Altova StyleVision 2023 Professional Edition

Manual del usuario y referencia

Altova StyleVision 2023 Professional 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 Introducción 17

1.1 Versión 2023: novedades 18
- 1.1.1 Versión 2020 18
- 1.1.2 Versión 2019 19
- 1.1.3 Versión 2018 19
- 1.1.4 Versión 2017 20

1.2 Características del producto 21

1.3 La vista Authentic en los productos Altova 25

1.4 ¿Qué es una hoja de estilos SPS? 26

1.5 Configurar StyleVision 28

1.6 Terminología 29

1.7 Información preliminar 33

2 Interfaz del usuario 35

2.1 Ventana principal 36
- 2.1.1 Vista Diseño 37
- 2.1.2 Vista Authentic 38
- 2.1.3 Vistas de resultados 39

2.2 Barras laterales 41
- 2.2.1 Vista general del diseño 45
- 2.2.2 Estructura del esquema 48
- 2.2.3 Estructura del diseño 52
- 2.2.4 Repositorio de estilos 55
- 2.2.5 Estilos 58
- 2.2.6 Propiedades 59
- 2.2.7 Proyecto 63
- 2.2.8 Mensajes 65
- 2.2.9 Buscar y reemplazar 65

3 Tutorial de introducción rápida 67

3.1 Crear un archivo SPS nuevo 68
3.2 Insertar contenido dinámico (de una fuente de datos XML) 73
3.3 Insertar contenido estático 80
3.4 Aplicar formato al contenido 86
3.5 Usar cálculos automáticos 93
3.6 Usar condiciones 98
3.7 Usar plantillas globales 106
3.8 ¡Eso es todo! 111

4 Resumen de características 112

4.1 Archivos SPS y fuentes de datos 113
4.2 Crear el diseño 114
4.3 Versiones XSLT y XPath 115
4.4 Compatibilidad con Internet Explorer 116
4.5 Archivos SPS y la vista Authentic 118
4.6 Sincronizar StyleVision y Authentic 120
4.7 Archivos generados 121
4.8 Proyectos de StyleVision 123
4.9 Catálogos en StyleVision 128
4.9.1 Funcionamiento de los catálogos 128
4.9.2 Estructura del catálogo en StyleVision 129
4.9.3 Personalizar catálogos 131
4.9.4 Variables para ubicaciones del sistema en Windows 132

5 Archivos SPS: contenido 134

5.1 Insertar contenido XML como texto 135
5.1.1 Insertar contenido con formato predefinido 137
5.1.2 Agregar elementos en la vista Authentic 138
5.1.3 Resto de contenidos 141
5.2 Insertar contenido de MS Word 143
5.3 Insertar contenido de MS Excel 146

5.4 Plantillas definidas por el usuario ... 148
5.5 Elementos y bloques de texto definidos por el usuario ... 151
5.5.1 Elementos definidos por el usuario ... 151
5.5.2 Bloques definidos por el usuario ... 152
5.6 Tablas ... 155
5.6.1 Tablas estáticas ... 157
5.6.2 Tablas dinámicas ... 158
5.6.3 Procesamiento condicional en tablas ... 162
5.6.4 Tablas en la vista Diseño ... 164
5.6.5 Formato de tablas ... 165
5.6.6 Presentación de filas y columnas ... 170
5.6.7 Tablas CALS/HTML ... 172
5.7 Listas ... 178
5.7.1 Listas estáticas ... 178
5.7.2 Listas dinámicas ... 180
5.8 Contenido gráfico ... 183
5.8.1 Imágenes: datos insertados y URI ... 183
5.8.2 Tipos de imagen y formatos de salida ... 185
5.8.3 Ejemplo: una plantilla para imágenes ... 189
5.9 Controles de formulario ... 190
5.9.1 Campos de entrada y campos de entrada multilínea ... 191
5.9.2 Casillas de verificación ... 192
5.9.3 Cuadros combinados ... 194
5.9.4 Botones y botones de opción ... 197
5.10 Vínculos ... 199
5.11 Códigos de barras ... 200
5.12 Módulos de diseño ... 204
5.12.1 Contenedores de diseño ... 204
5.12.2 Cuadros de diseño ... 208
5.12.3 Líneas ... 212
5.13 Función Convertir en ... 215

6 Archivos SPS: estructura 219

6.1 Esquemas fuente ... 221

6.1.1 Esquemas XML y DTD ... 223
6.1.2 Esquemas de BD ... 228
6.1.3 Esquemas definidos por el usuario ... 229
6.1.4 Gestor de esquemas ... 232
6.2 Combinar datos XML de distintas fuentes ... 248
6.3 Archivos SPS modulares ... 251
6.3.1 Objetos modulares ... 252
6.3.2 Crear un archivo SPS modular ... 255
6.3.3 Ejemplo: una libreta de direcciones ... 259
6.4 Plantillas y fragmentos de diseño ... 265
6.4.1 Plantilla principal ... 265
6.4.2 Plantillas globales ... 266
6.4.3 Plantillas definidas por el usuario ... 270
6.4.4 Plantillas variables ... 273
6.4.5 Operaciones con plantillas nodo ... 274
6.4.6 Fragmentos de diseño ... 278
6.5 Plantillas XSLT ... 282
6.6 Múltiples documentos de salida ... 284
6.6.1 Insertar una plantilla de documento nuevo ... 285
6.6.2 Plantillas de documento nuevo y estructura del diseño ... 286
6.6.3 URL de las plantillas de documento nuevo ... 286
6.6.4 Vista previa de archivos y documentos de salida ... 289
6.6.5 Propiedades y estilos de los documentos ... 291

7 Archivos SPS: características avanzadas 292

7.1 Cálculos automáticos ... 293
7.1.1 Editar y mover cálculos automáticos ... 293
7.1.2 Actualizar nodos con cálculos automáticos ... 296
7.1.3 Cálculos automáticos basados en nodos actualizados ... 298
7.1.4 Ejemplo: una factura ... 299
7.2 Condiciones ... 303
7.2.1 Definir condiciones ... 303
7.2.2 Editar condiciones ... 307
7.2.3 Condiciones basadas en los resultados ... 307

7.2.4 Condiciones y cálculos automáticos ... 308
7.3 Presencia condicional ... 310
7.4 Agrupaciones ... 312
7.4.1 Ejemplo: agrupar-por (Persons.sps) ... 315
7.4.2 Ejemplo: agrupar-por (Scores.sps) ... 316
7.5 Ordenación ... 321
7.5.1 El mecanismo de ordenación ... 321
7.5.2 Ejemplo: múltiples claves de ordenación ... 323
7.6 Parámetros y variables ... 327
7.6.1 Parámetros declarados por el usuario ... 327
7.6.2 Parámetros para fragmentos de diseño ... 329
7.6.3 Parámetros del SPS para fuentes de datos ... 331
7.6.4 Variables ... 332
7.6.5 Variables editables en Authentic ... 334
7.7 Tablas de contenido, referencias y marcadores ... 337
7.7.1 Crear marcadores para incluir elementos en la TDC ... 340
7.7.2 Crear la plantilla TDC ... 347
7.7.3 Ejemplo: tabla de contenido básica ... 352
7.7.4 Ejemplo: TDC jerárquica y secuencial ... 356
7.7.5 Numeración automática en el cuerpo del diseño ... 359
7.7.6 Referencias cruzadas ... 363
7.7.7 Marcadores e hipervínculos ... 364
7.8 Ejemplo: documentos en varios idiomas ... 371

8 Archivos SPS: presentación 374

8.1 Formatos predefinidos ... 375
8.2 Caracteres de escape en documentos de salida ... 377
8.3 Formato de valores (formato de tipos de datos numéricos) ... 380
8.3.1 Funcionamiento ... 380
8.3.2 Sintaxis ... 383
8.4 Trabajar con estilos CSS ... 390
8.4.1 Hojas de estilos externas ... 391
8.4.2 Estilos globales ... 395
8.4.3 Estilos locales ... 398

8.4.4 Establecer valores de estilos ... 400
8.4.5 Propiedades de los estilos mediante XPath ... 402
8.4.6 Estilos compuestos ... 405
8.5 Flexibilidad de estilos en Authentic ... 408
8.5.1 Estilos compuestos ... 408
8.5.2 RichEdit ... 410
8.5.3 Iconos de estado del texto ... 414
8.6 Propiedades de un documento HTML ... 417
8.7 Diseñar medios impresos ... 419
8.7.1 Secciones del documento ... 420
8.7.2 Saltos de página ... 434
8.7.3 Notas al pie ... 435
8.7.4 Resolución en píxeles ... 437
8.7.5 Marcas de agua ... 439

9 Archivos SPS: otras funciones 443

9.1 Recursos globales de Altova ... 444
9.1.1 Definir recursos globales ... 444
9.1.2 Usar recursos globales ... 455
9.2 Propiedades de nodos Authentic ... 462
9.3 Al hacer clic, reemplazar el nodo primario por ... 464
9.4 Validación adicional ... 467
9.5 URI de entidades sin analizar ... 469
9.6 Diseño nuevo a partir de archivos XSLT, XSL-FO y FO ... 471
9.7 Funciones XPath definidas por el usuario ... 475
9.7.1 Definir una función XPath ... 477
9.7.2 Reutilizar funciones para encontrar nodos ... 480
9.7.3 Parámetros en funciones XPath ... 481
9.8 Trabajar con fechas ... 491
9.8.1 Usar el selector de fechas ... 492
9.8.2 Formato de fechas ... 493
9.9 Usar scripts ... 497
9.9.1 Definir funciones JavaScript ... 498
9.9.2 Asignar funciones a controladores de eventos ... 499

9.9.3 Archivos JavaScript externos ... 500
9.10 Importación HTML ... 502
9.10.1 Crear archivo SPS a partir de un archivo HTML ... 502
9.10.2 Crear el esquema y el diseño SPS ... 504
9.10.3 Crear elementos/atributos a partir de tablas y listas ... 506
9.10.4 Generar resultados ... 508
9.11 Interfaces ASPX para aplicaciones web ... 509
9.11.1 Ejemplo: host local en Windows 7 ... 510
9.12 Archivo PXF: contenedor para el archivo SPS y archivos relacionados ... 512
9.12.1 Crear un archivo PXF ... 512
9.12.2 Editar un archivo PXF ... 515
9.12.3 Implementar un archivo PXF ... 516

10 Archivos SPS y bases de datos 518

10.1 Bases de datos en StyleVision ... 521
10.2 Conectarse a un origen de datos ... 523
10.2.1 Iniciar el asistente para la conexión de base de datos ... 524
10.2.2 Resumen de controladores de base de datos ... 526
10.2.3 Conexiones ADO ... 529
10.2.4 Conexiones ADO.NET ... 534
10.2.5 Conexiones ODBC ... 541
10.2.6 Conexiones JDBC ... 544
10.2.7 Conexiones PostgreSQL ... 549
10.2.8 Conexiones SQLite ... 550
10.2.9 Usar una conexión a través de recursos globales ... 552
10.2.10 Ejemplos de conexión a bases de datos ... 553
10.3 Seleccionar datos de la BD ... 609
10.3.1 Bases de datos no XML ... 609
10.3.2 Bases de datos XML ... 617
10.4 Esquema XML y archivos XML de la BD ... 620
10.5 Filtros de BD: filtrar datos de la BD ... 623
10.6 Características de diseño para BD ... 628
10.7 Generar archivos de salida ... 632
10.8 Consulta de bases de datos ... 633

10.8.1 Orígenes de datos ... 634
10.8.2 Explorador: visualizar objetos de la BD ... 637
10.8.3 Panel de consulta: descripción y características ... 642
10.8.4 Panel de consulta: trabajar con consultas ... 644
10.8.5 Resultados y mensajes ... 645

11 Vista Authentic 648

11.1 Interfaz de la vista Authentic ... 649
11.1.1 La interfaz gráfica del usuario (IGU) ... 649
11.1.2 Iconos de la barra de herramientas de la vista Authentic ... 650
11.1.3 Ventana principal de la vista Authentic ... 652
11.1.4 Ayudantes de entrada de la vista Authentic ... 654
11.1.5 Menús contextuales de la vista Authentic ... 659
11.2 Edición en la vista Authentic ... 662
11.2.1 Edición básica ... 662
11.2.2 Tablas en la vista Authentic ... 667
11.2.3 Editar una BD ... 674
11.2.4 Trabajar con fechas ... 681
11.2.5 Definir entidades ... 683
11.2.6 Imágenes en la vista Authentic ... 685
11.2.7 Teclas de acceso rápido en la vista Authentic ... 685

12 Scripting con Authentic 687

12.1 Editor de scripts ... 690
12.2 Macros ... 691
12.2.1 Macros en elementos de diseño ... 691
12.2.2 Macros en elementos de menús contextuales ... 693
12.2.3 Botones personalizados ... 694
12.3 Controladores de eventos ... 697
12.4 Opciones de scripting ... 698

13 Procesamiento automatizado 699

13.1 Interfaz de la línea de comandos ... 700

13.1.1 StyleVision ... 700
13.1.2 StyleVision Server ... 702
13.2 Trabajar con RaptorXML ... 703
13.2.1 Documentos PDF de salida ... 703
13.3 Automatización con FlowForce Server ... 705
13.4 Automatizar el procesamiento de datos ... 707

14 StyleVision en Visual Studio 708

14.1 Instalar el complemento de StyleVision ... 709
14.2 Diferencias entre StyleVision y StyleVision Visual Studio ... 710

15 StyleVision en Eclipse 711

15.1 Instalar el complemento de StyleVision en Eclipse ... 712
15.2 Puntos de entrada de StyleVision en Eclipse ... 719
15.3 Otros puntos de entrada de Stylevision en Eclipse ... 722

16 Referencia del usuario 723

16.1 Símbolos de la vista Diseño ... 724
16.2 Diálogo Editar expresión XPath ... 729
16.2.1 Modo Evaluador ... 731
16.2.2 Modo Depurador ... 735
16.2.3 Generador de expresiones XPath ... 744
16.3 Barras de herramientas ... 748
16.3.1 Formato ... 750
16.3.2 Tabla ... 751
16.3.3 Authentic ... 753
16.3.4 RichEdit ... 755
16.3.5 Insertar elementos de diseño ... 755
16.3.6 Filtro para diseños ... 758
16.3.7 Recursos globales ... 759
16.3.8 Estándar ... 759
16.4 Menú Archivo ... 761
16.4.1 Nuevo ... 761

16.4.2 Abrir, Volver a cargar, Cerrar, Cerrar todos ... 768
16.4.3 Guardar diseño, Guardar todos ... 773
16.4.4 Guardar como ... 778
16.4.5 Exportar como archivo de diseño de MobileTogether ... 782
16.4.6 Guardar datos XML de Authentic, Guardar como ... 782
16.4.7 Guardar archivos generados ... 782
16.4.8 Implementar en FlowForce ... 785
16.4.9 Diseño web ... 787
16.4.10 Propiedades ... 787
16.4.11 Vista previa de impresión, Imprimir ... 791
16.4.12 Archivos usados recientemente, Salir ... 792
16.5 Menú Edición ... 794
16.5.1 Deshacer, Rehacer, Seleccionar todo ... 794
16.5.2 Buscar, Buscar siguiente, Reemplazar ... 794
16.5.3 Parámetros de la hoja de estilos ... 799
16.5.4 Contraer o expandir el marcado ... 800
16.6 Menú Proyecto ... 802
16.6.1 Proyecto nuevo, Abrir proyecto, Volver a cargar el proyecto ... 803
16.6.2 Cerrar el proyecto, Guardar el proyecto ... 803
16.6.3 Agregar archivos / recurso global / URL al proyecto ... 804
16.6.4 Agregar archivo activo y relacionados al proyecto ... 806
16.6.5 Agregar carpeta y carpetas externas al proyecto ... 806
16.7 Menú Vista ... 809
16.7.1 Barras de herramientas y barra de estado ... 809
16.7.2 Barras laterales de diseño ... 810
16.7.3 Filtro para diseños, Zoom ... 810
16.8 Menú Insertar ... 812
16.8.1 Contenido ... 812
16.8.2 Resto de contenidos ... 813
16.8.3 RichEdit ... 813
16.8.4 Controles de formulario ... 815
16.8.5 Control de base de datos ... 816
16.8.6 Cálculo automático ... 816
16.8.7 Selector de fecha ... 818

16.8.8 Párrafo, párrafo especial ... 818
16.8.9 Código de barras ... 818
16.8.10 Imagen ... 820
16.8.11 Línea horizontal ... 822
16.8.12 Tabla ... 822
16.8.13 Numeración y viñetas ... 823
16.8.14 Marcador ... 825
16.8.15 Hipervínculo ... 826
16.8.16 Nota al pie ... 828
16.8.17 Condición y condición basada en el resultado ... 829
16.8.18 Deshabilitado ... 831
16.8.19 Plantilla ... 831
16.8.20 Plantilla definida por el usuario ... 832
16.8.21 Plantilla variable ... 833
16.8.22 Fragmento de diseño ... 834
16.8.23 Contenedor de diseño, cuadro de diseño, línea ... 834
16.8.24 Tabla de contenido ... 834
16.8.25 Documento nuevo ... 835
16.8.26 Página, columna, sección de documento ... 835
16.8.27 Elemento definido por el usuario ... 836
16.9 Menú Incluir en ... 837
16.9.1 Plantilla ... 837
16.9.2 Plantilla definida por el usuario ... 838
16.9.3 Plantilla variable ... 838
16.9.4 Párrafo, párrafo especial ... 838
16.9.5 Numeración y viñetas ... 839
16.9.6 Marcadores e hipervínculos ... 840
16.9.7 Condición, condición basada en el resultado ... 840
16.9.8 Deshabilitado ... 842
16.9.9 Marcadores TDC y niveles TDC ... 842
16.9.10 Documento nuevo ... 843
16.9.11 Elemento definido por el usuario ... 843
16.10 Menú Tabla ... 844
16.10.1 Insertar tabla, eliminar tabla ... 844

16.10.2 Agregar encabezado y pie de tabla ... 845
16.10.3 Anexar/insertar fila/columna ... 845
16.10.4 Eliminar fila/columna ... 846
16.10.5 Combinar celda hacia la izquierda, derecha, arriba, abajo ... 846
16.10.6 Dividir la celda horizontal y verticalmente ... 846
16.10.7 Ver bordes de la celda, Ver marcado de la tabla ... 847
16.10.8 Propiedades de la tabla ... 847
16.10.9 Editar tablas CALS/HTML ... 848
16.10.10 Alineación vertical del contenido de la celda ... 848
16.11 Menú Authentic ... 850
16.11.1 Editar scripts de Authentic ... 850
16.11.2 Personalizar botones de la barra de herramientas ... 851
16.11.3 Revisar referencias a macros ... 854
16.11.4 Agregar selector de fecha automáticamente ... 855
16.11.5 Agregar controles de base de datos automáticamente ... 855
16.11.6 Volver a cargar la vista Authentic, Validar el documento XML ... 856
16.11.7 Seleccionar y editar una fila nueva con datos XML ... 856
16.11.8 Definir entidades XML ... 857
16.11.9 Ver marcado ... 858
16.11.10 RichEdit ... 858
16.11.11 Comandos de fila de tabla dinámica ... 859
16.12 Menú Base de datos ... 860
16.12.1 Consultar la base de datos ... 860
16.12.2 Editar/borrar filtro de la base de datos ... 861
16.13 Menú Propiedades ... 863
16.13.1 Editar numeración y viñetas ... 863
16.13.2 Cadenas de formato de valor predefinidas ... 863
16.14 Menú Herramientas ... 866
16.14.1 Ortografía ... 866
16.14.2 Opciones de ortografía ... 867
16.14.3 Recursos globales ... 870
16.14.4 Configuración activa ... 871
16.14.5 Gestor de esquemas ... 872
16.14.6 Personalizar ... 887

16.14.7 Restaurar barras de herramientas y ventanas ... 894
16.14.8 Opciones ... 894
16.15 Menú Ventanas ... 900
16.16 Menú Ayuda ... 901
16.16.1 Contenido, Índice, Buscar ... 901
16.16.2 Activación, Formulario de pedido, Registro, Actualizaciones ... 901
16.16.3 Enlaces al sitio web de Altova ... 905

17 Referencia del programador 906

17.1 Editor de scripts ... 907
17.1.1 Crear un proyecto de scripting ... 907
17.1.2 Comandos integrados ... 920
17.2 Application API ... 931
17.2.1 Overview ... 932
17.2.2 Interfaces ... 952
17.2.3 Enumerations ... 1045
17.3 ActiveX Integration ... 1051
17.3.1 Prerequisites ... 1051
17.3.2 Adding the ActiveX Controls to the Toolbox ... 1052
17.3.3 Integration at Application Level ... 1054
17.3.4 Integration at Document Level ... 1056
17.3.5 ActiveX Integration Examples ... 1059
17.3.6 Command Reference ... 1072
17.3.7 Object Reference ... 1086

18 Anexos 1107

18.1 Información sobre motores XSLT y XQuery ... 1108
18.1.1 XSLT 1.0 ... 1108
18.1.2 XSLT 2.0 ... 1108
18.1.3 XSLT 3.0 ... 1110
18.1.4 XQuery 1.0 ... 1111
18.1.5 XQuery 3.1 ... 1113
18.2 Funciones XSTL y XPath/XQuery ... 1114

18.2.1 Funciones de extensión de Altova ... 1115
18.2.2 Funciones de extensión varias ... 1197
18.3 Tipos de datos en esquemas XML generados a partir de BD ... 1217
18.3.1 ADO ... 1217
18.3.2 MS Access ... 1218
18.3.3 MS SQL Server ... 1219
18.3.4 MySQL ... 1220
18.3.5 ODBC ... 1220
18.3.6 Oracle ... 1221
18.3.7 Sybase ... 1222
18.4 Datos técnicos ... 1224
18.4.1 Requisitos de SO y memoria ... 1224
18.4.2 Motores XSLT y XQuery de Altova ... 1224
18.4.3 Compatibilidad con Unicode ... 1225
18.4.4 Uso de Internet ... 1225
18.5 Información sobre licencias ... 1227
18.5.1 Distribución electrónica de software ... 1227
18.5.2 Activación del software y medición de licencias ... 1228
18.5.3 Contrato de licencia para el usuario final ... 1229

Índice **1230**

1 Introducción

Altova StyleVision 2023 Professional Edition es una herramienta para diseñar y editar gráficamente archivos SPS (StyleVision Power Stylesheet) de forma gráfica y está disponible en versiones de 64 y 32 bits. StyleVision® 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. StyleVision y Altova MissionKit pueden integrarse con aplicaciones del conjunto de herramientas Microsoft Office (MS Access, MS Excel, MS Word), versión 2007 y superior.

