, 2222
Funciones de extensión Java para XSLT y XQuery,
ver Funciones de extensión Java, 2214
Funciones de extensión para XSLT y XQuery, 2213
Funciones de vínculo en taxonomías XBRL, 851
Funciones de vínculo en XBRL, 1517
Funciones para gráficos,
ejemplo, 2210
estructura de datos de gráficos, 2205
fragmento de código, 2200
Funciones XSLT,
en la ventana Esquema XSL, 497

© 2016-2022 Altova GmbH

G

Generación de perfiles, 558
Gráfico de llamadas, 558
Recuento, 558
Generación de perfiles con gráfico de llamadas, 558
Generación de perfiles con recuento, 558
Generador de código, 1138, 1369
Generador de perfiles, 558, 1403
Generador de perfiles XQuery,
gráficos de resultados, 570
Generador de perfiles XSLT,
gráficos de resultados, 570
Generador de perfiles XSLT/XQuery, 558
Generar,
código a partir de un esquema, 1138
estructura de BD basada en el esquema, 1467
Generate Sample XML, 1988, 1995, 1996
GenerateSampleXMLDlg,
Application, 1870
FillWithSampleData, 1871
NonMandatoryAttributes, 1872
NonMandatoryElements, 1872
Parent, 1873
RepeatCount, 1873
TakeFirstChoice, 1874
Gestión de proyectos en XMLSpy, 114
Gestor de esquemas,
aplicar un parche a un esquema, 433
desinstalar un esquema, 434
estado de los esquemas en el, 431
funcionamiento del, 429
help (comando ILC), 435
info (comando ILC), 436
initialize (comando ILC), 436
instalar un esquema, 433
install (comando ILC), 437
introducción, 426
introducción a la ILC, 435
list (comando ILC), 438
lista de esquemas por estado, 431
reset (comando ILC), 438
restaurar, 434
uninstall (comando ILC), 439
update (comando ILC), 440
upgrade (comando ILC), 441
Gestor de taxonomías,
actualizar una taxonomía, 825
aplicar un parche a una taxonomía, 825
desinstalar una taxonomía, 826
estado de las taxonomías en el, 824
funcionamiento, 822
help (comando ILC), 828
info (comando ILC), 828
initialize (comando ILC), 829
instalar una taxonomía, 825
install (comando ILC), 829
introducción, 818
introducción a la ILC, 827
list (comando ILC), 830
lista de taxonomías por estado, 824
reset (comando ILC), 831
restaurar, 826
uninstall (comando ILC), 831
update (comando ILC), 832
upgrade (comando ILC), 833
Global,
configuración, 1583
Gráficos,
agregar leyenda, 382
aspecto, 372
características de los gráficos circulares, 384
características de los gráficos de áreas, 384
características de los gráficos de barras, 384
características de los gráficos de líneas, 384
características de los gráficos de velas, 384
características de los medidores, 384
color de fondo, 382
color de la serie, 388
combinación de colores, 388
configuración del gráfico, 369
datos del gráfico, 366
definir colores, 388
eje X, 390
eje Y, 393
eje Z, 394
ejemplo (gráfico avanzado), 401
ejemplo (gráfico básico), 399
ejemplo (gráfico de velas), 407
en el generador de perfiles XSLT/XQuery, 570
exportación, 399
fuentes, 398
intervalo de colores, 388

© 2016-2022 Altova GmbH

Gráficos,
líneas de la cuadrícula, 390, 393, 394
márgenes, 396
multicapa, 368
opciones de configuración 3D, 395
quitar leyenda, 382
resumen, 346
seleccionar eje X, 357
seleccionar eje Y, 362
tamaño de las marcas de graduación, 396
tamaños, 396
título, 382
varias pestañas para, 346
volver a cargar, 346
XPath de origen, 353
Gráficos circulares, 362
Gráficos circulares (características), 384
Gráficos de áreas (características), 384
Gráficos de barras (características), 384
Gráficos de líneas (características), 384
Gráficos de velas (características), 384
Gráficos en la vista Cuadrícula, 320
Gráficos en la vista Cuadrícula JSON, 703
Gramática, 1587
GridView,
CurrentFocus, 1877
Deselect, 1877
IsVisible, 1878
OnBeforeDrag, 1875
OnBeforeDrop, 1875
OnBeforeStartEditing, 1876
OnEditingFinished, 1876
OnFocusChanged, 1877
Select, 1878
SetFocus, 1878
Grupo de inicio,
agregar (menú contextual), 1580
Grupos de atributo en la vista Esquema, 213
Grupos de atributos globales, 182
Guardar archivos,
codificación de, 1268
Guía y referencia del usuario, 29
Guías de sangría, 1489

H

Habilitar punto de interrupción - depurador XSLT, 1407
Habilitar punto de seguimiento (depurador XSLT), 1408
Herramientas,
ver también Aplicaciones externas, 1568
Hipervínculos,
navegar por la vista Authentic, 627
Historial de navegación, 250
Hoja de estilos StyleVision Power Stylesheet,
asignar al archivo XML, 1413
editar en StyleVision, 1413
Hoja de estilos XSLT,
abrir, 1404
asignar documento XML a la, 1404
Hojas de estilos XSLT,
analizar tiempo de ejecución de, 558, 1403
asignar documentos XML, 1403
Hojas de estilos XSLT para FO,
asignar al documento XML, 1403
HTML, 656
HTTP, 805

I

IBM DB2,
asignar esquemas XML al archivo XML, 1443
conectarse por JDBC, 989
conectarse por ODBC, 991
gestión de esquemas XML, 1439
gestión y asignación de esquemas, 1439
IBM DB2 for i,
conectarse por JDBC, 997
conectarse por ODBC, 998
IBM Informix,
conectarse por JDBC, 1001
Icono,
agregar a menú/barra de herramientas, 1564
mostrar en tamaño grande, 1579
Imágenes en la vista Cuadrícula, 319
Imágenes en la vista Cuadrícula JSON, 702
Importación XML,
basada en esquema, 1466