Los archivos SPS (StyleVision Power Stylesheet) sirven para:

- Controlar una vista gráfica WYSIWYG de **documentos XML en la vista Authentic**, que es un editor de documentos XML disponible en estos productos de Altova: XMLSpy, StyleVision, Authentic Desktop y Authentic Browser. Con este editor puede crear con facilidad formularios electrónicos basados en documentos XML.
- Habilitar la **edición de bases de datos desde la vista Authentic** y generar informes de base de datos en formato HTML, RTF y texto.
- Generar **hojas de estilos XSLT** (compatibilidad con XSLT 1.0, XSLT 2.0 y XSLT 3.0). Las hojas de estilos XSLT se pueden usar fuera de StyleVision para transformar documentos XML en documentos HTML y RTF (formato de texto enriquecido, utilizado por aplicaciones de procesamiento de texto como MS Word).
- Generar documentos **HTML**, **RTF** y **texto desde StyleVision directamente** a partir de documentos XML. Si el archivo SPS está basado en una BD, StyleVision también puede generar un esquema XML basado en la BD y un documento de instancia XML basado en este esquema y con datos de la BD.

Última actualización: *10 October 2022*

1.1 Versión 2023: novedades

Versión 2023

- Nuevo Gestor de esquemas (hecho)
- Hemos añadido el comando Deshabilidado (hecho)
- Hemos añadido la inserción de encabezados
- Hemos ampliado la compatibilidad con Eclipse 711 a las versiones siguientes: 2022-09, 2022-06; 2022-03; 2021-12.
- Hemos ampliado la compatibilidad con Visual Studio 708 a su versión 2022 (64-bit).
- También hemos ampliado la compatibilidad con bases de datos 518 a: PostGreSQL 14.
- Hemos ampliado la compatibilidad FOP a la versión 2.7.

Versión 2022 Release 2

- Hemos ampliado la compatibilidad con Eclipse 711 a las versiones siguientes: 2021-12, 2021-09; 2021-06; 2021-03.
- Hemos ampliado la compatibilidad con Visual Studio 708 a su versión 2022 (64 bits).
- También hemos ampliado la compatibilidad con bases de datos 518 a: PostGreSQL 14.
- Hemos ampliado la compatibilidad FOP a la versión 2.7.

Versión 2022

- Hemos introducido un formato de salida nuevo. Ahora puede transformar diseños de StyleVision en salida de texto. Puede generar una vista previa de ese resultado y generarlo con XSLT.
- Compatibilidad con más bases de datos 518
- Hemos ampliado la compatibilidad con Eclipse 711 a las versiones: 2021-09, 2021-06; 2021-03; 2021-12.

1.1.1 Versión 2020

Versión 2020 Release 2

Estas son las características nuevas y actualizadas de la **versión 2020r2** de StyleVision:

- Mejoras en el Depurador XPath 735.
- Compatibilidad con más bases de datos 518.
- La compatibilidad con Eclipse 711 se ha ampliado a Eclipse 4.13 y 4.14.

Versión 2020

Estas son las características nuevas y actualizadas de la **versión 2020** de StyleVision:

- Ahora StyleVision cuenta con un depurador de expresiones XPath 735 al cuadro de diálogo Editar expresión XPath 729. Esto permite probar y depurar expresiones XPath con respecto a archivos XML directamente en el cuadro de diálogo.

© 2016-2022 Altova GmbH

- Hay una nueva versión simplificada y optimizada del cuadro de diálogo Editar expresión XPath 729.
- La compatibilidad con Visual Studio 708 se ha ampliado a Visual Studio 2019.
- La compatibilidad con Eclipse 711 se ha ampliado a Eclipse 4.11 y 4.12.

1.1.2 Versión 2019

Versión 2019 Release 3

Estas son las características nuevas y actualizadas de la **versión 2019r3** de StyleVision:

- El cuadro de diálogo "Opciones" 894 (**Herramientas | Opciones**) ofrece la opción de indicar la ruta de acceso a un equipo virtual Java; esta opción tiene preferencia frente a la ruta detectada automáticamente.
- Compatibilidad con Windows Server 2019.
- La compatibilidad con Eclipse 711 se ha ampliado a Eclipse 4.9 y 4.10.

Versión 2019

Estas son las características nuevas y actualizadas de la **versión 2019** de StyleVision:

- Mejoras en la velocidad del motor XSLT
- Mejoras en la conformidad con las especificaciones del motor XSLT
- StyleVision ahora también es compatible con las 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 el complemento Eclipse 4.8.

1.1.3 Versión 2018

Versión 2018 Release 2

Estas son las características nuevas y actualizadas de la **versión 2018r2** de StyleVision:

- Posibilidad de exportar diseños con texto enriquecido como diseños de Altova MobileTogether 782
- Compatibilidad con las bases de datos Teradata 16 y MariaDB 10
- Configuración detallada del proxy de red

Versión 2018

Estas son las características nuevas y actualizadas de la **versión 2018** de StyleVision:

- Considerables mejoras en la velocidad del motor XSLT.
- Mejoras en la conformidad del motor XSLT con la especificación XSLT.
- Compatibilidad con Sybase 16, PostgreSQL 9.6, MySQL 5.7.
- Integración con Eclipse 4.7.

1.1.4 Versión 2017

Versión 2017 Release 3

Estas son las características nuevas y actualizadas de la **versión 2017 Release 3** de StyleVision:

- Generación de archivos de diseño de MobileTogether a partir de archivos SPS 782
- Asignación de licencias con Altova LicenseServer 901
- Solución de errores y mejoras internas

Versión 2017

Estas son las características nuevas y actualizadas de la **versión 2017** de StyleVision:

- Ahora puede crear archivos SPS basados en un documento MS Excel 2007+ 146Solución de errores y mejoras internas

© 2016-2022 Altova GmbH

1.2 Características del producto

A continuación se enumeran las principales características de StyleVision:

Características generales del producto

- Las ediciones Enterprise y Professional Edition están disponibles como aplicaciones de 32 y 64 bits.
- Las funciones de StyleVision se pueden invocar desde StyleVision Server.
- Las funciones de StyleVision se pueden integrar en aplicaciones externas 1051 y también en los entornos Visual Studio 708 y Eclipse 711.

Fuentes de datos

Los diseños SPS se pueden basar en esquemas XML y DTD, pero también pueden utilizar otros archivos fuente como archivos XML y CSS.

- Puede usar recursos globales de Altova 444 para encontrar archivos fuente como esquemas, documentos XML y archivos CSS. Gracias al mecanismo de recursos globales podrá completar las fases de desarrollo y pruebas con mayor rapidez y eficacia porque los recursos globales permiten alternar el origen de datos y usar características de otras herramientas de Altova desde StyleVision.
- Puede convertir documentos HTML en XML 502.

Interfaz

- Puede abrir varios diseños SPS 35 a la vez. Cada diseño SPS se abre en una pestaña distinta.
- Puede personalizar la presentación del documento SPS con filtros de plantillas 758, que permiten ocultar las plantillas que no están en edición en ese momento.
- Ocultar marcado en la vista Diseño 37: puede ocultar y contraer las etiquetas de marcado en la vista **Diseño** para despejar la vista y trabajar mejor. Mientras diseña el documento SPS, puede abrir la vista **Authentic** 38, las vistas de resultados 39 y las hojas de estilos haciendo clic en sus respectivas pestañas para ver una vista previa inmediata de los resultados/del código XSLT y probar las características del diseño en la vista **Authentic**.
- Cuando se asocia un SPS a un archivo XML de origen 223 o a una BD de origen 228, el documento de origen se puede editar directamente en la vista **Authentic** de StyleVision.

Bases de datos

- Puede ver informes de BD 518 en StyleVision o guardarlos en archivos HTML y RTF.
- StyleVision es compatible con bases de datos IBM DB2 523, que contienen columnas XML.
- StyleVision incluye una función para consultar bases de datos 633.

Documentos de salida

- StyleVision es compatible con XSLT 1.0, 2.0 y 3.0 115.
- En las ediciones Enterprise y Professional Edition puede generar archivos en varios formatos de salida 121 (HTML, RTF y texto en la edición Professional y HTML, RTF, texto, PDF y Word 2007+ en la edición Enterprise) a partir de un solo diseño SPS.

- Puede definir condiciones en los componentes del diseño SPS para procesarlos de forma diferente dependiendo del formato de salida 307. Gracias a esta flexibilidad puede estructurar los formatos de salida teniendo en cuenta las características particulares del formato.
- Puede generar y guardar 121 los archivos XSLT y los archivos de salida 121 desde la interfaz gráfica de StyleVision directamente o llamando a StyleVision desde la StyleVision Server.
- El formato PXF 512 es un formato de archivo especial desarrollado por Altova que permite guardar un archivo SPS junto con todos sus archivos de datos y archivos de origen relacionados. Esto permite transportar proyectos SPS enteros en lugar de tan solo el archivo SPS.
- Interfaz ASPX para aplicaciones web 509: gracias a esta función las páginas web HTML se pueden actualizar con gran rapidez. StyleVision genera todos los archivos necesarios para la aplicación ASPX a partir del diseño SPS. Cuando la página web (un archivo `.aspx`) se actualiza, los datos de origen (incluidos los cambios) se transforman de forma dinámica en la página web mediante XSLT.

Características de diseño de los archivos SPS

- El archivo SPS puede incluir texto estático 135 (el que se introduce en el archivo SPS directamente) y texto dinámico 135 (el que se selecciona en el documento de origen 221).
- El contenido dinámico 135 se inserta en el diseño arrastrando nodos desde el esquema fuente 221 hasta el panel de diseño. También puede insertar primero los elementos de diseño (como párrafos, listas, imágenes, etc.) y después asignarles un nodo XML de la estructura del esquema.
- Puede insertar contenido dinámico 135 como texto o en forma de dispositivo de entrada de datos 190 (por ejemplo, un campo de entrada 191 o un cuadro combinado 194). Si se inserta como dispositivo de entrada de datos, como por ejemplo un cuadro combinado 194, StyleVision le ofrece muchas opciones más. Por ejemplo, puede diseñarlo para que el usuario de la vista **Authentic** seleccione el valor del nodo en una lista de enumeraciones.
- La estructura del diseño 219 se especifica y se controla en una sola plantilla principal 29. Esta estructura se puede modificar con plantillas opcionales para los diferentes elementos del diseño (conocidas como plantillas globales 265 porque se pueden aplicar a cada tipo de elemento de forma global).
- También puede crear plantillas globales 266 para cada tipo de datos, permitiendo un procesamiento basado en tipos.
- Múltiples documentos de salida 284: puede diseñar el SPS para que los resultados se dividan en varios documentos. En el diseño puede crear plantillas de documento nuevo y colocar contenido dentro de estas plantillas. Cada plantilla de documento nuevo genera un documento de salida diferente.
- Plantillas definidas por el usuario 148: puede generar una plantilla para una secuencia de elementos con ayuda de una expresión XPath. Estos elementos pueden ser valores atómicos o nodos. Una expresión XPath permite seleccionar nodos con gran precisión gracias al uso de condiciones y filtros.
- Elementos definidos por el usuario 151: esta característica permite presentar elementos de cada lenguaje (HTML, XSLT y XSL-FO) e insertarlos en cualquier posición del diseño.
- Bloques de texto XML definidos por el usuario 151: puede insertar bloques de texto XML en cualquier posición del diseño y estos bloques se crearán en dicha posición en la hoja de estilos XSLT generada.
- Fragmentos de diseño 278: permiten crear módulos y reutilizar plantillas dentro del SPS y en otros archivos SPS (*archivos SPS modulares* 251) y se utilizan como las funciones.
- Módulos SPS 251: puede añadir módulos a otros módulos SPS, para poner a disposición de otros módulos los objetos definidos en otro módulo. Esto permite la reutilización de objetos en varios archivos SPS y facilita el mantenimiento de los diseños.
- Plantillas XSLT 282: puede importar archivos XSLT en las hojas de estilos generadas. Si un nodo del documento XML de instancia se empareja con una plantilla del archivo XSLT importado y no hay otra plantilla que tenga prioridad sobre la plantilla importada, entonces se usa la plantilla importada. Además puede llamar a las plantillas con nombre del archivo XSLT importado desde el diseño.

 © 2016-2022 Altova GmbH

- Archivo nuevo a partir de un archivo XSLT 471: puede crear un archivo SPS a partir de una hoja de estilos XSLT-HTML o XSLT-FO. La estructura y el estilo del XSLT se recrea en el SPS. Después puede modificar los componentes del SPS, agregar contenido y aplicar formato.
- Funciones XPath definidas por el usuario 475: el usuario puede definir funciones que se pueden usar en cualquier posición del documento que admita el uso de funciones XPath.
- Secciones de documento 420: puede dividir los documentos en varias secciones y cada sección puede tener características propias (como propiedades de diseño de página, por ejemplo). Esto permite representar de forma distinta cada parte del documento.
- Contenedores de diseño 204: un contenedor de diseño es un bloque donde puede colocar elementos de diseño con una posición absoluta.
- Imágenes modelo 207: puede usar una imagen de un formulario dentro de un contenedor de diseño como imagen modelo para el diseño.
- Una de las principales características de los documentos XML es su estructura de datos periódica. Por ejemplo, un departamento de una empresa tiene varios empleados. Los datos de cada empleados deben almacenarse en una estructura de datos que se repetirá para cada empleado. En el archivo SPS, el procesamiento de dichas estructuras de datos 135 se define una sola vez y se aplica a los nodos de uno en uno.
- En diseños SPS para XSLT 2.0 y XSLT 3.0 puede insertar varias tablas de contenido 337.
- También puede insertar estructuras de datos periódicas en forma de tablas dinámicas 155. Esto permite recorrer los datos en formato de tabla estructurado. Cada bucle da lugar a una fila (o una columna si es necesario) en la tabla.
- Un elemento periódico (o repetido) se puede ordenar según varios criterios de ordenación 321 y el conjunto de elementos ordenados se envía al documento de salida (HTML y RTF).
- Variables 332: puede declarar una variable en una plantilla y la variable puede tomar el valor especificado con una expresión XPath. En versiones anteriores el valor de una variable se limitaba a la selección del nodo en el que se creaba. A partir de la versión 2010 se puede usar cualquier expresión XPath para especificar el valor de la variable.
- Puede agrupar 312 nodos según los datos que tienen en común (por ejemplo, el valor que tienen en común para un atributo determinado) y según su posición.
- Gracias a las plantillas condicionales 303 se pueden procesar las plantillas de un grupo si se cumplen determinadas condiciones del documento XML o del entorno del sistema. Esto permite un procesamiento condicional dependiendo de los datos del documento de origen o de la información que tiene el diseñador del archivo SPS a la hora de crear el archivo (por ejemplo, la fecha de procesamiento). Las condiciones disponibles son aquellas que se pueden probar con expresiones XPath.
- Cálculos automáticos 293: permiten manipular los datos de los documentos de origen y ver el resultado. Esto es práctico para realizar cálculos con números (por ejemplo, sumar los importes de una factura), para manipular cadenas de texto (por ejemplo, cambiar los guiones por barras diagonales), para generar contenido, etc. Las manipulaciones de datos disponibles son aquellas que se pueden llevar a cabo mediante expresiones XPath. En las expresiones XPath de los cálculos automáticos también puede usar funciones Java y .NET nativas.
- Cuando edite datos en la vista **Authentic**, el resultado de los cálculos automáticos 293 se puede enviar a un nodo del documento de origen. Este procedimiento se denomina actualización del nodo XML 296 (con el valor del cálculo automático).
- Validación adicional 467: permite validar los nodos del documento XML no solo con el esquema, sino con la expresión XPath definida para cada nodo. De este modo el usuario de la vista **Authentic** puede saber si los datos que introduce son válidos. Para informar al usuario se utiliza un mensaje de error personalizable para cada nodo.
- Códigos de barras 200: este componente de diseño permite insertar códigos de barras en el diseño. Las imágenes de código de barras se generan instantáneamente y se colocan en los documentos de salida.

- Presencia condicional 310: ciertos componentes de diseño tienen una propiedad de presencia condicional. El componente de diseño condicional solamente se crea en el documento de salida si se cumple la condición especificada.
- Imágenes: 183 puede insertar imágenes en el diseño y el URI de la imagen puede ser estático (introducido en el SPS directamente), dinámico (tomado de un nodo del documento de origen) o parcialmente estático, parcialmente dinámico.
- Imágenes de datos inline 155: puede generar imágenes a partir de texto codificado base-16 y base-64 del documento XML. Por tanto, puede almacenar imágenes en el documento XML como texto directamente. Los archivos SPS pueden decodificar este texto y representar la imagen.
- Puede crear dos tipos de listas 178 en los diseños SPS: listas estáticas 178, donde cada elemento de la lista se define en el archivo SPS directamente, y listas dinámicas 180, donde los elementos de la lista se corresponden con nodos del documento de origen.
- Puede insertar hipervínculos dinámicos y estáticos 364 en el diseño. El URI de destino puede ser estático (si se introduce en el SPS directamente), dinámico (si se toma de un nodo del documento de origen) o parcialmente estático, parcialmente dinámico.
- Puede insertar marcadores 364 estáticos y utilizarlos como anclaje y enlazar a ellos mediante hipervínculos.
- Parámetros 327: se pueden declarar de forma global para todo el SPS. Cada parámetro se declara con un nombre y un valor de cadena y se puede usar en expresiones XPath en el archivo SPS. El valor del parámetro que se declara es el valor predeterminado y se puede reemplazar con otro valor desde StyleVision Server.
- Formato de entrada 380: el contenido de los nodos de tipo de datos numéricos XML Schema puede tener el formato necesario para la vista **Authentic** y para los demás formatos de salida. Esta característica también se puede usar para aplicar formato al resultado de los cálculos automáticos 293.
- Funciones JavaScript: 497 puede usar estas funciones en el SPS para añadir funciones a la vista **Authentic** del documento y a los documentos HTML de salida.
- Scripting de Authentic 687: aporta una mayor flexibilidad y más posibilidades de interacción en la vista **Authentic**.
- Formatos HTML predefinidos 375: en la interfaz gráfica puede acceder a numerosos formatos HTML predefinidos y aplicarlos a los componentes del diseño SPS.
- Puede aplicar un gran número de propiedades de formato de diseño y texto CSS a los componentes del archivo SPS desde la ventana Estilos 398.
- Además puede definir estilos CSS para los selectores HTML a nivel global 395 y en hojas de estilos CSS externas. Estas reglas de estilo se aplicarán en la vista **Authentic** y en los documentos HTML de salida.
- Asignar estilos con expresiones XPath: 402 puede seleccionar los valores de las propiedades de estilo desde documentos XML y establecer sus valores de forma condicional.
- En el caso de formatos para medios impresos (RTF, PDF y Word 2007+) puede especificar numerosas opciones de diseño de página 419, como la orientación, los márgenes, la numeración, encabezados y pies de página, etc.