© 2016-2022 Altova GmbH

Importar,
archivo de texto como XML, 1452
datos de BD como XML, 1455
datos de BD en base al esquema XML, 1466
documento de MS Word como XML, 1460
estructura de BD como esquema XML, 1461
Importar taxonomías, 1521
Imprimir,
desde la vista Authentic, 611
Información general, 2248
Información legal, 2251
Información rápida,
mostrar, 1579
mostrar teclas de acceso rápido en, 1579
Información sobre derechos de autor, 2251
Información técnica, 2248
Iniciar,
depurador XSLT, 1405
Inline XBRL en XMLSpy, 863
Insertar,
fila (en la vista Authentic), 1420
punto de interrupción - depurador XSLT, 1407
punto de seguimiento - depurador XSLT, 1407
Insertar o quitar comentarios,
en documentos XML en la vista Texto, 290
Instrucción de procesamiento stylesheet, 1403, 1404
Instrucciones de procesamiento en la vista Esquema, 182
Integrating,
XMLSpy in applications, 2052
Interfaz del usuario,
configurar usando complementos, 1670
Internet, 1636
Ir a la línea o carácter, 1488
Ir al archivo, 1489

J

Java, 2063
API, 2001
ClassPath, 2001
código, 1138
configuración, 1239
JDBC,
como interfaz de conexión de datos, 952
conectarse a Teradata, 1033
configurar una conexión (Windows), 974
JRE,
para el complemento de XMLSpy para Eclipse, 1130
JSON,
convertir esquema JSON en esquema XML y viceversa, 1480
convertir instancia JSON en instancia XML y viceversa, 1476
expresiones XQuery para, 760
generar archivo a partir de esquema JSON, 1365
y la ventana Resultados XPath/XQuery, 758
y XPath, 758
y XQuery, 758
JSON Grid View,
envolver componente en matriz, 1333, 1352
envolver componente en objeto, 1333, 1352
flip rows and columns of table, 1334, 1353
sort column in table, 1334, 1353
Table Display command, 1334, 1353
Juego de caracteres,
codificación, 1589

L

LAP en XQuery Update, 519
Lenguaje de scripting, 1625
Lib,
configuración del compilador, 1239
Licencia, 2253
información sobre, 2251
para su software de Altova, 1633
Licencia del producto de software, 2253
Línea,
ir a, 1488
Líneas JSON, 673
Líneas vacías,
en documentos XML en la vista Texto, 290
Lista de actualizaciones pendientes (LAP), 519
Little-endian, 1589
Llamar a plantillas con nombre, 497
loading, 1846
Localizador de objetos,
de la ventana de consulta de BD, 1428
Longitud de línea,
ajuste automático de línea en la vista Texto, 1488

© 2016-2022 Altova GmbH

M

Macro,
agregar a menú/barra de herramientas, 1575
botón, 1580
Macros,
desarrollar, 1647
ejecutar, 1667
ejecutar macros de aplicación, 1547
habilitar, 1654, 1666
MainCatalog, 1337
Mapa de teclado, 1632
MapForce, 1365
y XBRL, 1528
Marcado,
en la vista Authentic, 614, 617
Marcado (en la vista Authentic),
mostrar en tamaño pequeño/grande/mixto, 1419
ocultar, 1419
Marcado en tamaño grande (en la vista Authentic), 1419
Marcado en tamaño pequeño (en la vista Authentic), 1419
Marcado mixto (en la vista Authentic), 1419
Marcador, 1631
Marcadores,
insertar y quitar, 1296
navegar, 1296
Marcadores de espacios en blanco, 1489
Marcadores de final de línea, 1489
Marcadores en la vista Texto, 149
Marcadores en los scripts SQL, 1432
Margen de marcadores, 1489
Margen de números de línea, 1489
Margen de plegamiento, 1489
MariaDB,
conectarse por ODBC, 1003
Medición de licencias,
en los productos de Altova, 2252
Medidores (características), 384
Memoria,
almacenamiento de la información del esquema, 1369
insuficiente - excepciones, 1238
Memoria insuficiente, 1238
Mensajes de aserción, 240
Mensajes de validación, 125
Menú, 1573
agregar macro a, 1575
agregar/eliminar comando, 1564
Authentic, 1411
Ayuda, 1631
Conversión, 1452
Diseño de esquemas, 1370
DTD/Esquema, 1355
Edición, 1279
eliminar comandos de, 1573
Herramientas, 1538
JSON, 1351
personalizar, 1573
predeterminado/XMLSPY, 1573
Proyecto, 1298
SOAP, 1504
Ventanas, 1629
Vista, 1484
WSDL, 1492
XBRL, 1515
XML, 1332
XSL/XQuery, 1392
Menú Archivo, 1258
Menú Authentic, 1411
edición de tablas dinámicas, 614
presentación del marcado, 614
Menú Ayuda, 1631
Menú contextual,
comandos, 1573
para personalización, 1580
Menú Conversión, 1452
Menú Diseño de esquemas, 1370
Menú DTD/Esquema, 1355
Menú Edición, 1279
Menú Explorador, 1491
Menú Herramientas, 1538
Menú JSON, 1351
Menú Proyecto, 1298
Menú SOAP, 1504
Menú Ventanas, 1629
Menú Vista, 1484
Menú WSDL, 1492
Menú XBRL, 1515
Menú XML, 1332
Menú XSL/XQuery, 1392
Menús contextuales,
en la vista Authentic, 624
Método,

© 2016-2022 Altova GmbH

Método,
reservar nombre, 1238
Microsoft Access,
conectarse por ADO, 958, 1005
Microsoft Office 2007, 281, 938
Microsoft SQL Server,
conectarse por ADO, 1007
conectarse por ODBC, 1010
Microsoft® SharePoint® Server, 1322
MIME, 1587
Modelo de contenido,
activar o desactivar atributos, 77
crear un modelo básico, 54
guardar el diagrama, 1373
Modelos de contenido,
de los componentes del esquema, 190
Modelos de contenido abierto, 209
Modelos de contenido abierto predeterminado, 209
Modelos de grupo, 182
Modo,
compatible con v2005R3, 1239
Modo de compatibilidad,
v2005R3, 1239
Modo generador, 590
Modo XSD, 173
MongoDB y BSON, 742
Mostrar diagrama, 1382
Mostrar marcado en tamaño grande, 614, 617
Mostrar marcado en tamaño pequeño, 614, 617
Mostrar marcado mixto, 614, 617
Mostrar nodos en ejecución,
depurador XSLT, 1407
Mostrar nodos globales, 1382
Motores,
de los productos de Altova, 2248
MS Access,
conversión de tipos de datos en esquemas XML generados a partir de BD, 2234
tipos de datos de esquemas XML, 2240
MS SQL Server,
configuración del esquema, 1385, 1386
conversión de tipos de datos en esquemas XML generados a partir de BD, 2235
extensiones de esquema, 1384
tipos de datos de esquemas XML, 2242
MS Visual Source Safe, 1097
MSXML, 1612
Multiusuario, 1584
MySQL,
conectarse por ODBC, 1016
conversión de tipos de datos en esquemas XML generados a partir de BD, 2236
tipos de datos de esquemas XML, 2244

N

Navegación,
accesos directos en el diseñador de esquemas, 81
Nillable,
no compatible, 1238
Nodo,
mostrar nodo en ejecución, 1407
Nombre de la aplicación, 1239
Novedades, 30
Números de línea en la vista Texto, 149

O

OASIS,
Catálogo XML, 1337
Objetos globales,
de SPL, 1244
Ocultar marcado, 614, 617
Ocultar marcado (en la vista Authentic), 1419
Ocultar ventanas automáticamente, 119
ODBC,
como interfaz de conexión de datos, 952
conectarse a MariaDB, 1003
conectarse a Teradata, 1035
configurar una conexión, 971
conversión de tipos de datos en esquemas XML generados a partir de BD, 2236
Office Open XML, 281, 938
OIM, 936, 1482, 1483
OLE DB,
como interfaz de conexión de datos, 952
OOXML,
ver Office Open XML, 281, 938
Opciones,
características de los gráficos circulares, 384
características de los gráficos de áreas, 384
características de los gráficos de barras, 384
características de los gráficos de líneas, 384

© 2016-2022 Altova GmbH

Opciones,
características de los gráficos de velas, 384
características de los medidores, 384
colores del gráfico, 388
cuadrícula del gráfico, 390, 393, 394
eje X del gráfico, 390
eje Y del gráfico, 393
eje Z del gráfico, 394
fondo del gráfico, 382
fuentes del gráfico, 398
gráficos 3D, 395
leyenda del gráfico, 382
tamaños del gráfico, 396
título del gráfico, 382
XQuery/XQuery Update, 1616
Opciones de impresión, 1275
Opciones de la copia de seguridad automática, 1584
Opciones de ortografía, 1542
Opciones de soporte, 139
Opciones de validación, 1584
Opciones para abrir,
archivos, 1584
Opciones XULE, 1624
Open Office XML,
archivos de ejemplo, 942
crear en la vista Archivo, 940
editar en la vista Archivo, 940
openContent, 182
Optimizador de velocidad XSL, 503, 1394
opciones, 1616
Oracle,
conectarse por JDBC, 1018
conectarse por ODBC, 1020
configuración del esquema, 1384
conversión de tipos de datos en esquemas XML generados a partir de BD, 2237
extensiones de esquema, 1383
tipos de datos de esquemas XML, 2246
Oracle XML DB, 1447
administrar esquemas XML, 1447
examinar base de datos, 1450
Override (elementos), 182
Overview,
of XMLSpy API, 1687

P

Página de códigos, 1606
Panel AARID, 212
Panel de consulta,
en la ventana de consulta de base de datos, 1432
Panel Explorador,
de la ventana de consulta de BD, 1428
Paquete base, 1239
Paquete de integración de XMLSpy, 1124, 1130
Parámetros,
en las consultas de BD, 640
pasarlos a la hoja de estilos a través de la interfaz, 1396
Parámetros de tabla, 912
Parámetros XSLT,
pasarlos a la hoja de estilos a través de la interfaz, 1396
Parent, 1836
Paso a paso para salir,
depurador XSLT, 1406
Paso a paso por instrucciones,
depurador XSLT, 1406
Paso a paso por procedimientos,
depurador XSLT, 1406
PDF,
transformar en PDF en XMLSpy, 493
Pegar,
como texto, 627
como XML, 627
Pegar (comando), 1280
Pegar como,
Texto, 624
XML, 624
Período de evaluación,
de los productos de software de Altova, 2251
Personalización, 136
Personalizar, 1573
comandos de menú/barra de herramientas, 1564
macros, 1575
menú, 1573
menú contextual, 1573
menú contextual Personalizar, 1580
Personalizar catálogos, 459
Perspectiva de XMLSpy en Eclipse, 1132
Perspectiva XMLSpy en Eclipse, 1135