© 2016-2022 Altova GmbH

1.3 La vista Authentic en los productos Altova

La vista **Authentic** es un editor gráfico de documentos XML disponible en estos productos de Altova:

* Altova XMLSpy
* Altova Authentic Desktop Edition
* Altova Authentic Browser Edition
* Altova StyleVision

En StyleVision la vista **Authentic** puede verse en la pestaña **Formulario electrónico de Authentic** de la ventana principal 38.

1.4 ¿Qué es una hoja de estilos SPS?

Una StyleVision Power Stylesheet (o archivo SPS) es una hoja de estilos XSLT ampliada con la que puede:

- controlar la presentación y la entrada de datos de documentos XML y bases de datos en la vista Authentic 27 y
- especificar el diseño de salida de la transformación de documentos XML.

Los archivos SPS tienen la extensión de archivo `.sps`.

Diseño del archivo SPS

Los archivos SPS se crean de forma gráfica en StyleVision y se basan en un esquema, bien en una DTD o bien en un XML Schema. Si el SPS se debe usar con una BD, entonces se basa en un esquema XML generado automáticamente por StyleVision a partir de la estructura de la BD. El diseño del archivo SPS es flexible: puede incluir contenido dinámico y estático. Se denomina *contenido dinámico* 29 el conjunto de datos procedentes de documentos XML y de bases de datos. El *contenido estático* 29 es el contenido que se introduce en el diseño SPS directamente. El contenido dinámico se puede incluir en el diseño como texto simplemente o dentro de componentes como campos de entrada, cuadros combinados y tablas. Además, puede manipular el contenido dinámico (con cálculos automáticos) y presentarlo solamente si se cumplen ciertas condiciones en el documento de origen. También puede colocar partes diferentes del contenido en posiciones distintas del diseño SPS. Asimismo, el SPS puede incluir otros componentes como imágenes, hipervínculos y funciones JavaScript. El usuario puede definir el formato y la presentación de cada componente del SPS.

El archivo SPS y la vista Authentic

Si un documento XML o una BD tienen asociado un archivo SPS, entonces pueden editarse en la vista Authentic 27. La vista Authentic es una solución ideal para la distribución y edición gráfica de documentos XML y bases de datos. En su interfaz gráfica pueden trabajar varios usuarios para editar el documento XML o la BD. En StyleVision puede ver una vista previa y probar el diseño SPS mientras lo diseña en la pestaña de la vista Authentic. Para más información sobre el funcionamiento de archivos SPS en la vista Authentic, consulte el apartado Archivos SPS y la vista Authentic 118.

El archivo SPS y las hojas de estilos XSLT

Cuando termine de diseñar el archivo SPS, puede generar hojas de estilos XSLT basadas en el diseño recién creado. StyleVision es compatible con XSLT 1.0, XSLT 2.0 y XSLT 3.0 y puede usar un único archivo SPS para generar hojas de estilos XSLT en HTML, RTF, XSL-FO, texto y Word 2007 y versiones posteriores (*XSL-FO, texto y Word 2007 y versiones superiores solamente en la edición Enterprise; RTF y texto en las ediciones Enterprise y Professional; en la edición Basic solo se puede generar HTML*). Las hojas de estilos XSLT generadas se pueden usar en transformaciones externas para transformar documentos XML basados en el mismo esquema utilizado en el SPS con el que se generó la hoja de estilos XSLT. Para más información consulte el apartado Archivos generados 121.

El archivo SPS y los documentos de salida

También puede usar StyleVision para generar documentos de salida directamente. Las pestañas de las vistas de resultados 39 muestran el resultado del documento SPS activo. Estos resultados se pueden exportar a un archivo desde la interfaz gráfica con el comando **Archivo | Guardar archivos generados** 782 o con StyleVision Server.

© 2016-2022 Altova GmbH

La vista Authentic 27 de los productos de Altova

La vista **Authentic** es un editor gráfico de documentos XML disponible en estos productos de Altova:

* Altova XMLSpy
* Altova Authentic Desktop Edition
* Altova Authentic Browser Edition
* Altova StyleVision

1.5 Configurar StyleVision

Altova StyleVision es compatible con estas plataformas: Windows 7 SP1 con actualización de la plataforma, Windows 8, Windows 10, Windows 11. Después de descargar StyleVision del sitio web de Altova, haga doble clic en el archivo ejecutable (`.exe`) para ejecutar el programa de instalación. El programa de instalación instalará StyleVision en el directorio elegido. Los motores XSLT (1.0 y 2.0) de Altova están integrados en StyleVision y se utilizan para todas las transformaciones internas. Por tanto, no necesita instalar otro motor XSLT. El programa de instalación de StyleVision instala también el procesador FOP de Apache (que sirve para generar PDF a partir de XSL-FO) en el directorio `C:\ProgramData\Altova\SharedBetweenVersions`.

No obstante, es necesario tener instalados estos componentes:

- Internet Explorer 5.5 o superior, para las vistas **Authentic**, **HTML** y **Diseño**. Internet Explorer 6.0 (o superior) ofrece una mayor compatibilidad con XML.
- Microsoft Word 2000+, para la vista previa en formato RTF. Recomendamos usar Microsoft Word 2003+. Para poder copiar y pegar contenido de documentos Word 143 (y contenido que se pueda pegar en documentos Word, como tablas Excel y contenido de páginas HTML) es necesario Word 2007+.
- Microsoft Word 2007+ o Microsoft Word 2007+ Viewer para la vista previa de Word 2007+. Puede usar Microsoft Word 2003 con paquete de compatibilidad para la vista previa de documentos Word 2007+, pero a veces no genera la vista previa correctamente.

Nota: en esta documentación se usa **Word 2007+** para hacer referencia a la versión 2007 o superior de Microsoft Word.

Nota: si hay un problema con la vista previa incrustada, StyleVision intenta abrir el documento de vista previa en una aplicación externa (como MS Word o Adobe Reader) y emite un mensaje de error sobre la vista previa incrustada. Si el documento de vista previa se abre en una aplicación externa, será necesario cerrar la aplicación externa antes de volver a generar el documento de salida temporal. De lo contrario, aparece un mensaje diciendo que el archivo está siendo utilizado por otro proceso. Antes de cerrar el diseño SPS debe cerrar también la aplicación externa. De lo contrario, StyleVision no podrá cerrar el documento de salida temporal dado que la aplicación externa bloquea el documento.

 © 2016-2022 Altova GmbH

1.6 Terminología

Aquí se definen los términos utilizados en la interfaz gráfica de StyleVision y en la presente documentación. Los términos están agrupados por tema y, dentro de cada tema, por orden alfabético:

- Términos relacionados con productos de Altova
- Términos y conceptos XML generales
- Términos XSLT y XPath
- Términos específicos de StyleVision

Términos relacionados con productos de Altova

Recurso global alias de un conjunto de archivos, carpetas o bases de datos. Cada alias tiene un conjunto de configuraciones y cada configuración se asigna a un recurso. Si utiliza un recurso global en StyleVision, puede cambiar de recurso cambiando la configuración activa.

SPS abreviatura de StyleVision Power Stylesheet utilizada en la presente documentación para designar el documento de diseño creado en StyleVision y guardado con la extensión de archivo `.sps`. Para más información consulte el apartado ¿Qué es una hoja de estilos SPS? 26.

Vista Authentic editor de documentos XML disponible en estos productos de Altova: Altova XMLSpy; Altova StyleVision; Altova Authentic Desktop; Altova Authentic Browser. Para más información consulte el sitio web de Altova.

Términos XML generales

esquema término general para designar cualquier tipo de esquema. StyleVision es compatible con *esquemas XML* y esquemas DTD.

esquema XML esquema que cumple con la especificación XML Schema del W3C. Los esquemas XML son un tipo concreto de *esquema*.

URI y URL en esta documentación se utiliza únicamente el término general *URI*, incluso cuando el identificador solamente tiene el aspecto `locator` e incluso para los identificadores que usan el modelo `http`.

Términos XSLT y XPath

Algunos términos de XSLT 1.0 y XPath 1.0 ahora se llaman de otra manera en XSLT 2.0 y XPath 2.0. Por ejemplo, el antiguo *nodo raíz* de XPath 1.0 ahora se llama *nodo de documento* en XPath 2.0. En la presente documentación se utiliza la terminología de XSLT 2.0 y XPath 2.0.

elemento de contexto / nodo de contexto el *elemento de contexto* es el elemento (nodo o valor de cadena) en relación al cual se evalúa una expresión. Un *nodo de contexto* es un *elemento de contexto* que es un nodo. El *elemento de contexto* puede cambiar dentro de una expresión o dentro de una expresión de filtro (predicado).

elemento de documento en un documento XML con formato correcto, el elemento más exterior se denomina *elemento de documento*. Es un elemento secundario del *nodo de documento*. En un documento XML con formato correcto solamente hay un *elemento de documento*. En la interfaz gráfica de StyleVision, el elemento de documento se llama *elemento raíz*.

nodo actual nodo que está actualmente en procesamiento. El nodo actual es el mismo que el *nodo de contexto* de las expresiones que no tienen subexpresiones. Pero si hay subexpresiones, el *nodo de contexto* puede cambiar. No olvide que la función `current()` es una función XSLT y no una función XPath.

nodo de documento el *nodo de documento* representa y contiene todo el documento. Se trata del nodo raíz de la representación en forma de árbol del documento y en las expresiones XPath se representa como "/". En la ventana Estructura del esquema de StyleVision se presenta como **/** `Elementos raíz`.

XPath absoluta expresión de ruta de acceso que empieza por el nodo raíz del árbol que contiene el *nodo de contexto*. Cuando se escriben expresiones de ruta en los cuadros de diálogo de StyleVision, la expresión puede ser absoluta (si marca la casilla *XPath absoluta*) o relativa al *nodo de contexto* (si no marca la casilla).

Términos específicos de StyleVision

Estos términos designan mecanismos, conceptos y componentes propios de StyleVision.

archivo de plantilla XML los archivos de plantilla XML se asignan a un SPS en StyleVision (ediciones Enterprise y Professional). Se trata de un *archivo XML* que ofrece los datos de partida de un documento XML nuevo creado con un SPS dado cuando dicho SPS se abre en la vista **Authentic**. El archivo de plantilla XML debe cumplir las normas del *esquema* en el que está basado el archivo SPS.

archivo XML/XBRL de trabajo archivo de datos XML que se asigna a un SPS en StyleVision para poder obtener una vista previa de la vista **Authentic** y los resultados del documento XML. Sin un archivo XML/XBRL de trabajo, el archivo SPS no puede procesar datos XML dinámicos. Si el SPS está basado en un *esquema* que tiene varios elementos globales, puede ser difícil determinar qué elemento global es el *elemento de documento*. Para evitar esta ambigüedad puede asignar un archivo XML/XBRL de trabajo al SPS (porque un documento XML válido tiene por definición un solo *elemento de documento*). Recuerde que XBRL solamente es compatible con la edición Enterprise Edition.

 © 2016-2022 Altova GmbH

bloques de texto XML definidos por el usuario bloques de texto XML que se pueden insertar en cualquier posición del diseño.

componente SPS puede tratarse de (i) un nodo del esquema (por ejemplo, un nodo elemento), (ii) un componente SPS estático como un cálculo automático o una cadena de texto o (iii) un formato predefinido (que se representa mediante etiquetas de inicio y cierre en el SPS).

contenedor de diseño bloque de diseño en donde se pueden colocar elementos de diseño con posición absoluta. Si desea basar un diseño en un formulario, puede crearlo como contenedor de diseño para poder crear los elementos de diseño del formulario con posiciones absolutas. Otra opción es crear un diseño de movimiento libre y colocar contenedores dentro del diseño.

documento XML este término tiene dos acepciones: (i) puede referirse a un documento XML determinado o (ii) puede referirse a cualquier origen de datos XML, incluidos orígenes de datos de BD (a partir de los cuales se pueden crear documentos XML para usarlos con un SPS). La acepción utilizada en cada caso dependerá del contexto.

elemento definido por el usuario elemento que no es ni un nodo del árbol del esquema, ni un elemento predefinido ni un elemento de diseño, sino un elemento definido por el usuario. Puede definirse con atributos.

elementos dinámicos elementos procedentes de fuentes de datos XML. Los elementos dinámicos pueden ser texto, tablas, listas, imágenes e hipervínculos (si los URI son dinámicos).

elementos estáticos elementos procedentes del archivo SPS que no se encuentran en los orígenes de datos XML. Los elementos estáticos pueden ser texto, etiquetas, listas, imágenes, hipervínculos y marcadores (cuando los URI son estáticos).

elemento global elemento de la lista "Elementos globales" de la ventana Estructura del esquema. Si se trata de un *esquema XML*, todos los elementos definidos como elementos globales se enumeran en la lista "Elementos globales". Si se trata de una DTD, todos los elementos que son elementos globales se enumeran en la lista "Elementos globales". Solamente se pueden definir plantillas globales para elementos globales.

esquema principal en el SPS se define como esquema principal uno de los esquemas fuente asignados. El *nodo de documento* del *archivo XML de trabajo* asociado con el esquema principal es el punto de partida para la *plantilla principal*.

imagen modelo imagen utilizada como imagen de fondo en un contenedor de diseño que suele ser una copia escaneada de un formulario. El diseño SPS puede basarse en esta imagen escaneada para recrear el diseño del formulario.

plantilla término general que designa un conjunto de instrucciones necesarias para procesar un nodo o un grupo de nodos.

plantilla definida por el usuario plantilla creada para una secuencia especificada con una expresión XPath.

plantilla global las plantillas globales se pueden definir para los *elementos globales*. Una vez definida, la plantilla global se puede usar para el elemento correspondiente en todas las posiciones donde aparezca el elemento. Otra opción es definir el procesamiento del elemento global en una *plantilla local*.

plantilla local plantilla que define cómo se procesa un elemento (global o no) dentro de la *plantilla principal*. La plantilla local afecta solamente a dicha instancia concreta del elemento en la *plantilla principal*. En lugar de una plantilla local, también puede aplicar una *plantilla global* a la instancia del elemento.

plantilla principal plantilla de punto de entrada principal. En StyleVision esta plantilla coincide con el elemento de documento y es la primera plantilla que evalúa el procesador XSLT. En la ventana Estructura del esquema aparece como secundario del *nodo de documento*. La plantilla principal define la estructura básica del documento de salida y define cómo se deben procesar los documentos de entrada. Puede incluir *plantillas locales* y hacer referencia a *plantillas globales*.

resultado resultado que se obtiene al procesar un documento XML con una hoja de estilos XSLT. StyleVision puede generar archivos de resultados en formato HTML y RTF. La vista **Authentic** no se considera un resultado. Las hojas de estilos XSLT generadas con StyleVision tampoco se consideran resultados.

 © 2016-2022 Altova GmbH

1.7 Información preliminar

La presente documentación es el manual del usuario incluido con StyleVision. Está disponible en el sistema de ayuda en pantalla de StyleVision y en el sitio web de Altova, tanto en formato web como en PDF.

El manual del usuario se divide en varias secciones:

- Introducción: explica qué es un archivo SPS y presenta las características y los conceptos principales de StyleVision.
- Interfaz del usuario 35: describe brevemente la interfaz gráfica del usuario de StyleVision.
- Tutorial de introducción rápida 67: ofrece un ejercicio práctico para familiarizarse con las funciones de StyleVision.
- Resumen de características 112: describe de forma general cómo usar las características principales de StyleVision. Por ejemplo, cómo usar esquemas fuente para crear un SPS, el proceso general de diseño, la implementación en la vista Authentic y el uso de proyectos.
- Archivo SPS: contenido 134: explica cómo crear y editar componentes estáticos (generados por la hoja de estilos) y dinámicos (generados por el documento XML) en el diseño SPS.
- Archivo SPS: estructura 219: explica cómo estructurar un archivo SPS, cómo crear módulos y cómo trabajar con plantillas.
- Archivo SPS: características avanzadas 292: describe las funciones avanzadas de diseño como la generación automática de cálculos, condiciones, agrupaciones y criterios de ordenación definidos por el usuario, así como la creación de tablas de contenido y referencias cruzadas en el documento de salida.
- Archivo SPS: presentación 374: explica cómo colocar los componentes SPS y cómo darles formato.
- Archivo SPS: otras funciones de edición 443: describe funciones adicionales para optimizar los diseños SPS. Entre ellas destacan los recursos globales, con los que podrá aprovechar recursos creados en otros productos de Altova, la validación adicional, los scripts, las variables y los parámetros.
- Archivos SPS y bases de datos 518: explica cómo utilizar bases de datos en los diseños SPS.
- Vista Authentic 649: describe cómo editar documentos XML en la vista Authentic. La IGU de StyleVision incluye una vista Authentic donde puede probar inmediatamente sus formularios de Authentic.
- Procesamiento automatizado 699: explica cómo puede automatizar la generación de archivos de salida.
- Integración de StyleVision 1051: contiene documentación para integrar StyleVision con otras aplicaciones e información sobre cómo usar StyleVision en Visual Studio 708 y Eclipse 711.
- Referencia del usuario 723: describe uno a uno todos los menús, símbolos y comandos de StyleVision.
- Anexos 1107 con información sobre los motores XSLT de Altova y sobre la conversión de tipos de datos de BD en tipos de datos XML Schema, datos técnicos sobre la herramienta e información sobre las licencias.

¿Cómo usar esta documentación?

Sugerimos empezar por las secciones Introducción, Interfaz del usuario 35 y Resumen de características 112 para familiarizarse con las características y funciones de StyleVision. Después puede hacer el Tutorial 67 para aprender a crear un archivo SPS. Las secciones dedicadas al archivo SPS (Archivo SPS: contenido 134, Archivo SPS: estructura 219, Archivo SPS: características avanzadas 292, Archivo SPS: presentación 374, Archivo SPS: otras funciones 443, Archivo SPS y bases de datos 518) describen en detalle cómo utilizar las funciones de StyleVision. En la sección Referencia del usuario 723 encontrará información sobre los iconos de las barras de herramientas, símbolos de los diseños y comandos de menú. La Referencia del usuario dedica un apartado a cada menú de StyleVision. En las secciones Vista Authentic 649 encontrará información sobre las funciones de edición de esta vista.

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

Ayuda y soporte técnico

Si tiene problemas o alguna pregunta sobre StyleVision:

1. Consulte el archivo de Ayuda 17 (esta documentación). El archivo de ayuda contiene una función de búsqueda y un índice de palabras clave.
2. Consulte las páginas de Preguntas frecuentes y Foro de debate del sitio web de Altova.
3. Póngase en contacto con el equipo de soporte técnico de Altova.

Abreviaturas

En esta documentación utilizamos frecuentemente estas abreviaturas:

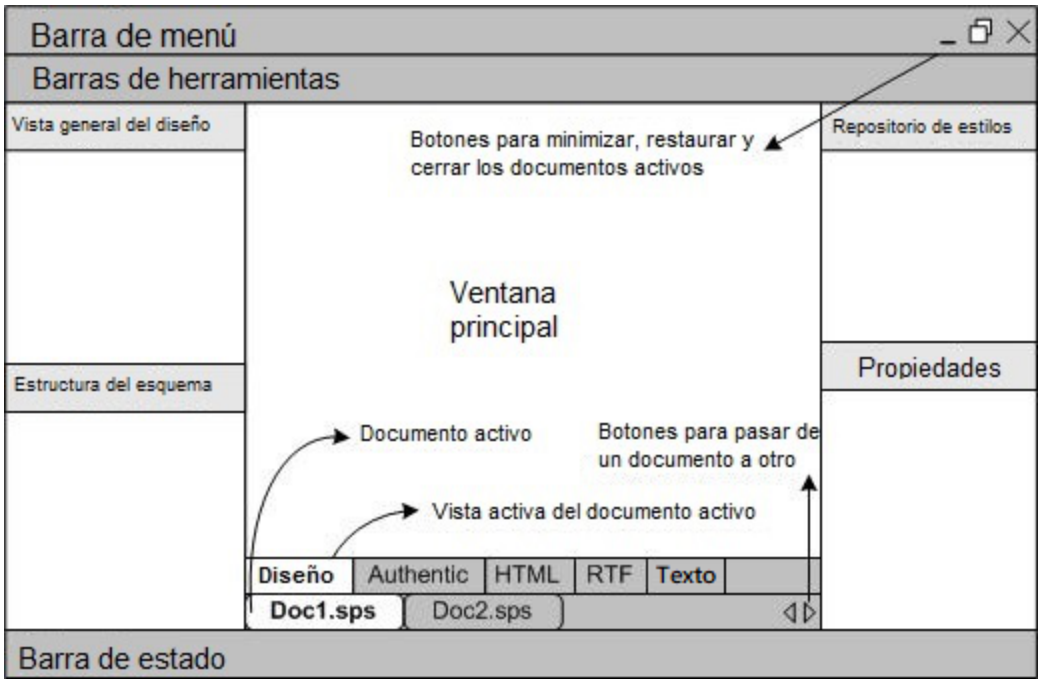
- **SPS**: StyleVision Power Stylesheet
- **BD**: base de datos
- **CSS**: hojas de estilos en cascada

© 2016-2022 Altova GmbH

2 Interfaz del usuario

La interfaz gráfica de usuario de StyleVision se divide en varias secciones:

- La **barra de menús**. Haga clic en un menú para ver todos sus comandos. Los menús y sus respectivos comandos se describen detalladamente en la sección Referencia del usuario 723. La barra de menús también incluye botones para **Minimizar**, **Restaurar** y **Cerrar** los documentos activos.
- El área de las **barras de herramientas**. Las barras de herramientas 748 y sus respectivos accesos directos de comando se describen detalladamente en la sección Referencia del usuario 723.
- Una **ventana principal** 36 dividida en pestañas. La ventana principal muestra los documentos SPS que están abiertos. En esta ventana puede editar el diseño del archivo SPS 37, editar el contenido de la vista Authentic 38 (en la pestaña **Formulario electrónico de Authentic**) y la vista previa de las hojas de estilos XSLT y los archivos de salida 39.
- Las ventanas de las **barras laterales de diseño** 41 Vista general del diseño 45, Estructura del esquema 48, Estructura del diseño 52, Repositorio de estilos 55, Estilos 58, Propiedades 59 y Proyecto 63. Estas ventanas se pueden acoplar dentro de la interfaz o pueden quedarse como ventanas flotantes.
- La **barra de estado**. Esta barra muestra información sobre el estado de la aplicación. Si usa la versión de 64 bits de StyleVision, en la barra de estado aparece el sufijo `(x64)` después del nombre de la aplicación. Para la versión de 32 bits no se incluye ningún prefijo.

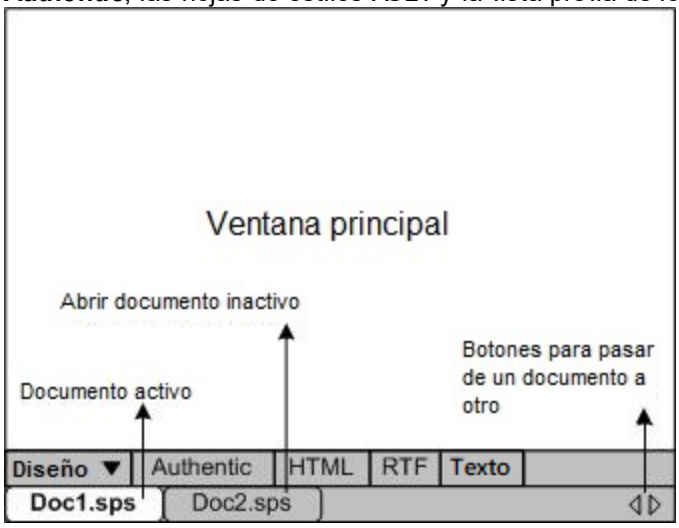


La ventana principal 36 y las barras laterales de diseño 41 se describen con más detalle en los diferentes apartados de esta sección.

Nota: para mover la barra de menús y las barras de herramientas a otra posición, haga clic en sus controladores y arrastre las barras hasta su nueva posición.

2.1 Ventana principal

La ventana principal (*ver ilustración*) es la sección de la IGU donde se presenta el diseño SPS, la vista **Authentic**, las hojas de estilos XSLT y la vista previa de los documentos de salida.



Documentos SPS en la ventana principal

- En StyleVision puede abrir varios documentos SPS a la vez, pero solamente puede haber un documento activo. El nombre de los documentos que están abiertos aparecen en unas pestañas situadas en la parte inferior de la ventana principal. La pestaña resaltada es la del documento activo.
- Para pasar del documento activo a otro documento, haga clic en su pestaña. También puede usar las opciones del menú **Ventanas**.
- Si abre una gran cantidad de documentos y no todos ellos están visibles en la barra de pestañas, haga clic en los botones de desplazamiento (a la derecha de la barra de pestañas) para ver las demás pestañas que faltan en la vista.
- Para cerrar el documento activo, haga clic en el botón **Cerrar documento** de la barra de menús (en la esquina superior derecha de la ventana de la aplicación) o seleccione **Archivo | Cerrar** 768.

Vistas de los documentos

En StyleVision puede ver los documentos en varias vistas diferentes y solamente puede haber una vista activa:

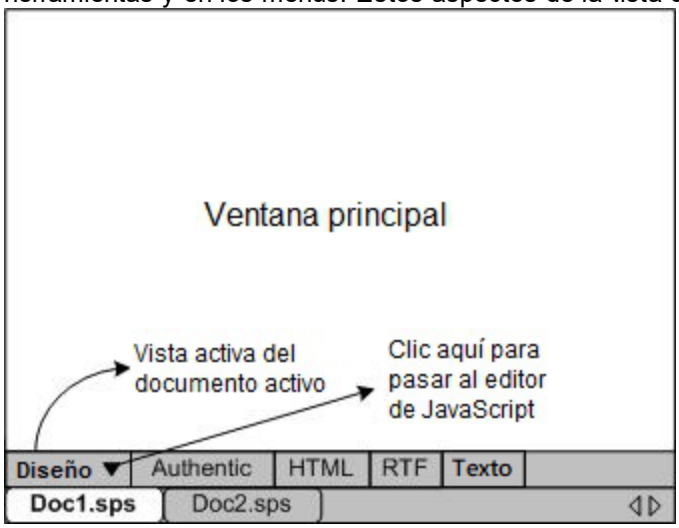
- Vista 37 **Diseño** 37: se trata de la vista donde se diseña el SPS y se editan las funciones JavaScript que se utilizarán en el SPS. Para pasar de la vista de diseño al editor JavaScript haga clic en la punta de flecha de la pestaña **Diseño** y seleccione **Diseño** o **JavaScript** según corresponda.
- Vista 38 **Authentic** 38 (pestaña **Formulario electrónico de Authentic**): permite ver inmediatamente la vista **Authentic** de un documento XML (el archivo XML de trabajo 30). El SPS se aplica dinámicamente al archivo XML de trabajo 30, lo cual permite probarlo en la vista **Authentic**.
- Vistas de 39 **resultados** 39: (pestañas **HTML**, **RTF** y **texto**). Estas pestañas ofrecen una vista previa del formato de salida correspondiente y de la hoja de estilos XSLT utilizada para generar el documento de salida. Puede cambiar entre la vista previa del documento de salida y de la hoja de estilos XSLT haciendo clic en la punta de flecha de la pestaña y seleccionando la opción pertinente.

© 2016-2022 Altova GmbH

Todas estas vistas se corresponden con una pestaña de la vista principal (en la barra de vistas). Para seleccionar una vista, haga clic en la pestaña correspondiente. La pestaña de la vista seleccionada aparece resaltada.

2.1.1 Vista Diseño

La vista **Diseño** (*ver ilustración*) es la vista donde se diseña el archivo SPS. En la vista **Diseño** puede crear el diseño del documento de salida (i) insertando contenido con ayuda de las barras laterales, el teclado y las funciones de creación y edición de contenido disponibles en los menús y en las barras de herramientas y (ii) aplicando formato al contenido con ayuda de las funciones de formato disponibles en las barras de herramientas y en los menús. Estos aspectos de la vista de diseño se explican más abajo.



En la vista **Diseño** también puede abrir el editor de JavaScript 498 o el editor de script de Authentic. En el editor de JavaScript puede crear y editar funciones de JavaScript 497 para utilizarlas después en el SPS. En el editor de scripts puede crear y editar scripts para la vista Authentic. Para cambiar al editor de JavaScript 498 o al editor de scripts de Authentic, haga clic en la punta de flecha de la pestaña **Diseño** (*ver ilustración*) y seleccione **JavaScript** o **Scripts de Authentic** en el menú desplegable que aparece. Para volver a la vista **Diseño**, haga clic otra vez en la punta de flecha y seleccione **Diseño**.

El documento SPS puede tener varias plantillas en la vista **Diseño**: la plantilla principal, plantillas globales, plantillas de diseño de página y fragmentos de diseño. Puede controlar qué tipos de plantilla aparecen en la vista **Diseño** gracias a los filtros de plantillas 758, disponibles en forma de iconos en la barra de herramientas 758. Estos filtros de presentación sirven para optimizar la vista y visualizar el diseño SPS de maneras diferentes.

Ver etiquetas de marcado

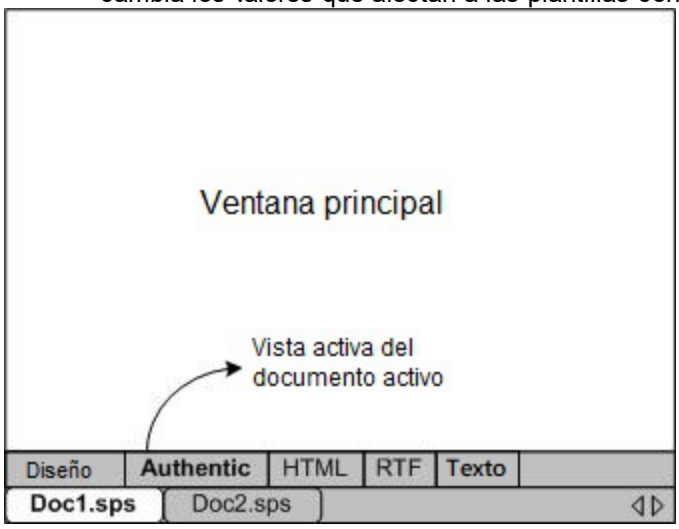
En la vista **Diseño** puede controlar la visualización de las etiquetas de marcado con los iconos de marcado que aparecen a continuación:

Si activa el icono de la izquierda, el marcado del diseño aparece en tamaño pequeño (las etiquetas aparecen sin nombre). Si hace clic en el de la derecha, el marcado del diseño aparece en tamaño grande (las etiquetas aparecen con nombre). Cuando se activa el marcado pequeño, la ruta del nodo aparece al pasar el cursor del ratón por encima del nodo.

2.1.2 Vista Authentic

En la vista **Authentic** de la ventana principal (la pestaña **Formulario electrónico de Authentic**) puede ver y editar el archivo XML de trabajo [30] tal y como aparecería en la vista **Authentic**. Esta vista permite (i) comprobar el aspecto que tendrá su documento XML para Authentic y (ii) probar la vista **Authentic** creada con el diseño SPS. Esto es muy práctico a la hora de probar las características dinámicas de la vista **Authentic**. Por ejemplo, puede probar cómo se comportaría la vista **Authentic** cuando el usuario:

- añade elementos y atributos nuevos
- añade párrafos o filas de tablas nuevos
- cambia los valores que afectan a las plantillas condicionales



La vista Authentic y el archivo XML de trabajo

Para poder ver la vista **Authentic** del diseño es necesario asignar [48] un archivo XML de trabajo [30] al documento SPS activo. Este archivo XML de trabajo debe ser válido con respecto al esquema en el que se basa el archivo SPS.

StyleVision crea un archivo XML temporal basado en el archivo XML de trabajo y es este archivo temporal el que se presenta en la pestaña de la vista **Authentic**. Los cambios realizados en la vista **Authentic** modifican el archivo XML temporal. El archivo XML de trabajo propiamente dicho no se modifica hasta que guarde los cambios de forma explícita (con el comando **Archivo | Guardar datos XML de Authentic** [782]).

Si el SPS no tiene asignado ningún archivo XML de trabajo, al hacer clic en la pestaña de la vista **Authentic**, StyleVision solicita que asigne un archivo XML de trabajo al SPS.

 © 2016-2022 Altova GmbH

Restricciones de la vista Authentic

La vista **Authentic** de la ventana principal es similar a la vista **Authentic** de XMLSpy y Authentic Desktop excepto por un par de características:

- Los ayudantes de entrada de la vista **Authentic** no están disponibles en la IGU. Para insertar o anexar nodos haga clic con el botón derecho y seleccione una opción del menú contextual.
- No se pueden insertar tablas CALS/HTML.
- No están disponibles los iconos de estado del texto.

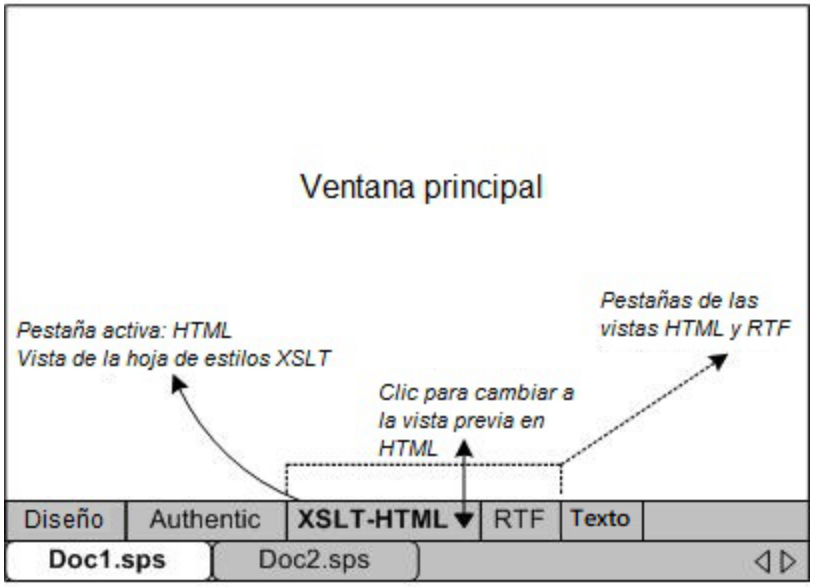
Para probar estas características utilice la vista **Authentic** de XMLSpy o de Authentic Desktop. En la sección Edición en la vista Authentic 662 puede encontrar información detallada sobre el uso de la vista **Authentic**. Para más información consulte el tutorial de la vista **Authentic** disponible en el manual del usuario de XMLSpy o de Authentic Desktop.

2.1.3 Vistas de resultados

En StyleVision hay tres pestañas de resultados: **HTML**, **RTF** y **texto**. Cada pestaña (*ver ilustración*) puede mostrar: (i) la hoja de estilos XSLT para el formato de salida correspondiente y (ii) una vista previa del documento que se genera al transformar el archivo XML de trabajo 30 con la hoja de estilos XSLT.

Nota: para que todas las vistas funcionen correctamente necesita instalar los programas relevantes para cada una de ellas, como se describe en la sección Configurar StyleVision 28.

En las pestañas de las vistas de resultados puede cambiar de la hoja de estilos XSLT a la vista previa del documento de salida haciendo clic en la punta de flecha de la pestaña y seleccionando la opción correspondiente.



Vista XSLT
La vista **XSLT** muestra la hoja de estilos XSLT generada para el formato de salida correspondiente a partir del archivo SPS activo. Cada vez que haga clic en la pestaña de la vista **XSLT**, la hoja de estilos se genera desde cero.

Las hojas de estilos aparecen en la vista de resultados con numeración de líneas y plegamiento de código. Haga clic en los iconos + y - del margen izquierdo para expandir/contraer elementos. La hoja de estilos de la vista **XSLT** no se puede editar pero puede buscar texto (seleccione **Editar | Buscar** 794) y copiar texto en el portapapeles (seleccione **Edición | Copiar** 794).

Nota: las hojas de estilos XSLT generadas a partir del archivo SPS se pueden generar y guardar por separado con el comando **Archivo | Guardar archivos generados** 782.

Vista previa de resultados

Las pestañas de vista previa muestran el resultado que se obtiene al transformar el archivo XML de trabajo 30 con la hoja de estilos XSLT en cada formato de salida. El resultado se genera desde cero cada vez que se hace clic en la pestaña de resultados. Recuerde que lo que se transforma es la versión guardada del archivo XML de trabajo y no la versión temporal que se edita en la vista **Authentic**. Esto significa que los cambios realizados en la vista **Authentic** solamente se reflejan en la vista previa de los resultados si guardó los cambios en el archivo XML de trabajo 30 (**Archivo | Guardar datos XML de Authentic** 782).

Si al hacer clic en la pestaña de vista previa no hay asignado ningún archivo XML de trabajo 30, StyleVision solicita un archivo XML de trabajo. Para los archivos SPS basados en bases de datos, no hace falta asignar un archivo XML de trabajo 30 porque StyleVision genera automáticamente un archivo XML no editable nada más cargarse la BD y este archivo XML se usa como archivo XML de trabajo 30.

Nota: los archivos de resultados generados a partir del SPS se pueden generar y guardar por separado con el comando **Archivo | Guardar archivos generados** 782.

 © 2016-2022 Altova GmbH

2.2 Barras laterales

Las barras laterales (o ventanas de las barras laterales) son componentes de la interfaz que ayudan a diseñar el SPS y ofrecen información relacionada con la vista activa. Las barras laterales (*ver lista*) se describen en los apartados de esta sección.

- Vista general del diseño 45
- Estructura del esquema 48
- Estructura del diseño 52
- Repositorio de estilos 55
- Estilos 58
- Propiedades 59
- Proyecto 63
- Messages 65
- Find and Replace 65

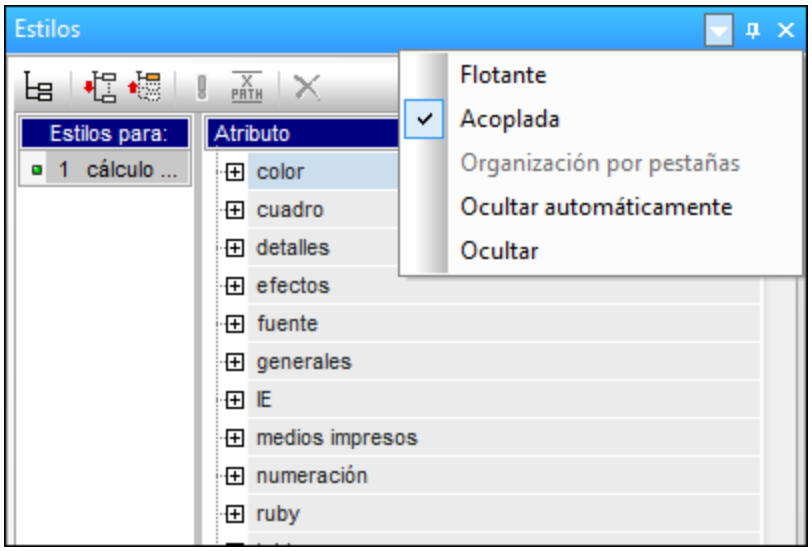
Organización de las vistas

Cuando hablamos de *organización de las vistas* nos referimos a qué barras laterales están disponibles en cada vista y qué posición tienen en la interfaz. La organización de las vistas se puede personalizar: (i) mostrando/ocultando las barras laterales de la vista (con el menú **Vista** o haciendo clic con el botón derecho en la barra de título de la ventana y seleccionando **Ocultar**) y (ii) cambiando la posición de la barra lateral. La nueva organización de la vista se conserva hasta que el usuario vuelva a cambiarla o restaure la vista a su organización predeterminada. Por ejemplo, en la vista **Diseño** puede activar todas las barras laterales excepto la ventana Estilos. Esta organización de la vista **Diseño** se conserva hasta que el usuario decida cambiarla o restaurarla. No olvide que la organización definida para las vistas previas (pestañas **HTML**, **RTF** y **texto**) afecta a la vista previa de todos los documentos de salida. Estas son las vistas que puede personalizar según lo que acabamos de explicar: (i) la vista principal sin abrir ningún documento, (ii) la vista **Diseño**, (iii) las vistas de resultados y la vista **Authentic**.

Ventanas flotantes y acopladas

Las ventanas de las barras laterales pueden estar acopladas en la interfaz de StyleVision o pueden flotar en la pantalla. Para acoplar una ventana, arrástrela por su barra de título y suéltela en una de las flechas interiores o exteriores que aparecen al arrastrar la ventana. Las flechas interiores acoplan la ventana en uno de los cuatro laterales de la ventana en la que aparecen las flechas. Las flechas exteriores acoplan la ventana en uno de los cuatro laterales de la ventana de la interfaz. Si quiere que una ventana flote en la pantalla, tiene dos opciones: (i) hacer doble clic en la barra de título o (ii) arrastrar la ventana por la barra de título hasta que flote en la pantalla.

También puede hacer clic en el botón de **menú** (*imagen siguiente*) de la parte superior derecha de la ventana acoplada y seleccionar **Flotante** en el menú emergente. También puede acceder a este menú haciendo clic con el botón derecho en la barra de título de la ventana acoplada.