© 2016-2022 Altova GmbH

Perspectivas del depurador de XMLSpy en Eclipse, 1137
Pila de llamadas (XPath/XQuery), 582
Plantillas,
de documentos XML en la vista Authentic, 1412
Plantillas con nombre, 497
Plantillas en la vista Authentic, 600
Plantillas XSLT,
en la ventana Esquema XSL, 497
Plataforma Eclipse,
Perspectiva de XMLSpy en la, 1132
perspectiva XMLSpy en, 1135
perspectivas del depurador de XMLSpy, 1137
y el paquete de integración de XMLSpy, 1130
y XMLSpy, 1129
Plataformas,
para productos de Altova, 2248
Plegamiento de código en la vista Texto, 149
Posición, 1488
Carácter, 1488
Línea, 1488
PostgreSQL,
conectarse directamente (de forma nativa), 979
conectarse por ODBC, 1026
PowerPoint 2007, 281, 938
Predeterminada,
codificación, 1589
Predeterminado,
menú, 1573
Prefijo de espacio de nombres,
insertar en la vista Cuadrícula, 1344
Presentación, 1597
Pretty-print,
en la vista Texto, 1288
Procesador XQuery,
de los productos de Altova, 2248
Procesador XSL externo, 1612
Procesadores XSLT,
de los productos de Altova, 2248
Procesar XULE, 1534
Proceso de la comunicación,
depurador SOAP, 792
Proceso de la comunicación SOAP, 792
Productos de Altova, 139
Progress OpenEdge (base de datos),
conectarse por JDBC, 1027
conectarse por ODBC, 1029
Proveedor,
de control de código fuente, 1097
Proyecto,
abrir, 1301
agregar archivo activo al, 1318
agregar archivos al, 1317
agregar carpetas al, 1318
agregar carpetas externas al, 1319
agregar carpetas web externas al, 1322
agregar recursos globales al, 1317
agregar URL al, 1317
agregarlo al control de código fuente, 1309
cerrar, 1301
crear nuevo, 1301
guardar, 1302
propiedades, 1328
volver a cargar, 1301
Proyectos,
agregar archivos relacionados al, 1318
cómo crear y editar, 1061
guardar, 1061
introducción, 1298
nombrar, 1061
procesamiento por lotes con, 1066
propiedades de, 1061
usados recientemente, 1331
usar, 1066
ventajas de usar, 1066
vista general de, 1060
Proyectos en XMLSpy,
creación de, 114
ventajas, 114
PUBLIC,
identificador - catálogo, 1337
Punto de interrupción,
cuadro de diálogo, 1408
Punto de seguimiento,
cuadro de diálogo, 1408
Puntos de depuración en el depuradorXPath/XQuery, 582
Puntos de interrupción,
uso en el depurador SOAP, 802
uso en el depurador XSLT/XQuery, 548
Puntos de interrupción (cuadro de diálogo), 534
Puntos de interrupción en el depurador XPath/XQuery, 582
Puntos de seguimiento,
uso en el depurador XSLT/XQuery, 551

© 2016-2022 Altova GmbH

Puntos de seguimiento en el depurador XPath/XQuery, 582
PVCS Version Manager, 1097

Q

Quitar,
punto de interrupción - depurador XSLT, 1407
punto de seguimiento - depurador XSLT, 1407

R

RaptorXML,
configuración activa, 1563
gestionar servidores, 1559
Recargar, 1584
Recursos,
acelerar actualización de estado en segundo plano, 1097
Recursos globales, 1042
activación de barras de herramientas, 1566
cambiar de configuración, 1058
configuración activa para, 1558
definir, 1043, 1557
definir el tipo de archivo, 1045
definir el tipo de base de datos, 1051
definir el tipo de carpeta, 1050
usar, 1054, 1057, 1058
usar tipo de archivo y tipo de carpeta, 1054
Recursos globales de Altova,
ver Recursos globales, 1042
Redefine (elementos), 182
Reemplazar, 129
texto, 1289
texto en el documento, 1293
texto en varios archivos, 1294
Referencia del programador, 1639
Regiones en los scripts SQL, 1432
Registrar,
complemento, 1669
Registrar su software de Altova, 1633
Registro,
configuración, 1583
Reglas,
para validación de esquemas (ver Reglas de esquema), 449
Reglas de esquema, 449
agregar conjuntos de reglas a un esquema, 449
definir, 451
Rehacer (comando), 1280
Reiniciar,
depurador XSLT, 1405
Relaciones de elementos sin nombre,
configuración del esquema de MS SQL Server, 1386
Relaciones de esquema con nombre,
configuración del esquema de MS SQL Server, 1385
Relaciones en Taxonomías, 851, 853, 855
Repeticiones,
número de, 54
Repositorios, 1097
Requisitos de memoria, 2248
Reservar,
nombre de método, 1238
Restablecer,
acceso rápido, 1569
barras de herramientas y comandos de menú, 1566
comandos de menú, 1573
Restricción de identidad,
activar o desactivar atributos en la vista del modelo de contenido, 77
Restricciones de identidad en la vista Esquema, 212, 220
Restricciones inteligentes, 244
Resumen del funcionamiento de los catálogos, 456
Retorno de carro,
ver Tecla Entrar, 652
Revisar,
ortografía, 1539
Revisión de sintaxis de documentos JSON, 674
Revisión ortográfica, 1539
diccionario personalizado, 1539
RichEdit, 1420
RichEdit 3.0, 1603
RootCatalog.xml, 457
Rutas de acceso de archivos,
insertar en documento XML, 290

S

Saltos de línea, 1584
Sangría,
en la vista Texto, 1288
Sangría en la vista Texto, 146

© 2016-2022 Altova GmbH

save, 1838
Save as image, 1287
schema, 1736
SchemaAgent, 472
abrir esquemas desde XMLSpy, 472
conectarse al servidor desde XMLSpy, 1387
desconectarse del servidor, 1388
mostrar esquemas en, 1388
trabajar con, 466, 467, 472
validación con, 1388
SchemaAgent en XMLSpy, 463
SchemaAgent Server,
conectarse a, 464
SchemaDocumentationDlg,
AllDetails, 1879
Application, 1880
IncludeAll, 1881
IncludeAttributeGroups, 1882
IncludeComplexTypes, 1882
IncludeGlobalElements, 1882
IncludeGroups, 1883
IncludeIndex, 1883
IncludeLocalElements, 1884
IncludeRedefines, 1884
IncludeSimpleTypes, 1884
OptionsDialogAction, 1885
OutputFile, 1885
OutputFileDialogAction, 1886
OutputFormat, 1886
Parent, 1886
ShowAnnotations, 1886
ShowAttributes, 1887
ShowChildren, 1887
ShowConstraints, 1888
ShowDiagram, 1887
ShowEnumerations, 1888
ShowNamespace, 1888
ShowPatterns, 1889
ShowProgressBar, 1889
ShowProperties, 1889
ShowResult, 1889
ShowSingleFacets, 1890
ShowSourceCode, 1890
ShowType, 1890
ShowUsedBy, 1891
schemanativetype, 1242
Script, 1625
Scripts de Authentic,
configuración de seguridad, 1421
ubicaciones de confianza, 1421
Scripts en XSLT/XQuery,
ver Funciones de extensión, 2213
Secciones CDATA,
insertar en la vista Authentic, 627
Segundo plano,
actualización de estado - aumentar el intervalo, 1097
Seleccionar todo (comando), 1289
Selector de fecha,
usar en la vista Authentic, 646
Servicio web,
conectarse a, 783
Servidor web, 1636
Settings,
overview of, 49
SharePoint® Server, 1322
Sistema de ayuda, 1631
SO,
para productos de Altova, 2248
SOAP, 775, 788, 1506
crear solicitud nueva, 1504
detener el servidor proxy, 1513
enviar solicitud al servidor, 1506
enviar solicitud desde WSDL, 785
iniciar el servidor proxy, 1512
opciones del depurador, 1513
parámetros de la solicitud, 1507
sesión de depurador, 1511
solicitud, 1513
solicitudes, 1506, 1512, 1513
Soporte técnico, 1636
Soporte técnico de Altova, 139
SPL, 1241
bloques de código, 1241
condiciones, 1247
foreach, 1248
objetos globales, 1244
subrutinas, 1250
usar archivos, 1246
variables, 1243
SPS,
asignar a un archivo XML nuevo, 1258
SpyProject,
CloseProject, 1892
ProjectFile, 1892
RootItems, 1892
SaveProject, 1892

© 2016-2022 Altova GmbH

SpyProject,
SaveProjectAs, 1893
SpyProjectItem,
ChildItems, 1893
FileExtensions, 1893
ItemType, 1894
Name, 1894
Open, 1894
ParentItem, 1894
Path, 1894
ValidateWith, 1894
XMLForXSLTransformation, 1895
XSLForXMLTransformation, 1895
XSLTransformationFileExtension, 1895
XSLTransformationFolder, 1895
SpyProjectItems,
AddFile, 1896
AddFolder, 1896
AddURL, 1896
Count, 1897
Item, 1897
RemoveItem, 1897
SQL Server,
administrar esquemas XML, 1445
conectarse por ADO, 958
conectarse por ADO.NET, 964
SQLite,
configurar una conexión (Windows), 980
StarTeam, 1097
StyleVision, 1365
para editar la hoja de estilos StyleVision Power Stylesheet, 1413
y XBRL, 1528
Subconjuntos de esquema, 1389
Subconjuntos de esquemas, 445, 1390
Subir/bajar,
fila (vista Authentic), 1421
Sybase,
conectarse por JDBC, 1032
conversión de tipos de datos en esquemas XML generados a partir de BD, 2238

T

Tabla,
generar automáticamente, 1587
Tabla de datos de gráfico,
cómo se construye, 349
Tablas,
editar tablas dinámicas (SPS), 614
en la vista Authentic, 604
Tablas (SPS) dinámicas en la vista Authentic,
uso de, 632
Tablas (SPS) estáticas en la vista Authentic,
uso de, 632
Tablas de en la vista Cuadrícula (XML),
Copiar como TSV o XML, 326
Tablas dinámicas,
edición, 614
Tablas en la vista Authentic,
iconos para editar tablas XML, 637
usar tablas SPS (estáticas y dinámicas), 632
usar tablas XML, 633
uso de, 632
Tablas en la vista Cuadrícula (JSON),
copiar como TSV o JSON, 709
Tablas en la vista Cuadrícula JSON,
copiar como TSV o JSON, 709
Tablas en la vista Cuadrícula XML,
Copiar como TSV o XML, 326
Tablas SPS,
editar tablas dinámicas, 614
Tablas SPS en la vista Authentic,
uso de, 632
Tablas XML en la vista Authentic,
iconos para editar, 637
uso de, 633
Table Display commands (JSON Grid View), 1334, 1353
Tamaño, 1603
Tamaño de tabulación,
configurar, 1489
y opción pretty-print, 1489
Taxonomías,
agregar elementos a, 847
archivos en, 835, 845
crear nueva, 838
espacio de nombres de destino de, 843
espacios de nombres de destino en, 1519
espacios de nombres en, 843, 1519
importar, 839, 1521
pasos para crear, 834
relaciones en, 851, 853, 855
y asistente para la creación de taxonomías, 838
y bases de enlaces, 845

© 2016-2022 Altova GmbH

Taxonomías,
y funciones de vínculo, 851
Taxonomías en XBRL, 834, 835
Taxonomías XBRL, 834
ver Taxonomías, 835
Tecla de acceso rápido, 1569
Tecla Entrar,
efectos de usar, 652
Teclas de acceso rápido, 283, 1569
mostrar en información rápida, 1579
Tema,
ver en la tabla de contenido, 1631
Teradata,
conectarse por JDBC, 1033
conectarse por ODBC, 1035
terminate, 1739
TextImportExportSettings,
DestinationFolder, 1898
EnclosingCharacter, 1898
Encoding, 1898
EncodingByteOrder, 1898
FieldDelimiter, 1898
FileExtension, 1899
HeaderRow, 1899
ImportFile, 1899
Texto,
buscar en el documento, 1289
buscar y reemplazar, 1293
editar en la vista Authentic, 627
formato en la vista Authentic, 627
fuentes, 1603
prepara para pretty-print, 1288
Texto estructurado, 1590
Texto JSON de fuentes externas,
añadir rápidamente a documento, 756
Texto XML, 1590
copiar, 1281
Texto XML de fuentes externas,
añadir rápidamente a documento, 339
tipo,
extensión en documento XML, 90
Tipo base,
modificar, 242
tipo de elemento,
especificar tipo de elemento en documento XML, 90
Tipo simple,
definir facetas de, 234
en definiciones de esquema, 67
facetas assertion en, 234
facetas enumeration de, 234
facetas pattern de, 234
valores de muestra de, 234
Tipos,
integrados, 1252
Tipos alternativos condicionales, 205
Tipos complejos,
anónimos, 182
con nombre, 182
globales, 182
Tipos de archivo, 1587
Tipos de datos,
conversión de BD en XML Schema, 2233
para bases de datos generadas a partir de esquemas XML, 2240
ver Tipos de datos XML Schema, 2234
Tipos de datos XML Schema,
al generar esquemas XML a partir de BD ADO, 2233
al generar esquemas XML a partir de BD MS Access, 2234
al generar esquemas XML a partir de BD MS SQL Server, 2235
al generar esquemas XML a partir de BD MySQL, 2236
al generar esquemas XML a partir de BD ODBC, 2236
al generar esquemas XML a partir de BD Oracle, 2237
al generar esquemas XML a partir de BD Sybase, 2238
convertidos en tipos de datos MS Access, 2240
convertidos en tipos de datos MS SQL Server, 2242
convertidos en tipos de datos MySQL, 2244
convertidos en tipos de datos Oracle, 2246
Tipos derivados,
modificar el tipo base de, 242
Tipos simple,
anónimos, 182
con nombre, 182
globales, 182
Título de la pantalla, 1597
Transformación,
ver Transformación XSLT, 1395
Transformación XSL,
ver XSLT, 110
Transformación XSLT, 1394, 1395
asignar archivo XSLT, 110
en FO, 1395
en PDF, 1395
en XMLSpy, 111
tutorial, 110
Transformaciones JSON con XSLT/XQuery, 758