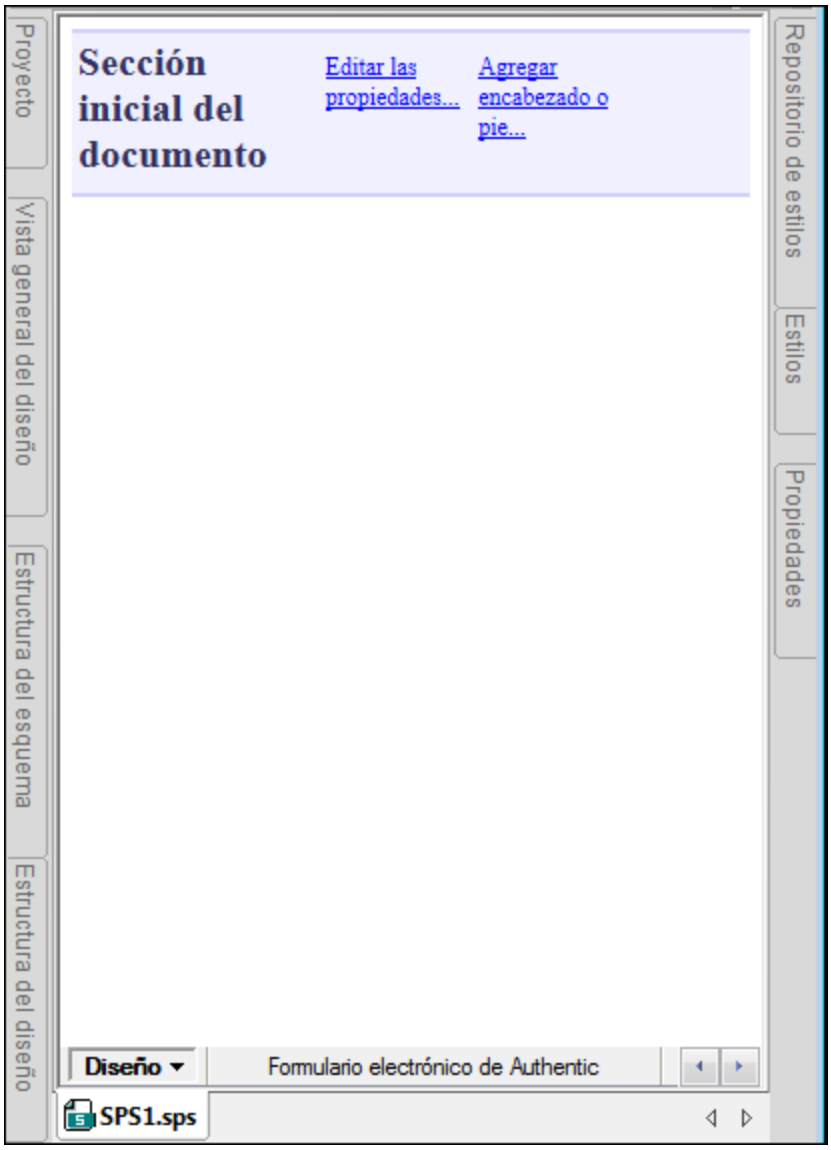


Para acoplar una ventana flotante, haga clic con el botón derecho en su barra de título y seleccione **Acoplada** en el menú emergente. La ventana se acoplará en la posición donde estuvo acoplada por última vez.

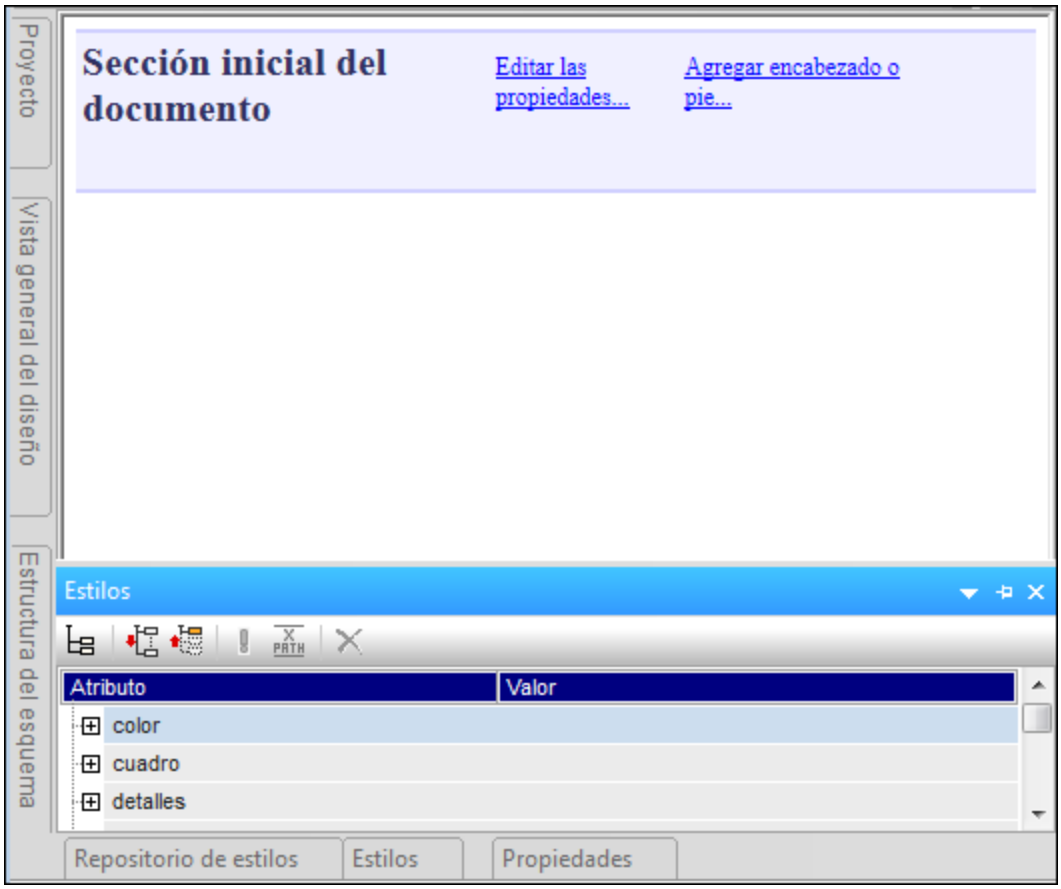
Ocultar automáticamente las ventanas de las barras laterales

Las ventanas acopladas se pueden ocultar automáticamente. Cuando una ventana de las barras laterales se oculta automáticamente, se minimiza en una pestaña situada en el borde de la IGU. En la imagen siguiente, por ejemplo, se ocultaron automáticamente cuatro barras laterales: dos en el borde izquierdo y dos en el borde derecho.

 © 2016-2022 Altova GmbH



Al pasar el puntero del ratón por las pestañas, la ventana que estaba oculta se despliega en la ventana principal. Por ejemplo, en la imagen siguiente, si pasamos el puntero por la pestaña Estilos, la ventana Estilos se despliega sobre la ventana principal.



Para volver a ocultar la ventana, mueva el puntero del ratón fuera de la ventana y de su pestaña.

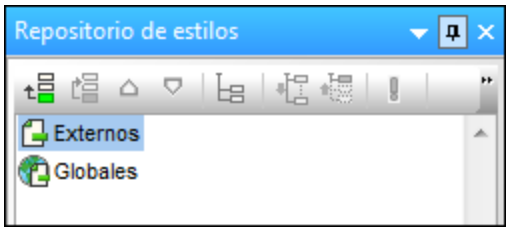
Esta función permite esconder las barras laterales menos utilizadas y tener más espacio libre en la pantalla. Cuando necesite usar estas ventanas otra vez, puede abrirlas con total facilidad.

Para ocultar una ventana automáticamente en una ventana acoplada, haga clic en el botón **Ocultar automáticamente** (el icono en forma de chincheta) situado en la parte superior derecha de la ventana (*imagen siguiente*). Otra opción es hacer clic en el botón de **menú** 42/hacer clic con el botón derecho en la barra de título y seleccionar la opción **Ocultar automáticamente**.

La ventana se oculta automáticamente.

Para desactivar el estado **Ocultar automáticamente** de una ventana, pase el puntero por su pestaña para que aparezca la ventana y después haga clic en el botón **Ocultar automáticamente** (*imagen siguiente*). Otra opción es hacer clic en el botón de **menú** 42/hacer clic con el botón derecho en la barra de título y seleccionar la opción **Ocultar automáticamente**.

 © 2016-2022 Altova GmbH



Nota: cuando se desactiva el estado **Ocultar automáticamente** de una ventana, su icono en forma de chincheta apunta hacia abajo. Cuando está activado, el icono apunta hacia la izquierda.

Ocultar (cerrar) las ventanas de las barras laterales

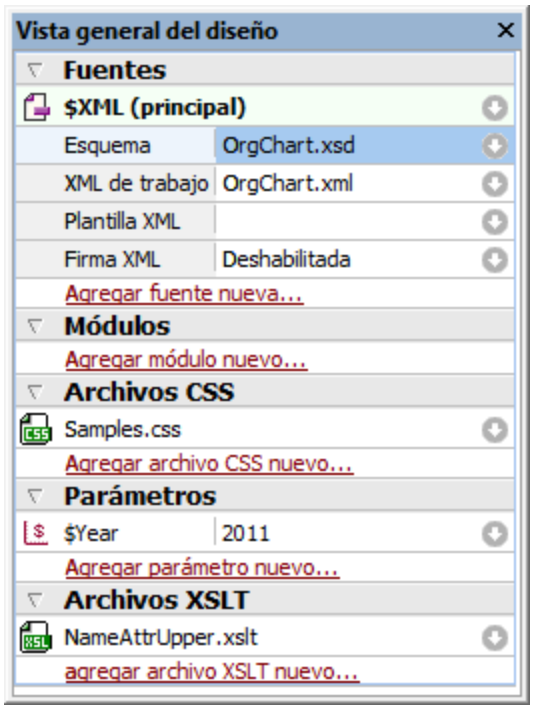
Cuando se oculta una ventana de las barras laterales, la ventana deja de estar visible en la IGU, tanto en su formato maximizado (acoplada o flotante) como en su formato minimizado (como pestaña en el borde de la IGU gracias a la función **Ocultar automáticamente** 42).

Para ocultar una ventana, haga clic en el botón **Cerrar** situado en la esquina superior derecha de la ventana flotante o acoplada. Otra opción es hacer clic en el botón de **menú** / hacer clic con el botón derecho en la barra de título y seleccionar la opción **Ocultar**.

Para volver a ver una ventana oculta (o cerrada), haga clic en el menú **Vista** 810 y seleccione el nombre de la ventana que desea volver a ver. La ventana aparece otra vez en la posición donde estaba cuando se ocultó (flotante o acoplada).

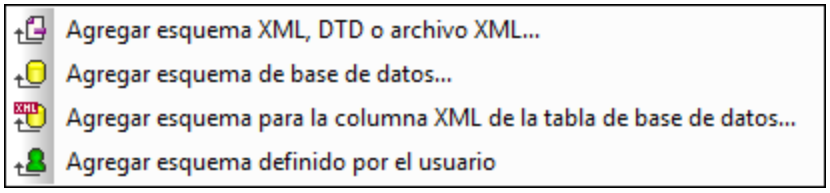
2.2.1 Vista general del diseño

La barra lateral **Vista general del diseño** (*imagen siguiente*) permite agregar esquemas fuente, parámetros globales, módulos SPS y archivos CSS al diseño SPS activo. Ofrece una visión de conjunto de estos componentes y permite gestionarlos cómodamente desde el mismo sitio.



Agregar esquemas fuente

Puede agregar esquemas fuente a los diseños SPS vacíos. Para añadir un esquema fuente haga clic en el vínculo `Agregar fuente nueva...` situado debajo de la sección **Fuentes**. Aparece un menú desplegable (*imagen siguiente*) donde puede elegir entre añadir: (i) un esquema XML, una DTD o un esquema XML generado automáticamente a partir de un archivo XML; (ii) un esquema generado automáticamente a partir de una BD; o (iii) un esquema definido por el usuario.



El archivo XML de trabajo y el archivo de plantilla XML

Tras añadir el esquema, este aparece bajo la sección **Fuentes**. Cada esquema tiene dos subelementos:

- el archivo XML de trabajo 30 y
- el archivo de plantilla XML 30.

Agregar módulos, archivos CSS, parámetros y archivos XSLT

Haga clic en uno de los comandos **Agregar...nuevo** de las secciones **Módulos**, **Archivos CSS**, **Parámetros** y **Archivos XSLT** para agregar un elemento nuevo.

 © 2016-2022 Altova GmbH

Características de la vista general del diseño

Estas son las características de las diferentes secciones (Fuentes, Parámetros, etc.) de la Vista general del diseño:

- Las secciones se pueden expandir o contraer haciendo clic en la punta de flecha situada a la izquierda del nombre de sección.
- En las secciones **Fuentes**, **Módulos** y **Archivos CSS** solo se enumeran los nombres de archivo. Al pasar el puntero del ratón por un nombre de archivo aparece su ruta de acceso completa.
- Los elementos que aparecen en gris están presentes en un módulo importado y no en el archivo SPS que está activo en la ventana principal.
- Cada sección incluye un comando **Agregar...nuevo** con el que puede agregar un elemento nuevo a la sección. Por ejemplo, haga clic en el vínculo `Agregar parámetro nuevo...` para añadir un parámetro nuevo al diseño SPS y a la sección **Parámetros** de la Vista general del diseño.
- Cada elemento tiene un menú contextual. Para abrirlo haga clic con el botón derecho en el elemento o clic en el icono (situado a la derecha del nombre del elemento).
- El icono **Quitar** del menú contextual quita el elemento del diseño.
- El comando **Editar archivo en XMLSpy** del menú contextual abre el archivo seleccionado en la aplicación Altova XMLSpy.
- Los comandos **Subir** y **Bajar** del menú contextual solamente están disponibles si selecciona uno de los diferentes módulos 255 de la sección **Módulos**. Estos comandos suben o bajan el módulo seleccionado una posición con respecto al módulo adyacente.

Fuentes

La sección **Fuentes** enumera los esquemas en los que se basa el diseño SPS y el archivo XML de trabajo y de plantilla XML asignado al diseño SPS. Puede cambiar estos archivos con el menú contextual (clic con el botón derecho o clic en el icono) y seleccionando el comando **Asignar esquema/archivo...** correspondiente.

Módulos

La sección **Módulos** enumera los módulos SPS 251 utilizados por el diseño SPS activo. Puede anexar nuevos módulos a la lista haciendo clic en el vínculo `Agregar módulo nuevo...` y buscando el archivo SPS pertinente. Si utiliza varios módulos, no olvide que el orden en que se enumeran los módulos 255 es importante. Si desea cambiar el orden de los módulos, utilice los comandos **Subir** / **Bajar** (disponibles en el menú contextual de cada módulo). El menú contextual de cada módulo incluye también un comando para abrir el módulo seleccionado en StyleVision.

Nota: la Vista general del diseño enumera todos los módulos del diseño para que pueda gestionarlos a nivel de archivo. Sin embargo, los diferentes objetos de módulo 252 (objetos incluidos dentro de un módulo) se enumeran en la ventana **Estructura del diseño** 52.

Archivos CSS

La sección **Archivos CSS** enumera todos los archivos CSS utilizados por el diseño SPS activo. Puede anexar nuevos archivos CSS a la lista haciendo clic en el vínculo `Agregar archivo CSS nuevo...` y buscando el archivo CSS pertinente. Si utiliza varios archivos CSS, no olvide que el orden en que se enumeran los módulos 391 es importante. Si desea cambiar el orden de los módulos, utilice los comandos **Subir** / **Bajar** (disponibles en el menú contextual de cada archivo CSS). El menú contextual de cada módulo incluye también un comando para abrir el módulo seleccionado en XMLSpy.

Nota: la Vista general del diseño enumera todos los archivos CSS para que pueda gestionarlos a nivel de archivo. Sin embargo, las diferentes reglas 391 CSS 391 de cada archivo CSS se enumeran en la ventana **Repositorio de estilos** 55.

Parámetros

La sección **Parámetros** enumera todos los parámetros globales utilizados en el diseño SPS activo. Puede añadir nuevos parámetros con el vínculo `Agregar parámetro nuevo...`. Para editar el nombre o el valor del parámetro haga doble clic donde corresponda. Para quitar un parámetro, selecciónelo y haga clic en el comando **Quitar** de su menú contextual.

Archivos XSLT

La sección **Archivos XSLT** enumera los archivos XSLT que se importaron en el diseño SPS activo. Las plantillas XSLT de estos archivos XSLT estarán disponibles como plantillas globales. Para más información consulte el apartado Plantillas XSLT 282.

Temas relacionados

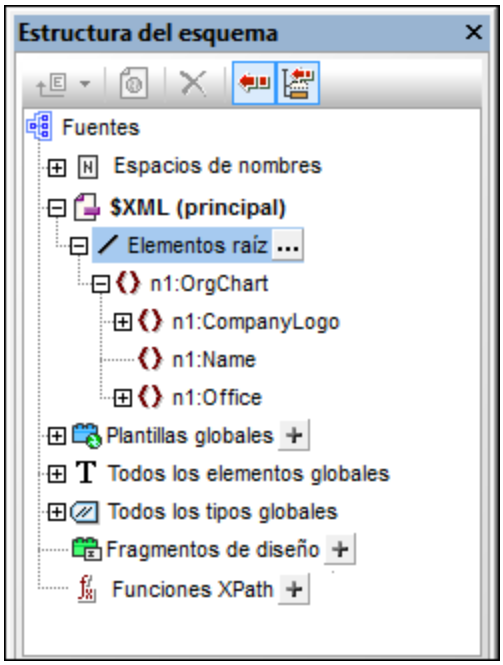
- Parámetros 327
- Estructura del esquema 48
- Archivos SPS modulares 251
- Fragmentos de diseño 278
- Usar scripts 497
- Configurar impresión 419

2.2.2 Estructura del esquema

La ventana **Estructura del esquema** (*imagen siguiente*) sirve para:

- Seleccionar varios elementos raíz (elementos de documento) para un esquema.
- Arrastrar nodos (elementos, atributos, tipos globales) desde la estructura del esquema y colocarlos en el diseño. Estos nodos representan el contenido XML que debe incluirse en los documentos de salida.
- Ver una lista de elementos y tipos globales del esquema fuente.
- Crear una plantilla global a partir de un elemento global o de un tipo global.
- Ver una lista de todos los espacios de nombres utilizados en el diseño SPS.
- Insertar y editar fragmentos de diseño 278.
- Insertar y editar funciones XPath 475 definidas por el usuario para el diseño SPS.

© 2016-2022 Altova GmbH

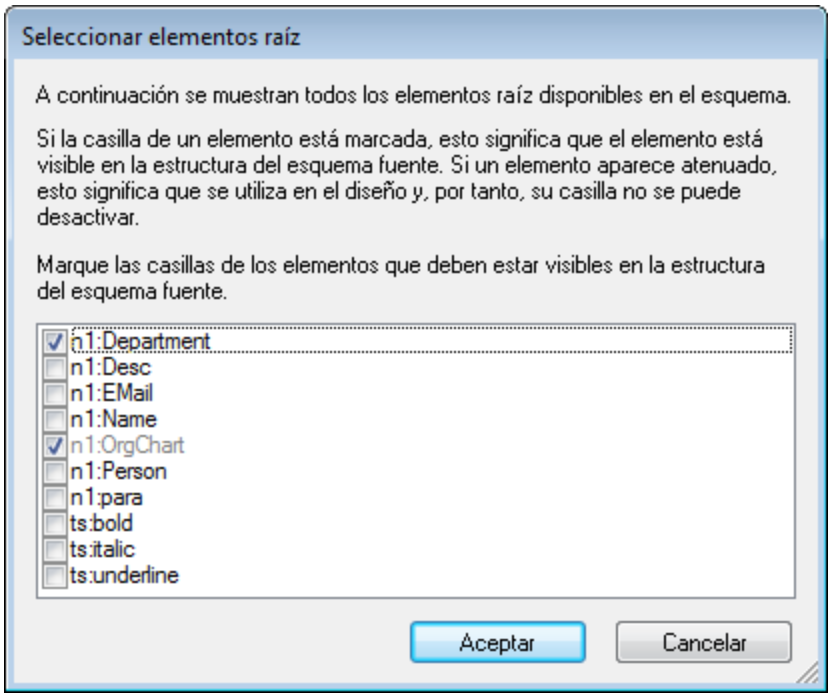


Elementos raíz

En la ventana Estructura del esquema, se enumeran los elementos raíz 30 (o elementos de documento 30) de cada esquema bajo la sección **$XML**. La lista de la sección **$XML** contiene todos los elementos raíz seleccionados para el esquema (más abajo explicamos cómo hacerlo). Cada elemento raíz se puede expandir para ver la estructura de su modelo de contenido. El contenido de la plantilla principal se crea a partir de los nodos de estos elementos raíz. No olvide que el punto de entrada de la plantilla principal es el nodo de documento del esquema principal, el cual puede seleccionar o cambiar siempre que quiera (más abajo explicamos cómo hacerlo).

Siga estos pasos para seleccionar los elementos raíz para un esquema:

1. Haga clic en el botón de selección ... situado a la derecha de `Elementos raíz`. Aparece el cuadro de diálogo "Seleccionar elementos raíz" (*imagen siguiente*).
2. Seleccione qué elementos globales del esquema deben utilizarse como elementos raíz. Para más información sobre las posibilidades que ofrece la selección de varios elementos raíz, consulte el apartado Archivo SPS: estructura | Esquemas fuente 221.



Además, bajo la sección **Todos los elementos globales** se enumeran todos los elementos globales del esquema. Por cada elemento global puede crear una plantilla global 266.

Elementos y tipos globales

Puede usar elementos y tipos globales para crear plantillas globales 266 que se pueden reutilizar en otras plantillas. Además, los tipos globales se pueden usar en las plantillas directamente.

Fragmentos de diseño

En la ventana Estructura del esquema se enumeran todos los fragmentos de diseño 278 del documento. Para ver la lista basta con expandir la sección **Fragmentos de diseño** de la ventana. Tenga en cuenta estos aspectos de la sección **Fragmentos de diseño** de la ventana Estructura del esquema:

- Puede crear un fragmento de diseño nuevo haciendo clic en el icono **Agregar** ⊞.
- Puede editar el nombre del fragmento de diseño haciendo doble clic en él.
- Puede habilitar o deshabilitar un fragmento de diseño con solo activar/desactivar su casilla.
- Puede arrastrar un fragmento de diseño desde la estructura del esquema hasta el diseño.

Para más información consulte el apartado Fragmentos de diseño 278.

Funciones XPath definidas por el usuario

Puede añadir una función XPath definida por el usuario haciendo clic en el botón **Agregar** ⊞ de la sección **Funciones XPath**. Una vez creada, la función XPath aparece en la ventana Estructura del esquema. Haga doble clic en una función para editarla. Tenga en cuenta que:

- Puede habilitar o deshabilitar una función XPath con solo activar/desactivar su casilla.

 © 2016-2022 Altova GmbH

- Puede renombrar o quitar una función XPath con solo hacer clic con el botón derecho y seleccionar la opción correspondiente en el menú contextual.

Para más información consulte el apartado Funciones XPath definidas por el usuario 475.

Espacios de nombres

En la ventana Estructura del esquema se enumeran todos los espacios de nombres (y sus prefijos) utilizados por el diseño SPS bajo la sección **Espacios de nombres**. Esta lista incluye dos tipos de espacios de nombres: (i) los espacios de nombres definidos en el esquema o los esquemas referenciados (*ver nota*) y (ii) los espacios de nombres que se añaden por defecto al SPS recién creado. Esta lista puede ser muy útil a la hora de escribir expresiones XPath. Además puede establecer un espacio de nombres predeterminado para todo el SPS haciendo doble clic en el valor de la entrada `xpath-default-ns` y escribiendo el espacio de nombres.

Nota: si desea añadir un espacio de nombres a un SPS o a una hoja de estilos XSLT que se genera a partir del SPS, debe añadir el espacio de nombres al elemento de nivel superior `schema` del esquema XML en el que se basa el SPS.

Iconos de la ventana y de su barra de herramientas

A continuación aparecen los iconos y comandos de la ventana Estructura del esquema.

En un esquema definido por el usuario 229 añade un elemento secundario al elemento de documento o anexa un elemento del mismo nivel al elemento seleccionado. Al hacer clic en la punta de flecha del icono aparece una lista desplegable con otros comandos con los que podrá crear un esquema definido por el usuario 229.

Convertir en plantilla global o quitarla. Se habilita al seleccionar un elemento o tipo global.

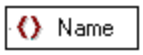
Eliminar el elemento seleccionado.

Sincronizar la estructura. Cuando está activo (cuando el icono aparece resaltado en azul), al seleccionar un nodo en la estructura, se selecciona (i) el nodo correspondiente en el diseño y (ii) el nodo correspondiente en la Estructura del esquema. Cuando este icono está desactivado, el nodo solamente se selecciona en la Estructura del esquema. Desactive la sincronización si le resultada difícil arrastrar un nodo desde la Estructura del esquema hasta el diseño.

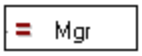
Contraer automáticamente los demás elementos de la estructura cuando está activado el icono **Sincronizar estructura** y se selecciona un elemento en el diseño. Este icono solamente se habilita cuando está activado el icono **Sincronizar estructura**.

Símbolos utilizados en la estructura del esquema

Estos son los símbolos utilizados en los nodos de la estructura del esquema:



Elemento

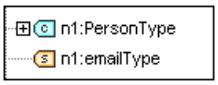


Atributo

Elemento con elemento secundarios. Al hacer doble clic en el elemento (o al hacer clic en el símbolo +/-) el elemento se

expande/contrae.

Se aplica un filtro de BD. Solamente afecta a los elementos de tabla de datos de nivel superior de la estructura del esquema.



Los tipos globales pueden ser complejos o simples. Los tipos complejos se señalan con un icono celeste. Los tipos simples se señalan con un icono marrón.

2.2.3 Estructura del diseño

La ventana **Estructura del diseño** (*imagen siguiente*) ofrece una visión de conjunto del diseño SPS.

En el nivel superior de la Estructura del diseño aparece el nombre del archivo SPS (pase el cursor del ratón sobre el nombre de archivo para ver su ruta de acceso completa). El siguiente nivel de la Estructura del diseño se divide en varias categorías:

- **Scripts** 497: muestra todas las funciones JavaScript definidas para el archivo SPS con ayuda del editor de JavaScript de StyleVision.
- **Plantilla principal** 265: muestra una estructura detallada de la plantilla principal.
- **Plantillas globales** 266: enumera las plantillas globales del SPS actual y las plantillas globales de todos los módulos SPS incluidos.
- **Fragmentos de diseño** 54: muestra todos los fragmentos de diseño del SPS y permite crear, renombrar y eliminar fragmentos.
- **Plantillas XSLT** 55: permite ver las plantillas XLST de archivos XSLT importados.
- **Funciones XPath definidas por el usuario** 475: permite crear, editar, renombrar y eliminar las funciones XPath del usuario.

Iconos de la ventana y de su barra de herramientas

A continuación aparecen los iconos y comandos de la ventana Estructura del diseño.

Añade un fragmento de diseño, una plantilla principal o un elemento de diseño al SPS. Haga clic en la parte izquierda del icono para añadir un fragmento de diseño. Haga clic en la punta de flecha

 © 2016-2022 Altova GmbH

para abrir un menú desplegable con más opciones.

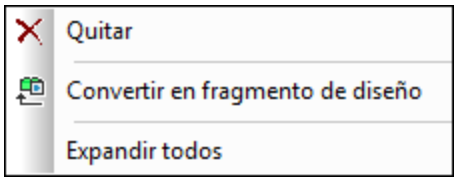
Elimina el elemento seleccionado. El icono se habilita cuando se selecciona un elemento de la categoría **Plantillas globales** o de la plantilla principal.

Sincronizar la estructura. Cuando está activo (cuando el icono aparece resaltado en azul), al seleccionar un nodo en la estructura, se selecciona (i) el nodo correspondiente en el diseño y (ii) el nodo correspondiente en la Estructura del diseño. Cuando este icono está desactivado, el nodo solamente se selecciona en la Estructura del diseño. Desactive la sincronización si le resultada difícil arrastrar un nodo desde la Estructura del diseño hasta el diseño.

Contraer automáticamente los demás elementos de la estructura cuando está activado el icono **Sincronizar estructura** y se selecciona un elemento en el diseño. Este icono solamente se habilita cuando está activado el icono **Sincronizar estructura**.

Cambiar la presentación de la estructura del diseño

Puede cambiar la presentación de la ventana Estructura del diseño con las opciones del menú contextual (*imagen siguiente*), que aparece al hacer clic con el botón derecho en un elemento de la estructura del diseño.



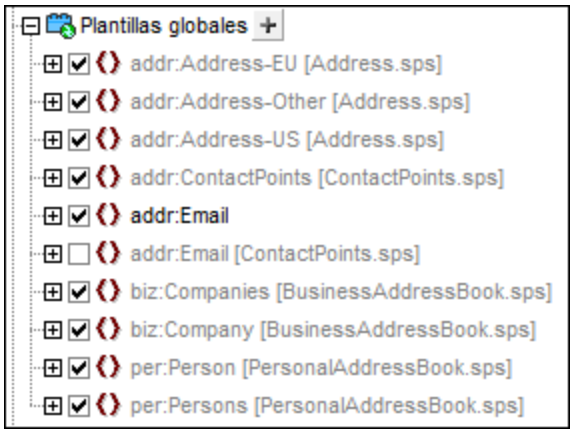
Haga clic en el comando **Quitar** para eliminar el elemento de la estructura del diseño. Haga clic en **Convertir en fragmento de diseño** para crear un fragmento de diseño 54 en el SPS y añadir el correspondiente elemento en la estructura del diseño. Haga clic en **Expandir todos** para expandir todos los elementos de la estructura del diseño.

Scripts y Plantilla principal

La categoría **Scripts** enumera todos los scripts del diseño, incluidos los que están en módulos importados. La categoría **Plantilla principal** muestra la estructura de la plantilla pricipal. Los elementos de la estructura y del diseño se pueden eliminar haciendo clic en ellos con el botón derecho y seleccionado **Quitar**.

Plantillas globales

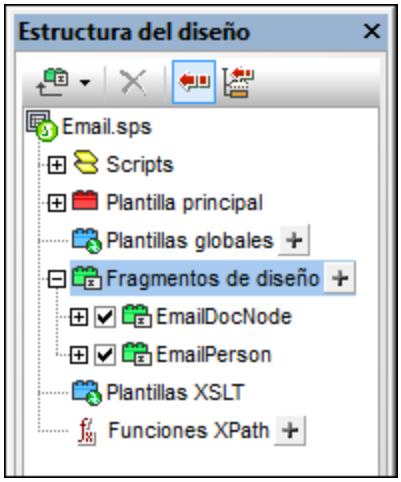
La categoría **Plantillas globales** 266 enumera todas las plantillas globales del SPS actual y de todos los módulos SPS importados. Las plantillas globales definidas en el SPS actual aparecen en color negro, mientras que las de los módulos importados aparecen en color gris (*imagen siguiente*). A la izquierda de cada plantilla global hay una casilla que sirve para activar/desactivar la plantilla. Cuando se desactiva una plantilla global, se elimina del diseño.



Para eliminar una plantilla global del SPS actual (no de un módulo importado) selecciónela y haga clic en el botón **Quitar** de la barra de herramientas o en el comando **Quitar** del menú contextual. El componente se elimina del diseño y de la Estructura del diseño.

Fragmentos de diseño

La categoría **Fragmentos de diseño** 278 enumera todos los fragmentos de diseño del SPS actual y de todos los módulos importados. Los fragmentos de diseño definidos en el SPS actual aparecen en negro y los de los módulos importados aparecen en gris (*imagen siguiente*). A la izquierda de cada fragmento de diseño hay una casilla que sirve para activar/desactivar el fragmento de diseño. Para eliminar un fragmento de diseño del SPS actual (no de un módulo importado) selecciónelo y haga clic en el botón **Quitar** de la barra de herramientas o en el comando **Quitar** del menú contextual. El componente se elimina del diseño y de la Estructura del diseño.

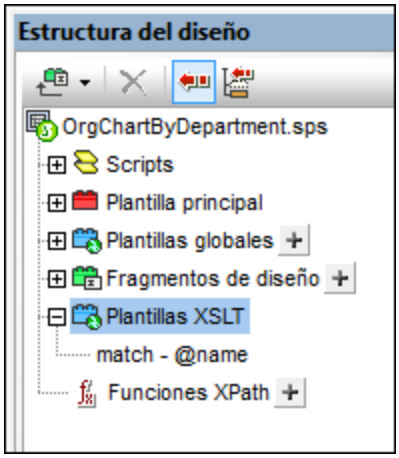


Para añadir un fragmento de diseño haga clic en el icono situado a la derecha de **Fragmentos de diseño**. Cada fragmento de diseño que se inserta tiene una estructura en forma de árbol, con nodos que se pueden expandir y contraer. Para quitar un componente del árbol del fragmento (del SPS actual), selecciónelo y haga clic en el botón **Quitar** de la barra de herramientas o en el comando **Quitar** del menú contextual. El componente se elimina del diseño y del árbol del fragmento.

 © 2016-2022 Altova GmbH

Plantillas XSLT

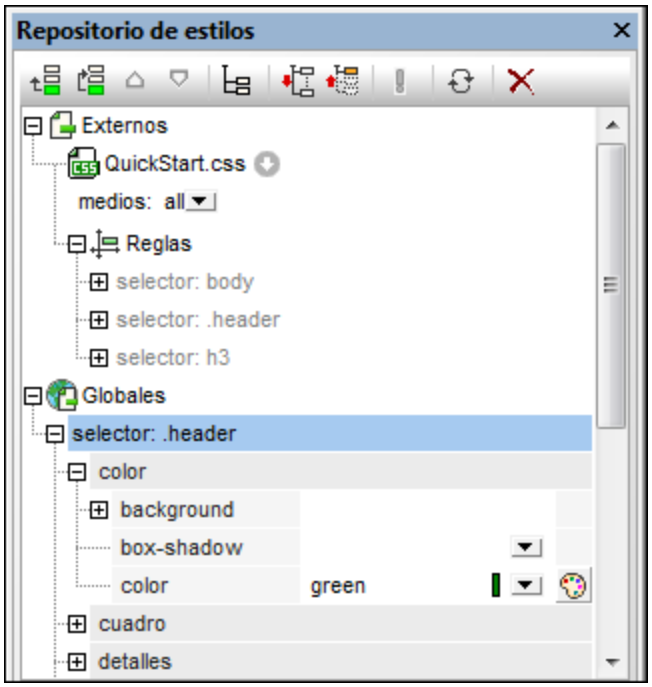
En la ventana **Estructura del diseño** (*imagen siguiente*), las plantillas XSLT incluidas en el archivo XSLT importado se enumeran bajo la categoría **Plantillas XSLT**.



Hay dos tipos de plantillas XSLT importadas: (i) plantillas `match` y (ii) plantillas `name` (plantillas con nombre). La ventana Estructura del diseño muestra también (i) el valor del atributo `select` de las plantillas `match` y (ii) el valor del atributo `name` de las plantillas con nombre. Para más información sobre cómo funcionan las plantillas XSLT, consulte el apartado Plantillas XSLT 282.

2.2.4 Repositorio de estilos

En la ventana **Repositorio de estilos** (*imagen siguiente*) puede asignar hojas de estilos CSS externas y definir estilos CSS globales para el archivo SPS. Las reglas de estilo de las hojas de estilos CSS externas y de los estilos CSS definidos globalmente afectan a la vista **Authentic** y al documento HTML de salida.



La ventana Repositorio de estilos incluye dos listas, la de estilos **Externos** y la de estilos **Globales**. Ambas listas tienen una estructura en forma de árbol. La lista **Externos** enumera todas las hojas de estilos CSS externas asociadas al archivo SPS. La lista **Globales** enumera todos los estilos globales asociados al archivo SPS.

Esta es la estructura de las listas del **Repositorio de estilos**:

```
Externos
- CSS-1.css (La ruta de acceso aparece al pasar el puntero del ratón por encima)
  - medios (se pueden definir en la ventana Repositorio de estilos)
  - Reglas (no se pueden editar. Solamente se pueden editar en el archivo CSS)
    - Selector-1
      - Property-1
      - ...
      - Propiedad-N
    - ...
    - Selector-N
+ ...
+ CSS-N.css
Globales
- Selector-1
  + Selector-1 Propiedades
- ...
+ Selector-N
```

Prioridad de las reglas de estilo

Si una regla de estilo global y una regla de una hoja de estilos CSS externa tienen selectores que identifican al mismo componente de documento, la regla de estilo global tiene prioridad sobre la de la hoja de estilos externa y, por tanto, se aplicará. Si dos reglas de estilo globales o más seleccionan el mismo componente de documento, entonces se aplica la última de las reglas. Igualmente, si dos o más reglas de estilo de hojas de

 © 2016-2022 Altova GmbH

estilos externas seleccionan el mismo componente de documento, se aplica la regla de la última hoja de estilos.

Administrar estilos en el Repositorio de estilos

Estos son los comandos disponibles en la ventana Repositorio de estilos (iconos de la barra de herramientas y comandos del menú contextual):

	Agregar	Añade una hoja de estilos externa nueva a la lista **Externos** o un estilo global nuevo a la lista **Globales**, dependiendo de cuál de las dos listas está seleccionada al hacer clic en el icono. La entrada nueva se anexa a la lista de componentes. El comando **Agregar** también aparece en el menú contextual. Para más información sobre el uso de hojas de estilos externas y estilos globales, consulte el apartado Trabajar con estilos CSS 390. Recuerde que también puede añadir/eliminar hojas de estilos CSS externas desde la ventana **Vista general del diseño** 45.
	Insertar	Inserta una hoja de estilos externa nueva encima de la hoja de estilos externa seleccionada (en la lista **Externos**) o un estilo global nuevo encima del estilo global seleccionado (en la lista **Globales**). El comando **Insertar** también está disponible en el menú contextual. Para más información sobre el uso de hojas de estilos externas y estilos globales, consulte el apartado Trabajar con estilos CSS 390.
/	**Subir / Bajar**	Sube o baja la hoja de estilos externa seleccionada o el estilo global seleccionado un nivel en relación a las demás entradas de la lista. Estos comandos son útiles a la hora de cambiar el orden de prioridad de las hojas de estilos externas y de las reglas de estilo globales. Estos comandos también están disponibles en el menú contextual. Para más información sobre el uso de hojas de estilos externas y estilos globales, consulte el apartado Trabajar con estilos CSS 390.
	Mostrar elementos no vacíos **Expandir todos** **Contraer todos**	Estos iconos controlan la presentación de los estilos de un selector. Todos los selectores, ya sean de hojas de estilos externas o selectores definidos globalmente, pueden presentarse de tres maneras diferentes. **Mostrar elementos no vacíos**: solamente se muestan las propiedades que tienen un valor. Esto permite despejar un poco la vista de la ventana. **Expandir todos** y **Contraer todos** permiten expandir y contraer respectivamente todas las definiciones de estilos del selector seleccionado. Estos comandos también están disponibles en el menú contextual.
	Marcar como importante	Active este icono para dar el valor CSS `!important` a la regla CSS seleccionada. Para quitarle la marca de importante, desactive el icono.
	Volver a cargar todo	Vuelve a cargar todas las hojas de estilos CSS externas.
	Restaurar	Elimina la hoja de estilos externa seleccionada o el estilo global seleccionado.

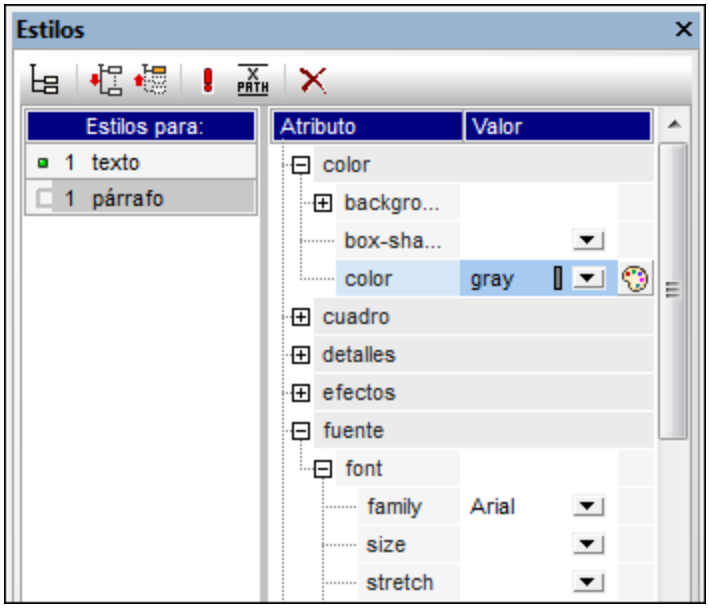
Editar estilos CSS en el Repositorio de estilos

Estos son los mecanismos de edición que ofrece la ventana **Repositorio de estilos**:

- Puede agregar o eliminar una hoja de estilos CSS y puede especificar los medios a los que se aplica cada hoja de estilos CSS externa. Para más información consulte el apartado Hojas de estilos CSS externas 391.
- Puede editar los selectores y las propiedades de los estilos globales directamente en la ventana Repositorio de estilos. Para más información consulte el apartado Definir estilos CSS globalmente 395.

2.2.5 Estilos

La barra lateral **Estilos** (*imagen siguiente*) permite definir localmente estilos CSS para los componentes SPS seleccionados en la vista **Diseño**. Para definir estilos globalmente utilice la ventana Repositorio de estilos 55.