© 2016-2022 Altova GmbH

TSV,
copiar tablas de la vista Cuadrícula JSON como, 709
Copiar tablas XML de la vista Cuadrícula como, 326
Tutorial,
para WSDL, 776
Tutorial de WSDL, 776

U

Ubicaciones de confianza para scripts de Authentic, 1421
Ubicaciones del archivo SPP, 1298
UCS-2, 1589
UML,
convertir esquemas en, 1363
Update Entry Helpers command, 1344
URL, 1742, 1838, 1846, 1847
enviar por correo electrónico, 1274
Uso de Internet,
en los productos de Altova, 2249
UTF-16, 1589

V

Validación, 136, 1337
asignar DTD al documento XML, 1355
asignar esquema XML al documento XML, 1356
con SchemaAgent, 1388
de esquemas relacionados con SchemaAgent, 472
extender con reglas de esquema, 449
WSDL, 1343
Validación automática, 1587
Validación con SchemaAgent, 472
Validación de esquemas XML, 238
Validación EDGAR en el servidor, 1536
Validación extendida de esquemas, 449
Validación SOAP, 788
Validación XSD, 238
Validar,
archivo WSDL, 783
documentos XML, 98
Validar al modificar el comando, 1344
Validar documentos XML, 288
Validar EDGAR, 1536
Valores de atributo,
introducir en la vista Authentic, 609
Valores de muestra de tipos simples, 234
Variables,
de SPL, 1243
Variables de entorno que se usan en los catálogos, 457
Variables en el depurador XPath/XQuery, 582
Velocidad,
acelerar actualización de estado en segundo plano, 1097
Ventana Contexto,
del depurador XSLT/XQuery, 542
ventana de información,
para documentos CSS, 659
para documentos HTML, 657
Ventana de información HTML, 657
Ventana de información, pestaña XSLT,
descripción de, 500
ver también Esquema XSL, 500
y crear carpetas ZIP, 500
y documentos XSLT, 500
y Proyectos, 500
Ventana de seguimiento, 551
Ventana Esquema XSL, 128, 497
Ventana Gráficos, 132
Ventana HTTP,
enviar la solicitud, 806
importar solicitud, 811
métodos HTTP, 806
recibir una respuesta, 814
y códigos de estado HTTP, 814
y encabezado Aceptar, 814
y WADL, 811
Ventana Información, 124
del depurador XSLT/XQuery, 546
Ventana Información CSS, 659
Ventana Inspección XPath,
del depurador XSLT/XQuery, 543
ventana Mensajes, 125
del depurador XSLT/XQuery, 544
Ventana Pila de llamadas,
del depurador XSLT/XQuery, 544
Ventana Plantillas,
del depurador XSLT/XQuery, 545
Ventana principal, 120
Ventana Proyecto, 122
Ventana Seguimiento,
del depurador XSLT/XQuery, 546
Ventana Variables,
del depurador XSLT/XQuery, 542

© 2016-2022 Altova GmbH

Ventana XPath, 127
Ventana XPath/XQuery, 573, 574
Ventana XULE, 134
para consultas interactivas en instancias XBRL, 924
Ventanas,
administrar la visualización de, 119
flotantes, acopladas, en pestañas, 119
ocultar automáticamente, 119
Ventanas acopladas, 119
Ventanas de información,
organizar, 546
Ventanas del depurador,
organización, 546
Ventanas flotantes, 119
Ver expresiones en el depurador XPath/XQuery, 582
Ver tablas en la vista Cuadrícula JSON,
editar en, 692
exportar a hojas de cálculo, 692
importar desde hojas de cálculo, 692
Versión del esquema JSON, 715
Vista,
Ancho de tabla óptimo, 1488
Contraer, 1487
Expandir, 1487
Ir a la línea o carácter, 1488
Ir al archivo, 1489
Vista Cuadrícula mejorada, 1485
Vista Diseño de esquemas, 1485
Vista Explorador, 1486
Vista Texto, 1484
vista Archivo, 281, 938
y archivos EPUB, 945
y archivos OOXML, 940
y archivos ZIP, 943
Vista Authentic,
abrir archivo XML nuevo en, 1412
abrir un documento XML en, 600
agregar nodos, 604
aplicar elementos, 604
ayudantes de entrada, 601
ayudantes de entrada en, 619
borrar elementos, 604
cambiar a, 1486
caracteres especiales en la, 607
edición de datos de BD en, 1412
editar datos en una BD XML, 1414
editar XML en, 331
entidades en la, 607
formato de texto en, 614
generar documentos de salida a partir del archivo PXF, 1421
iconos de la barra de herramientas, 614
iconos para tablas XML, 637
imprimir un documento XML desde la, 611
insertar entidades en la, 610
insertar nodos, 604
interfaz, 613
interfaz gráfica del usuario (IGU), 613
introducir datos en la, 607
introducir valores de atributo, 609
menú contextual, 601
menús contextuales, 624
mostrar etiquetas de marcado, 601
pegar como XML o como texto, 624
presentación de marcado en, 617
presentación del marcado en, 614
quitar nodos, 604
secciones CDATA en la, 607
tablas (SPS y XML), 632
tablas en la, 604
Tablas SPS, 632
tablas XML, 633
Tutorial, 599
uso de funciones importantes, 627
uso de tablas XML, 633
ventana principal de, 617
visualización de documentos, 617
Vista Avro, 773
Vista Cuadrícula, 102, 169, 1485, 1487
anexar elementos y atributos, 102
introducir datos, 92
usar los ayudantes de entrada, 102
Y vista Diseño de tabla, 104
Vista Cuadrícula (JSON),
abrir la vista Cuadrícula JSON, 679
Vista Cuadrícula (XML), 293
agregar acción eliminar nodos, 299
archivo de metadatos, 314, 316
arrastrar y colocar, 306
ayudantes de entrada, 299
buscar y reemplazar, 302
características de, 295
crear gráficos en, 320
editar el contenido de un documento, 302
editar la estructura del documento, 299
Editar la Vista Diseño de tabla, 309
editar tipos de nodo, 302