La ventana Estilos se divide en dos paneles:

- A la izquierda, el panel **Estilos para:** enumera los tipos de componente seleccionados. Cuando se selecciona una sección de la vista **Diseño**, la sección del diseño puede contener varios componentes. Los tipos de componente seleccionados se enumeran en el panel **Estilos para:**. Cada uno de estos tipos de componente se pueden seleccionar para darles estilos. Si solamente hay una instancia del tipo de componente, entonces se selecciona dicha instancia para darle estilos. Si hay varias instancias del tipo de componente, puede dar estilos a todas las instancias seleccionadas a la vez. Los estilos definidos se aplican localmente a cada instancia. Si quiere dar estilos a una instancia concreta, seleccione la instancia del componente en la vista **Diseño** y seleccione los estilos en la ventana **Estilos**. También puede seleccionar una serie de componentes: seleccione el primer componente de la serie y seleccione el último componente de la serie mientras pulsa la tecla **Mayús**. Para más información sobre la selección de tipos de componentes consulte el apartado Definir estilos CSS localmente 398.
- A la derecha, el panel **Definiciones de estilo** sirve para definir los estilos CSS para los tipos de componente seleccionados en el panel **Estilos para:**. Este panel puede verse de tres maneras diferentes (*ver más abajo*). Para más información sobre cómo definir estilos consulte el apartado Establecer valores de estilos CSS 400. El icono **XPath** activa/desactiva la aplicación de

 © 2016-2022 Altova GmbH

expresiones XPath como fuente de valores de estilos. Si selecciona una propiedad de estilo y el icono **XPath** está activado, puede introducir una expresión XPath para la propiedad y el valor devuelto de la expresión se utilizará como valor de la propiedad de estilo. Así se puede usar el valor de un nodo de un documento XML en tiempo de ejecución como valor de una propiedad. Cuando el icono **XPath** está desactivado, puede introducir un valor estático como valor de la propiedad.

Opciones de presentación de las definiciones

Puede cambiar la presentación de las definiciones dependiendo de lo que necesite. Para ello utilice los tres botones de la barra de herramientas o sus correspondientes comandos en los menús contextuales:

- **Mostrar elementos no vacíos** : si activa este icono, solamente se muestran las propiedades (del tipo de componente seleccionado en el panel izquierdo) que tienen definido un valor. Además las propiedades se enumeran por orden alfabético. Si el icono está desactivado, se enumeran todas las propiedades. Esta icono es útil si quiere ver qué propiedades se definieron para el tipo de componente seleccionado. Si quiere definir propiedades nuevas para el tipo de componente seleccionado, primero deberá desactivar este icono.

- **Expandir todos** : en el panel derecho se expanden todas las propiedades visibles del tipo de componente seleccionado en el panel izquierdo. Este comando se puede usar junto con el comando **Mostrar elementos no vacíos**.

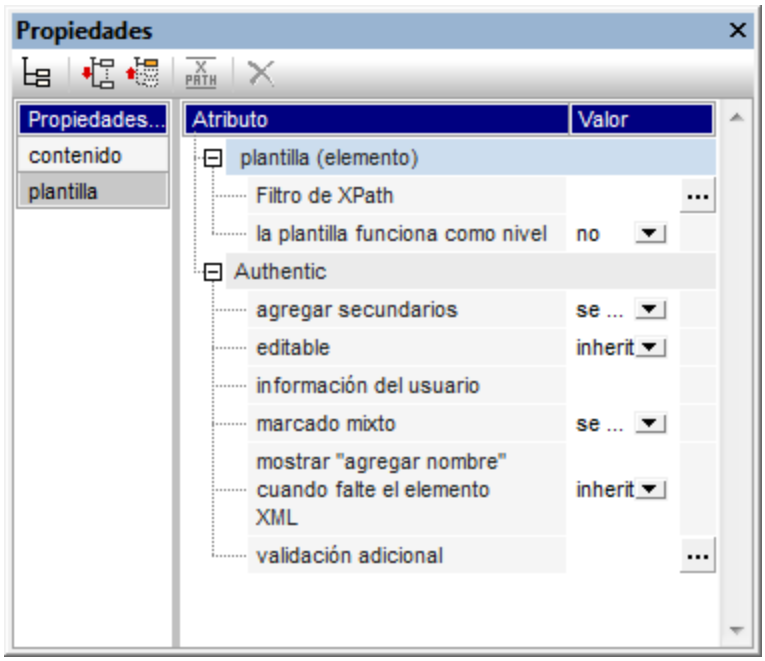
- **Contraer todos** : en el panel derecho se contraen todas las propiedades visibles del tipo de componente seleccionado en el panel izquierdo. Este comando se puede usar junto con el comando **Mostrar elementos no vacíos**.

Iconos Marcar como importante y Restaurar

Haga clic en el icono **Marcar como importante** para dar o quitar el valor CSS `!important` a la regla CSS seleccionada. Haga clic en el icono **Restaurar** para restaurar el valor de la propiedad seleccionada.

2.2.6 Propiedades

La barra lateral **Propiedades** (*imagen siguiente*) permite definir propiedades para los componentes SPS seleccionados en la vista **Diseño**.



La barra lateral Propiedades se divide en dos paneles:

- A la izquierda, el panel **Propiedades para:** enumera los tipos de componente seleccionados. Puede seleccionar un tipo de componente y asignarle propiedades. En la imagen anterior, por ejemplo, se seleccionó el tipo de componente *plantilla*. Para más información consulte más abajo el subapartado Componentes y grupos de propiedades 61.
- A la derecha, el panel **Definiciones de propiedades**, donde se definen las propiedades para el tipo de componente seleccionado en el panel izquierdo. El panel Definiciones de propiedades se puede ver de tres maneras diferentes (*ver más abajo*). Para más información sobre los diferentes grupos de propiedades consulte más abajo el subapartado Grupo de propiedades 62.

Opciones de presentación de las definiciones

Puede cambiar la presentación de las definiciones dependiendo de lo que necesite. Para ello utilice los tres botones de la barra de herramientas o sus correspondientes comandos en los menús contextuales:

- **Mostrar elementos no vacíos** : si activa este icono, solamente se muestran las propiedades (del tipo de componente seleccionado en el panel izquierdo) que tienen definido un valor. Además las propiedades se enumeran por orden alfabético. Si el icono está desactivado, se enumeran todas las propiedades. Esta icono es útil si quiere ver qué propiedades se definieron para el tipo de componente seleccionado. Si quiere definir propiedades nuevas para el tipo de componente seleccionado, primero deberá desactivar este icono.

- **Expandir todos** : en el panel derecho se expanden todas las propiedades visibles del tipo de componente seleccionado en el panel izquierdo. Este comando se puede usar junto con el comando **Mostrar elementos no vacíos**.

- **Contraer todos** : en el panel derecho se contraen todas las propiedades visibles del tipo de

 © 2016-2022 Altova GmbH

componente seleccionado en el panel izquierdo. Este comando se puede usar junto con el comando **Mostrar elementos no vacíos**.

Icono Restaurar

Haga clic en el icono **Restaurar** para restaurar el valor predeterminado de la propiedad seleccionada.

Componentes y grupos de propiedades

Dependiendo del componente seleccionado están disponibles unas propiedades u otras. La tabla siguiente muestra los diferentes componentes SPS y los grupos de propiedades que les corresponden.

Componente	Grupo de propiedades
Plantilla	Plantilla, Authentic
Contenido	Contenido, Authentic, Generales, Evento
Texto	Texto, Generales, Evento
Cálculo automático	Cálculo automático, Authentic, Generales, Evento
Rama condicional	Cuando
Dispositivo de entrada de datos	Authentic, Generales, [Dispositivo de entrada de datos], Evento, HTML
Imagen	Imagen, Authentic, Generales, Evento, HTML
Enlace	Enlace, Authentic, Generales, HTML
Tabla	Tabla, Authentic, Generales, HTML, Interactividad
Párrafo	Párrafo, Authentic, Generales, Evento, HTML

Es necesario destacar estos aspectos sobre los tipos de componentes:

- Los componentes de tipo *plantilla* son la plantilla principal, las plantillas globales y los nodos de esquema del diseño.
- Los componentes de tipo *contenido* son los marcadores de posición `contenido` y `resto de contenido`. Estos marcadores de posición representan el contenido de texto de un nodo o de varios nodos del documento XML.
- Un componente de tipo *texto* es una sola cadena de texto estático. Una sola cadena aparece entre dos componentes de cualquier tipo (excepto componentes de tipo *texto*) e incluye espacios en blancos.
- Los *dispositivos de entrada de datos* son campos de entrada, campos de entrada multilínea, cuadros combinados, casillas de verificación, botones de opción y botones. Sus propiedades afectan tanto al dispositivo de entrada de datos como a su contenido, si lo tuviera.
- Un componente de tipo *tabla* es cualquier estructura de tabla del diseño. Estas estructuras suelen incluir subcomponentes, que se consideran componentes de pleno derecho. Se trata de los subcomponentes de tipo *fila*, *columna*, *celda*, *encabezado* y *pie*.
- Un componente de tipo *párrafo* hace referencia a cualquier formato predefinido.

La tabla que aparece a continuación enumera y describe los grupos de propiedades disponibles.

Grupo de propiedades	Descripción
Cálculo automático	Estas propiedades se habilitan cuando se selecciona un cálculo automático. La propiedad `Formato de valor` especifica el formato 380 de un cálculo automático de tipo numérico o de tipo fecha. La propiedad `XPath` especifica la expresión XPath que se utiliza para el cálculo automático 293.
Authentic	Estas propiedades propias del SPS están disponibles para los componentes de tipo plantilla, contenido, cálculo automático, dispositivos de entrada de datos, imágenes, enlaces, tablas y párrafos. Las propiedades disponibles en cada grupo dependen del tipo de componente. Para más información, consulte el apartado Propiedades de nodos Authentic 462.
Generales	Este grupo de propiedades está disponible para todos los tipos de componente excepto para los tipos plantilla y cálculo automático. Contiene estas propiedades: `clase` (nombre de clase), `dir` (dirección del texto), `id` (id. único), `lang` (idioma) y `title` (nombre).
Dispositivo de entrada de datos	Estas propiedades especifican el rango de valor de cuadros combinados, casillas de verificación y botones de opción. Este grupo de propiedades no afecta a campos de edición ni botones.
Evento	Propiedades que permiten definir funciones JavaScript 497 para estos eventos HTML del lado cliente: `onclick`, `ondblclick`, `onkeydown`, `onkeypressed`, `onkeyup`, `onmousedown`, `onmousemove`, `onmouseout`, `onmouseover`, `onmouseup`.
HTML	Disponibles para los tipos de componente dispositivo de entrada de datos 190, imagen 183, enlace 364, tabla 155, párrafos 375. Recuerde que hay varios tipos de dispositivos de entrada de datos 190 y varios tipos de párrafos 375 y que las tablas 155 tienen subcomponentes. Estas propiedades son propiedades HTML que se pueden definir en los correspondientes elementos HTML (`img`, `table`, `p`, `div`, etc). Las propiedades disponibles dependen, por tanto, del componente seleccionado. Los valores de estas propiedades se pueden seleccionar mediante expresiones XPath.

StyleVision también ofrece propiedades específicas para los componentes de tipo imagen 183, enlace 366, párrafos y otros formatos predefinidos 137 y ramas condicionales 307. Estas propiedades se describen más adelante.

Establecer el valor de las propiedades

El valor de las propiedades se puede introducir de una, dos o tres maneras diferentes, dependiendo de la propiedad:

 © 2016-2022 Altova GmbH

- Puede escribir el valor en la columna *Valor* directamente. Primero seleccione la propiedad, haga doble clic en el campo de la columna *Valor*, escriba el valor y finalmente pulse la tecla **Entrar** o haga clic en otra parte de la IGU.
- Puede seleccionar un valor de la lista desplegable del cuadro combinado de la propiedad. Haga clic en la flecha del cuadro combinado para abrir la lista desplegable de valores de la propiedad y seleccione uno.
- Puede usar el botón **Editar** situado a la derecha de la columna *Valor* de la propiedad. Al hacer clic en este botón se abre un cuadro de diálogo donde puede editar el valor de la propiedad.

También puede usar expresiones XPath para indicar los valores de algunas propiedades del grupo ***Generales*** y ***HTML***. El icono **XPath** activa y desactiva la aplicación de expresiones XPath como fuente de valores de propiedad. Si selecciona una propiedad y el icono **XPath** está activado, puede introducir una XPath para esta propiedad y el valor devuelto de la expresión se utiliza como valor de la propiedad. Por ejemplo, puede devolverse el valor de un nodo de un documento XML en tiempo de ejecución y utilizarse como valor de la propiedad. Cuando el icono **XPath** está desactivado, puede introducir un valor estático para la propiedad. Consulte el apartado *Propiedades de los estilos mediante XPath* 402 para obtener más información.

Cambiar o eliminar el valor de una propiedad

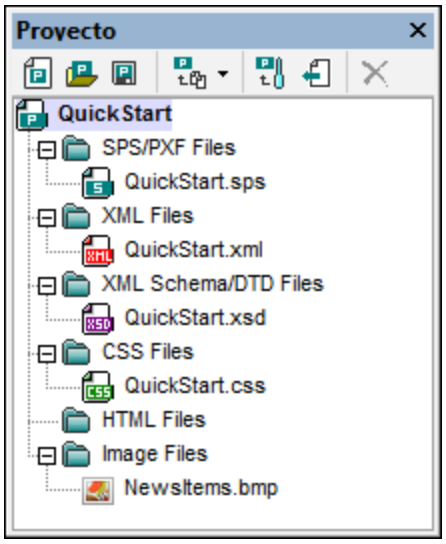
Para cambiar el valor de una propiedad, use los métodos descritos en el párrafo anterior *Establecer el valor de las propiedades* 62. Para eliminar el valor de una propiedad, seleccione la propiedad y haga clic en el icono **Restaurar** de la barra de herramientas de la ventana Propiedades.

2.2.7 Proyecto

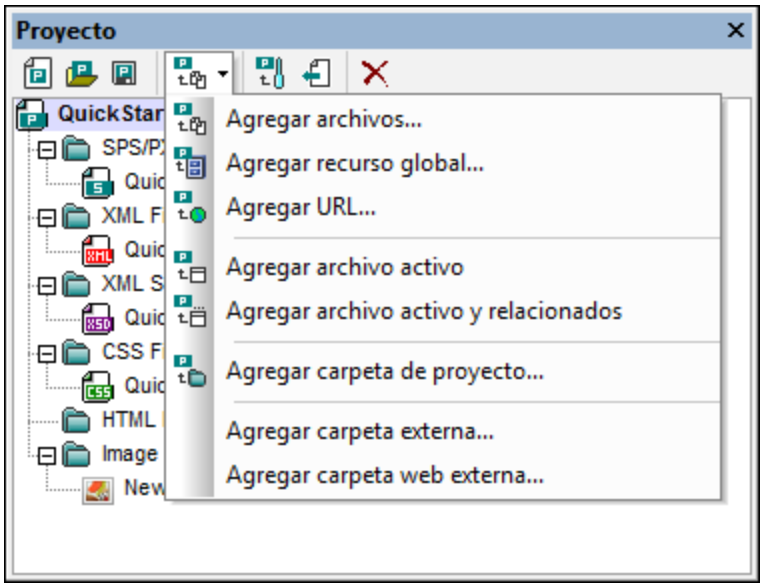
Los proyectos de StyleVision permiten agrupar, administrar y trabajar con archivos relacionados de forma eficaz. Una vez reunidos en un archivo de proyecto, los archivos se pueden abrir/editar desde la ventana Proyecto mientras diseña el SPS. Por ejemplo, puede arrastrar un archivo SPS desde la ventana Proyecto hasta la ventana Estructura del diseño para crear un módulo. También puede colocar un archivo de imagen en el diseño como imagen estática o arrastrar una hoja de estilos CSS hasta el Repositorio de estilos para crear una hoja de estilos externa.

Para más información sobre cómo trabajar con proyectos consulte el apartado Proyectos de StyleVision 123.

La ventana **Proyecto** (*imagen siguiente*) muestra el proyecto activo. Los comandos del menú **Proyecto** 802 afectan al proyecto activo. Para cambiar de proyecto activo puede crear un proyecto nuevo (**Proyecto | Proyecto nuevo** 803) o abrir un proyecto ya existente (**Proyecto | Abrir proyecto** 803). Estos dos comandos también están disponibles en la barra de herramientas de la ventana Proyecto. Se trata del primer y segundo icono de la barra de herramientas (*imagen siguiente*). Para guardar un proyecto nuevo y darle un nombre use el comando **Proyecto | Guardar proyecto** 803 (o el tercer botón de la barra de herramientas de la ventana Proyecto).



Al hacer clic en el botón **Agregar archivos** de la barra de herramientas (*el cuarto icono de la siguiente imagen*) aparece un menú contextual con varios comandos. Estos comandos sirven para añadir archivos y carpetas al proyecto activo y se describen en la Referencia del usuario 802. El comando **Agregar archivo activo y relacionados** es especialmente útil porque añade al proyecto todos los archivos relacionados con el archivo SPS activo. Por ejemplo, si abre el archivo `QuickStart.sps` y hace clic en este comando, se añaden al proyecto todos los archivos relacionados.

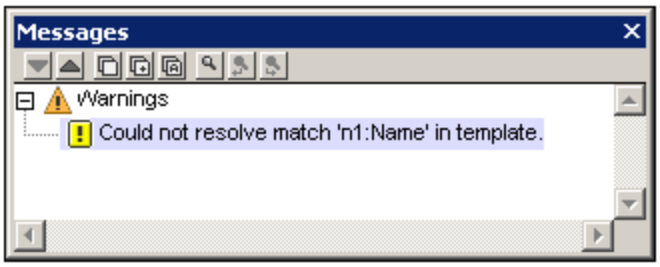


El comando **Propiedades del proyecto** (*quinto icono de la barra de herramientas*) abre el cuadro de diálogo "Propiedades" (i) del proyecto, (ii) de la carpeta seleccionada o (iii) del archivo seleccionado. Si selecciona el proyecto o un archivo y hace clic en **Propiedades**, aparece la ruta de acceso del archivo. El cuadro de diálogo "Propiedades" de una carpeta sirve para editar el nombre de la carpeta y especificar las extensiones de archivo que se deben asociar con esa carpeta. En la carpeta de proyecto solamente aparecen archivos con las extensiones de archivo asociadas. Para más información consulte el apartado Proyectos de StyleVision 123.

 © 2016-2022 Altova GmbH

2.2.8 Mensajes

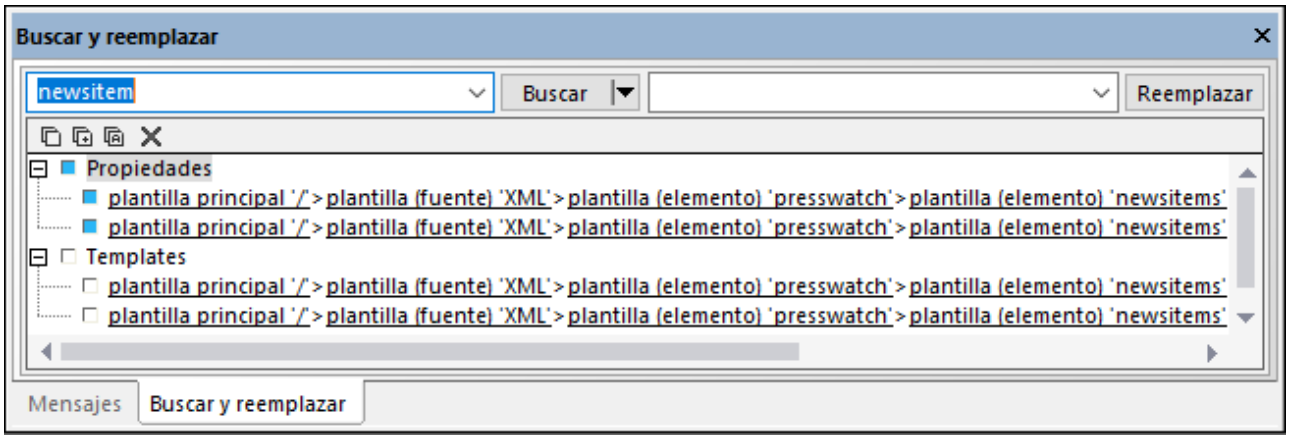
Al abrir StyleVision por primera vez, la barra lateral Mensajes (*imagen siguiente*) aparece bajo la ventana principal de la IGU. Para ocultar o ver esta barra lateral haga clic en Vista **View | Mensajes**.



En la barra lateral Mensajes puede ver las advertencias en la vista de diseño y vista Authentic. En la vista de diseño las advertencias se refieren al documento SPS y pueden ir de un archivo XML que falta a errores en la estructura del diseño. En vista Authentic las advertencias se refieren a la validez de los datos XML introducidos en relación con el esquema subyacente o conforme a otros criterios de validación.