© 2016-2022 Altova GmbH

Vista Cuadrícula (XML), 293
expresiones XQuery en, 314, 316
filtrar en la, 316
fórmulas para calcular un resultado, 314
imágenes en la, 319
menú contextual de, 323
validación, 302
visualizar documentos, 295
Vista Cuadrícula de documentos JSON,
abrir la vista Cuadrícula JSON, 679
Vista Cuadrícula JSON, 679
agregar componentes como secundarios, 1333, 1352
anexar componentes, 1333, 1352
archivo de metadatos, 696, 697
arrastrar y colocar, 689
comando Tipo, 1333, 1351
comandos del menú contextual en, 706
comandos dentro de las celdas en, 685, 686
configurar pretty-print, 1354
crear gráficos en, 703
editar el contenido del documento en, 686
editar la estructura del documento en, 685
expresiones XQuery en, 696, 697
filtrar la vista, 696
finalización automática, 686
fórmulas para calcular un resultado, 696, 697
Iconos de la vista Diseño de tabla, 692
insertar componentes, 1333, 1352
símbolos de distintos tipos, 682
ver documento en, 682
ver imágenes en, 702
Vista Diseño de tabla, 692
volver a evaluar todos los filtros y las fórmulas, 1354
Vista Cuadrícula mejorada, 1485
ver Vista Cuadrícula, 92
Vista Cuadrícula XML,
Ver la vista Cuadrícula, 293
Vista del modelo de contenido, 51
asignar tipos condicionales, 205
compositores y componentes, 192
configurar, 1378
descripción de la interfaz, 192
descripción general, 190
editar en, 199
objetos de los diagramas, 192
Vista Diseño de esquemas,
función de zoom, 1381
Mostrar diagrama, 1382
Mostrar nodos globales, 1382
Vista Diseño de tabla,
funcionamiento, 104
Vista Diseño de tabla la vista Cuadrícula XML,
editar en la, 309
importar/exportar a/desde hojas de cálculo, 309
Vista Diseño WSDL,
descripción de, 251
Enlaces, 252
funciones, 251
PortTypes, 252
Servicios, 252
Ventana principal, 252
ver archivos en, 251
Vista dividida en Vista Texto, 165
Vista Esquema, 171
ayudante de entrada Componentes, 229
ayudante de entrada Detalles, 232
ayudante de entrada Facetas, 234
ayudantes de entrada, 228
configurar la vista, 60
navegar hacia delante y hacia detrás, 250
Vista Esquema JSON, 714
agregar esquemas externos, 718
ayudante de entrada Detalles, 718
ayudante de entrada Restricciones, 718
ayudantes de entrada, 718
configurar, 749
definiciones globals y locales, 721
vista Diseño, 723
Vista Esquema, buscar en,
ver Búsqueda en esquemas, 475
Vista Explorador, 279, 1491
Vista general del esquema, 51
editar en, 179
iconos de, 179
ordenar componentes en, 179
y comentarios globales, 179
y presentación de la línea de los comentarios globales, 179
y vista del modelo de contenido, 177, 179
Vista predeterminada,
configuración de la ventana principal, 1587
Vista previa de los atributos, 1597
Vista previa del diseño de tabla, 916
generar fórmulas, 918
Vista Tabla, 1590
Vista Tabla o de base de datos,
funcionamiento, 104

© 2016-2022 Altova GmbH

Vista Texto, 145, 1484
ajuste automático de línea, 146
Ayudantes de entrada en la, 164
edición en la, 93
formato de texto, 146
fuentes del esquema, 1608
funciones de edición especiales para documentos XML, 290
funciones de edición inteligente, 151
habilitar/deshabilitar finalización automática, 1590
marcadores en la, 149
números de línea en la, 149
plegamiento de código fuente en la, 149
propiedades de las fuentes, 146
sangría, 146
sangría en la, 149
teclas de acceso rápido, 167
Vista dividida en, 165
y comentarios en documentos XML, 290
y líneas vacías en documentos XML, 290
Vista WSDL,
ayudante de entrada Detalles, 256
ayudantes de entrada, 256
ayudantes de entrada Vista general, 256
importar archivos al documento WSDL, 256
Vista XBRL, 264
ayudantes de entrada de, 272
configurar, 1527
Pestaña Cálculo de la ventana principal, 269
Pestaña Definición de la ventana principal, 269
Pestaña Elementos de la ventana principal, 265
Pestaña Presentación de la ventana principal, 269
Vista XBRL, buscar en,
ver Búsqueda en XBRL, 930
Vista XML,
agregar componentes, 54
configurar la vista, 60
crea un archivo nuevo, 51
Vistas de edición, 141
Visual Studio,
adding the XMLSpy ActiveX Controls to the toolbox, 2053
Visual Studio .Net,
diferencias con XMLSpy, 1125
y los depuradores de XMLSpy, 1128
y XMLSpy, 1123
Volcar a cargar,
archivos cambiados, 1267
VS .NET,
y el paquete de integración de XMLSpy, 1124

W

WADL,
usar para solicitudes HTTP, 811
Windows,
compatibilidad con productos de Altova, 2248
Word 2007, 281, 938
WSDL, 775, 783
abrir esquema, 783
componentes 1.1, 1492
componentes 2.0, 1495
conectarse a un servicio web, 783
converting from 1.1 to 2.0, 1503
crear documentación, 786
crear documento nuevo, 776
crear enlaces, 780
crear mensajes, 777
crear operaciones, 777
crear parámetros, 777
crear portTypes, 777
crear puertos, 781
crear servicios, 781
ejemplo del depurador SOAP, 790
enlaces en WSDL 1.1, 1494
enlaces en WSDL 2.0, 1497
enviar solicitud SOAP, 785
espacios de nombres, 776
generar documentación, 1499
interfaces en WSDL 2.0, 1495
mensajes en WSDL 1.1, 1492
operaciones en WSDL 1.1, 1493
portType en WSDL 1.1, 1494
reparse document, 1503
servicio web, 1504
servicios en WSDL 1.1, 1495
servicios en WSDL 2.0, 1498
tipos, 1498
validar, 783

X

XBRL, 817
espacios de nombres, 1519

© 2016-2022 Altova GmbH

XBRL, 817
espacios de nombres de destino, 1519
funciones de arco, 1515
funciones de vínculo, 1517
generar documentación, 1523
validación, 937
XULE, 921
y MapForce, 1528
y StyleVision, 1528
XBRL Formula,
editor, 864
XBRL inline,
gestionar espacios en blanco de, 1620
xBRL-CSV (OIM), 1483
xBRL-JSON (OIM), 1482
xBRL-XML (OIM), 1482
Xerces,
compatibilidad, 1239
XInclude,
insertar en documento XML, 290
insertar en la vista Cuadrícula, 1284
insertar en la vista Texto, 1284
XML,
Catálogo Oasis, 1337
copiar como texto estructurado, 1282
revisión ortográfica, 1539
XML Schema,
componentes globales, 182
xml:base, 248
xml:id, 248
xml:lang, 248
xml:space, 248
XMLData,
AppendChild, 1940
EraseAllChildren, 1941
EraseCurrentChild, 1942
GetChild, 1943
GetChildKind, 1944
GetCurrentChild, 1944
GetFirstChild, 1944
GetNextChild, 1945
HasChildren, 1947
HasChildrenKind, 1947
InsertChild, 1947
IsSameNode, 1948
Kind, 1949
MayHaveChildren, 1949
Name, 1949
Parent, 1949
TextValue, 1950
XMLSpy, 1257
ayuda, 139
complemento, 1669
funciones, 139
integration, 2052
registro del complemento, 1669
XMLSpy API,
documentation, 1686
overview, 1687
XMLSpy en Eclipse, 1129
XMLSpy Enterprise Edition,
Guía y referencia del usuario, 29
XMLSpyCommand,
in XMLSpyControl, 2092
XMLSpyCommands,
in XMLSpyControl, 2094
XMLSpyControl, 2095
documentation of, 2052
examples of integration at document level, 2060
integration using C#, 2060
object reference, 2091
XMLSpyControlDocument, 2102
XMLSpyControlPlaceHolder, 2109
XMLSpyDocumentEditor,
MarkUpView, 1982
XMLSpyLib, 1686, 1687
Application, 1724
AuthenticDataTransfer, 1745
AuthenticRange, 1750
AuthenticSelection (obsolete), 1963
AuthenticView, 1778
CodeGeneratorDlg, 1797
DatabaseConnection, 1803
Dialogs, 1809
DocEditEvent (obsolete), 1950
DocEditView (obsolete), 1965
Document, 1813
Documents, 1843
DTDSchemaGeneratorDlg, 1847
ElementList, 1851
ElementListItem, 1852
ExportSettings, 1853
FileSelectionDlg, 1856
GenerateSampleXMLDlg, 1869
GridView, 1875
ProjectItem, 1893

© 2016-2022 Altova GmbH

XMLSpyLib, 1686, 1687
SchemaDocumentationDlg, 1878
SpyProject, 1891
SpyProjectItems, 1895
TextImportExportSettings, 1897
XMLData, 1939
XPath,
evaluar, 127
originada de un nodo de un documento XML, 290
XPath 1.0,
en el evaluador de XPath, 1335
XPath 2.0,
en el evaluador de XPath, 1335
XPath al nodo seleccionado, 613
XPath del nodo seleccionado en el documento XML,
copiar en el portapapeles, 1283
XPaths,
configuración para puntos de seguimiento, 551
XPointer,
originado de un nodo de un documento XML, 290
XPointer del nodo seleccionado en el documento XML,
copiar en el portapapeles, 1283
XQuery,
abrir un archivo, 507
ayudantes de entrada, 509
color de sintaxis, 510
compatibilidad con BD, 526
edición, 507
edición en la vista Texto, 505
evaluación, 514
funciones de edición inteligente, 510
funciones de extensión, 2213
funciones para IBM DB2, 526
para consultar bases de datos XML, 526
pasar variables al documento XQuery, 1396
validación, 515
XQuery Update, 516
lista de actualizaciones pendientes, 519
vista previa, 519
XQuery Update Facility, 519
XQuery/XQuery Update,
opciones, 1616
XSD 1.0 y 1.1,
modos de edición, 173
xsi:type,
uso, 90
XSL,
optimizar, 503, 1394
ver XSLT, 1404
xsl:call-template, 497
XSL:FO,
y transformaciones XSLT, 493
xsl:param, 497
xsl:with-param, 497
XSLT, 1488
ayudantes de entrada para, 491
documentos, 491
finalización automática en la vista Texto, 491
funciones de extensión, 2213
funciones en XMLSpy, 490
modificar XSLT en XMLSpy, 112
procesador, 1612
transformaciones en XMLSpy, 493
validar, 491
y transformaciones por lotes, 493
XULE para XBRL, 921

Z

Zoom en la vista Texto, 149

© 2016-2022 Altova GmbH