2.2.9 Buscar y reemplazar

La barra lateral Buscar y reemplazar (*imagen siguiente*) permite buscar y reemplazar texto en la vista de diseño. Haga clic en la flecha del botón Buscar (resaltado en azul en la imagen siguiente) para visualizar y seleccionar las distintas opciones de búsqueda. Puede buscar texto, estilos, propiedades, variables, coincidencias de plantilla y expresiones XPath con cadenas de texto que puede introducir directamente en el campo *Buscar* o generar con expresiones regulares. Las búsquedas que realice en esta barra lateral se aplican a la vista de diseño. Los comandos de menú **Editar | Buscar** 794 y **Editar | Reemplazar** 794 activa esta barra lateral y coloca el cursor en el campo *Buscar* para que pueda realizar búsquedas en la vista de diseño. Los resultados de la búsqueda aparecen en la barra lateral. Puede hacer clic en los resultados para ir a la ubicación correspondiente del diseño. Para ocultar o ver la barra lateral Buscar y reemplazar haga clic en **Vista | Buscar y reemplazar**.



Para obtener información sobre la función de búsqueda en otras vistas (editor JavaScript, vista Authentic y hojas de estilos XSLT) consulte Buscar, Buscar siguiente, Reemplazar 794.

Buscar

Introduzca el término que quiera buscar en el campo *Buscar*. Después haga clic en el menú desplegable del botón **Buscar** (*resaltado en azul en la imagen anterior*) para configurar las opciones de búsqueda. Estas son las opciones que puede configurar:

- *Dónde buscar:* debe activar el elemento Incluir <*componente*> para que el componente correspondiente se incluya en la búsqueda.
- *Coincidencia de mayúsculas/minúsculas y/o palabras completas:* puede activar o desactivar estas opciones.
- *Expresiones regulares:* el texto que introduzca se entiende como expresión regular. Para aprender a usar expresiones regulares consulte Buscar, Buscar siguiente, Reemplazar 794.

Resultados

Los resultados están organizados en grupos en función del componente en el que aparece la cadena de texto de la búsqueda (*imagen anterior*). Cada uno de los elementos de los resultados aparece como una ruta de la jerarquía. Puede hacer clic en cualquiera de los enlaces de la jerarquía para ir al elemento correspondiente en la vista de diseño.

En el panel de resultados hay una barra de herramientas con iconos para estos comandos, empezando por la izquierda: copiar elementos o grupos de elementos en el portapapeles; borrar la información del panel de resultados.

Reemplazar

Después de que aparezcan los resultados puede seleccionar uno o varios de esos elementos para reemplazarlos. El elemento seleccionado se resalta con una marca azul (*imagen anterior*) y el botón **Reemplazar** se habilita. Introduzca la cadena de reemplazo en la caja de texto *Reemplazar* y haga clic en **Reemplazar**. Al reemplazar el elemento la marca pasa de ser azul a verde.

 © 2016-2022 Altova GmbH

3 Tutorial de introducción rápida

El objetivo de este tutorial es comprender los pasos clave del proceso de creación de diseños SPS. El tutorial explica cómo crear y configurar el diseño SPS, cómo insertar contenido, cómo aplicar formato a los componentes del SPS y cómo usar dos potentes características: los cálculos automáticos y las condiciones. Así aprenderá a estructurar sus documentos de salida de forma eficaz y a usar diferentes funciones estructurales y de presentación.

Archivos necesarios

Los archivos necesarios para hacer el tutorial de introducción rápida están en la carpeta (Mis) Documentos 34: `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\QuickStart`:

- `QuickStart.xsd`: archivo de esquema XML en el que se basa el archivo SPS.
- `QuickStart.xml`: archivo XML de trabajo que sirve de fuente de datos para la vista previa de resultados.
- `QuickStart.sps`: archivo SPS final. Puede comparar su archivo SPS con este archivo.
- `QuickStart.css`: hoja de estilos CSS externa utilizada en el tutorial.
- `NewsItems.BMP`: archivo de imagen utilizado en el SPS.

¿Cómo hacer el tutorial?

Recomendamos empezar el tutorial por el principio y avanzar por las diferentes secciones. También le recomendamos abrir los archivos XSD y XML antes de empezar el tutorial y mirar su estructura y contenido. Puede tener abiertos los archivos XSD y XML mientras hace el tutorial. Por último, guarde su archivo SPS con un nombre diferente a `QuickStart.sps` (p. ej. `MiTutorial.sps`) para no sobrescribir el archivo SPS que viene con el tutorial. Y no olvide guardar los cambios realizados en cada sección.

3.1 Crear un archivo SPS nuevo

En esta sección aprenderá a:

- Crear un documento SPS nuevo 68
- Añadir un esquema fuente para el archivo SPS 70
- Seleccionar la versión XSLT del archivo SPS 71
- Asignar el archivo XML de trabajo 71
- Especificar la codificación de salida 71
- Guardar el archivo SPS 72

Archivos de este apartado

Los archivos que se mencionan en este apartado se encuentran en la carpeta (Mis) Documentos 34, en `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\QuickStart`:

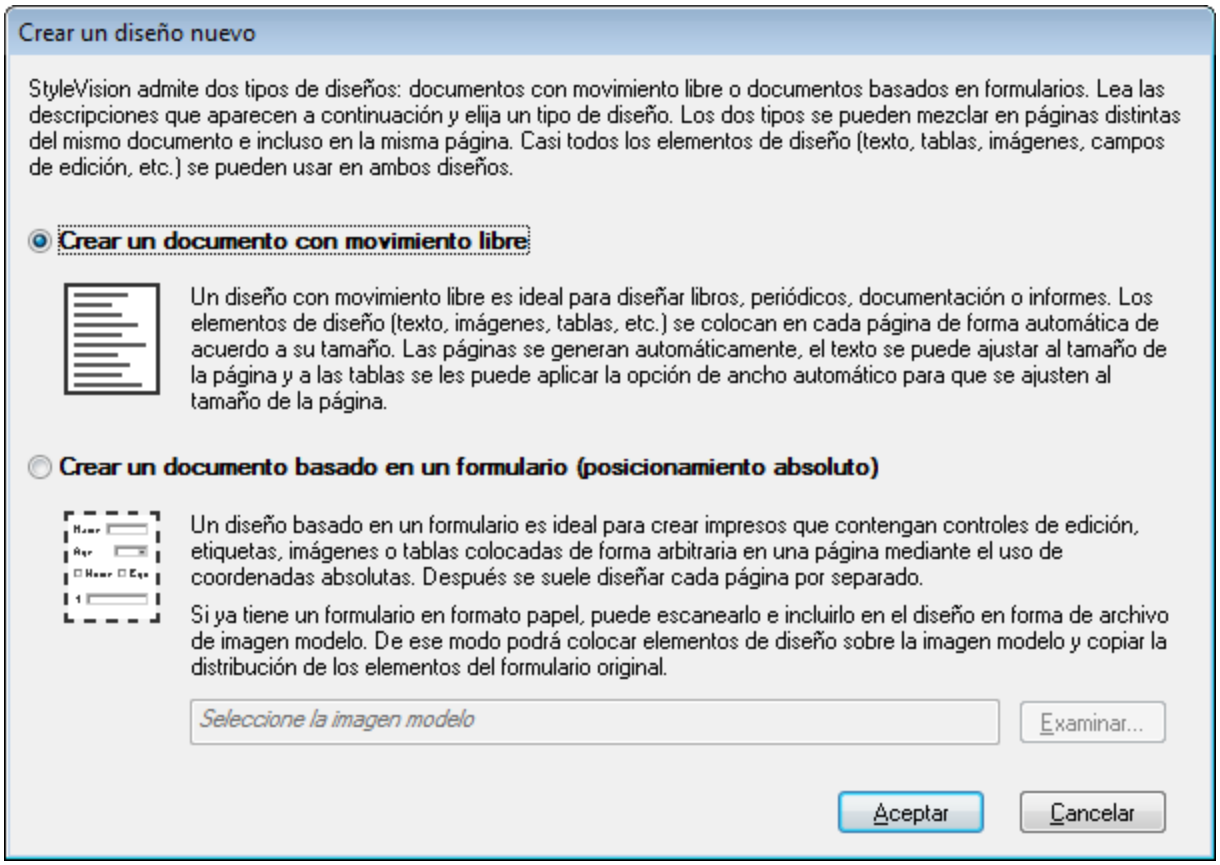
- `QuickStart.xsd`, el archivo de esquema XML en el que se basa el archivo SPS.
- `QuickStart.xml`, el archivo XML de trabajo origen de los datos que aparecen en las vistas previas de los resultados.
- `QuickStart.sps`, que es el archivo SPS final; puede comparar el archivo SPS creado por usted con este archivo.

Crear un documento SPS nuevo

Para crear un documento SPS nuevo haga clic en **Archivo | Nuevo | Archivo nuevo (vacío)** 761 o en el comando **Archivo nuevo (vacío)** de la lista desplegable del icono **Nuevo** 759 de la barra de herramientas de la aplicación. Aparece el cuadro de diálogo "Crear un diseño nuevo".

El cuadro de diálogo "Crear un diseño nuevo" (*imagen siguiente*) ofrece dos opciones. Puede crear: (i) un diseño de *documento con movimiento libre* o (ii) un diseño de *documento basado en un formulario*, en el que los componentes tienen posiciones absolutas (como en un programa de maquetación).

© 2016-2022 Altova GmbH

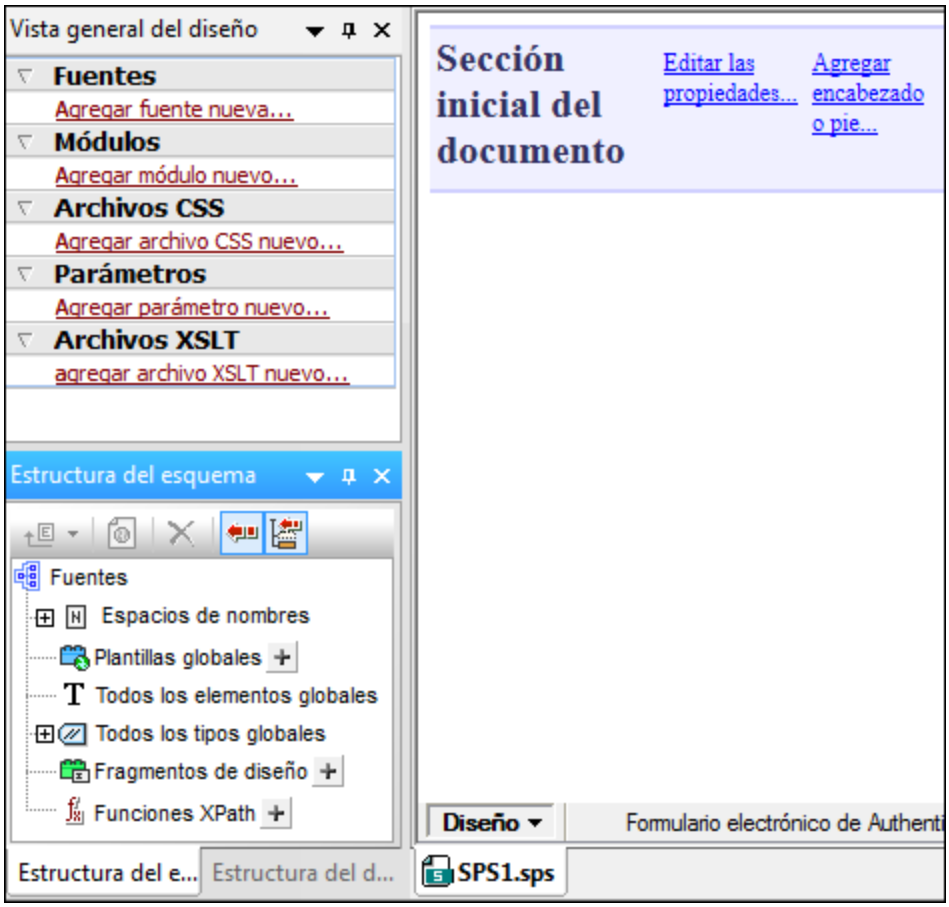


En los *documentos con movimiento libre*, el contenido se ajusta al formato de salida. Los elementos de contenido solamente se pueden colocar en relación a los demás y no con posición absoluta. Este tipo de diseño es ideal para la creación de informes, artículos y libros.

En los *documentos basados en formularios* se crea un solo contenedor de diseño 204, donde los componentes se pueden colocar con posición absoluta. Las dimensiones del contenedor de diseño las define el usuario, que puede colocar cuadros de diseño con posición absoluta dentro del contenedor de diseño. Después puede colocar el contenido del documento dentro de los cuadros de diseño. Si desea que el SPS sea igual que un formulario concreto, puede usar una imagen del formulario original como imagen modelo 204. La imagen modelo se puede insertar como imagen de fondo en el contenedor de diseño. La imagen modelo le ayudará a diseñar su formulario pero no se incluirá en los documentos de salida.

En este tutorial vamos a crear un documento con movimiento libre. Por tanto, seleccione el botón de opción Crear *un documento con movimiento libre* y haga clic en **Aceptar**.

En la vista 37 **Diseño** 37 (*imagen siguiente*) se crea un documento nuevo llamado `SPS1.sps`.

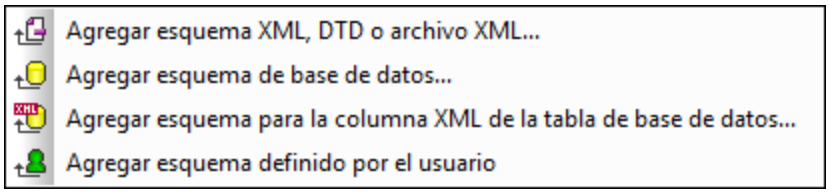


Y en la vista 37 **Diseño** 37 aparece una plantilla principal vacía. En las ventanas Vista general del diseño 45 y Estructura del esquema 48 no aparece ningún esquema.

Agregar un esquema fuente

Para este diseño SPS usaremos el esquema `QuickStart.xsd` como esquema fuente. Siga estos pasos para agregar este esquema como esquema fuente:

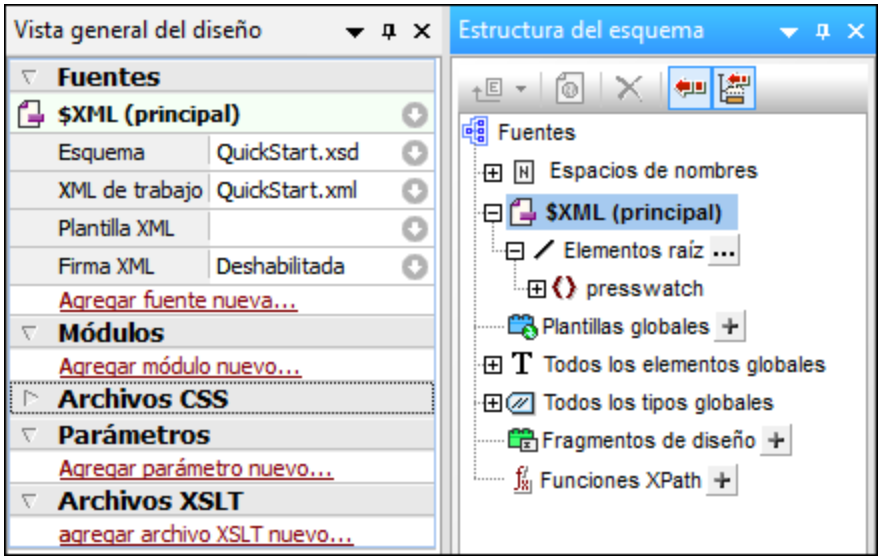
1. En la ventana Vista general del diseño, bajo la sección **Fuentes**, haga clic en el vínculo `Agregar fuente nueva...` (*imagen anterior*). Aparece un menú con varias opciones (*imagen siguiente*). Seleccione la opción **Agregar esquema XML, DTD o archivo XML...**.



2. Aparece el cuadro de diálogo "Abrir". Busque el archivo `QuickStart.xsd` en la carpeta (Mis) Documentos 68 y haga clic en **Abrir** (véase más arriba).
3. StyleVision le pide que seleccione un archivo XML de trabajo. Haga clic en el botón **Examinar** y busque el archivo `QuickStart.xsd` en la carpeta (Mis) Documentos 68. Después haga clic en **Abrir**.

 © 2016-2022 Altova GmbH

El esquema se añade como esquema fuente en la ventana Vista general del diseño y en la ventana Estructura del esquema (*imagen siguiente*). En la ventana Vista general del diseño, el archivo XML de trabajo se asigna al esquema fuente.



No olvide que: (i) en la ventana Vista general del diseño, la entrada `$XML` enumera el esquema y el archivo XML de trabajo [30] y el archivo de plantilla XML [30]; (ii) en la ventana Estructura del esquema, la entrada `Elementos raíz` enumera todos los elementos raíz (elementos de documento) [30] que seleccione de los elementos globales [31] definidos en el esquema. En el ejemplo que nos ocupa, el elemento `presswatch` está seleccionado por defecto porque es el único elemento global [31] del esquema que está en el nivel superior de la jerarquía definida en el esquema; (iii) todos los elementos globales [31] del esquema se enumeran bajo la entrada `Todos los elementos globales` [48].

Seleccionar la versión XSLT

Para este tutorial usaremos la versión XSLT 2.0. Para especificar la versión XSLT, haga clic en el icono de la barra de herramientas de la aplicación.

Asignar un archivo XML de trabajo o cambiar de archivo XML de trabajo

Además de añadir el esquema XML al SPS, en el paso anterior también asignó un archivo XML de trabajo [30] al esquema. Para asignar un archivo XML de trabajo [30], cambiar de archivo XML de trabajo o quitar la asignación, haga clic con el botón derecho en la línea del archivo XML de trabajo en la ventana Vista general del diseño o haga clic en el icono del archivo XML de trabajo. En el menú que aparece seleccione el comando correspondiente. El archivo XML de trabajo [30] se asigna y el nombre de archivo aparece en la ventana Vista general del diseño. Antes de seguir, compruebe que se asignó correctamente el archivo XML de trabajo `QuickStart.xml`, disponible en la carpeta (Mis) Documentos [34].

Especificar la codificación de salida

En la pestaña *Codificación predeterminada* del cuadro de diálogo "Opciones" (**Herramientas | Opciones** [894]), elija la codificación HTML `Unicode UTF-8` y la codificación RTF `UTF-8`.

Guardar el documento SPS

Cuando termine de configurar el archivo SPS siguiendo las instrucciones anteriores, guárdelo como `MiTutorial.sps` en la carpeta (Mis) Documentos [68] `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\QuickStart`. Para ello use el comando de menú **Archivo | Guardar el diseño** [773] o pulse **Ctrl+S**. Aparece el cuadro de diálogo "Guardar el diseño". Elija la opción *Guardar como archivo SPS*, escriba el nombre del archivo y haga clic en **Aceptar**.

© 2016-2022 Altova GmbH

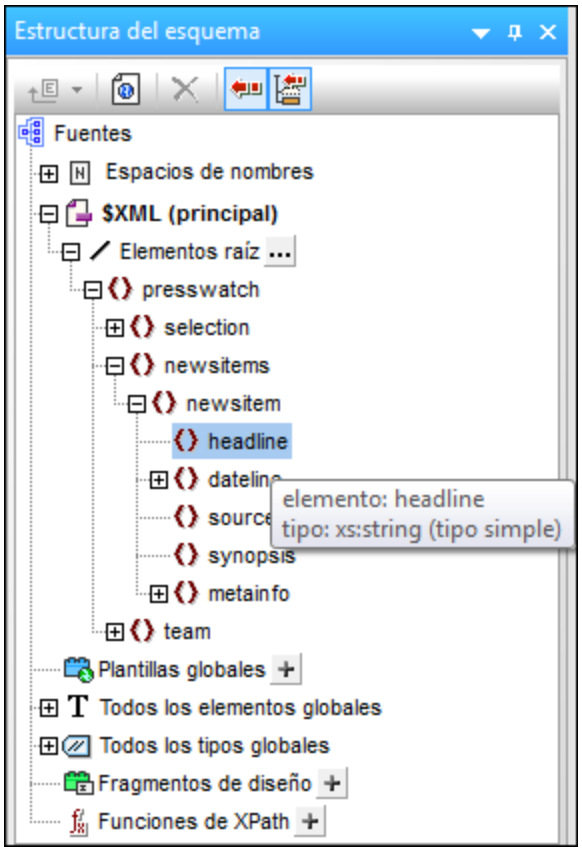
3.2 Insertar contenido dinámico (de una fuente de datos XML)

Esta sección presenta los mecanismos disponibles para insertar datos desde nodos del documento XML. Aprenderá a arrastrar nodos de elemento y atributo desde la estructura del esquema hasta el diseño y crear contenido a partir de dichos nodos. Cuando se crea contenido a partir de un nodo, los datos del nodo se representan como cadena de texto, es decir, se concatena el contenido de los nodos de texto secundarios del elemento y los nodos de texto de todos sus elementos descendientes.

Insertar el contenido de elementos

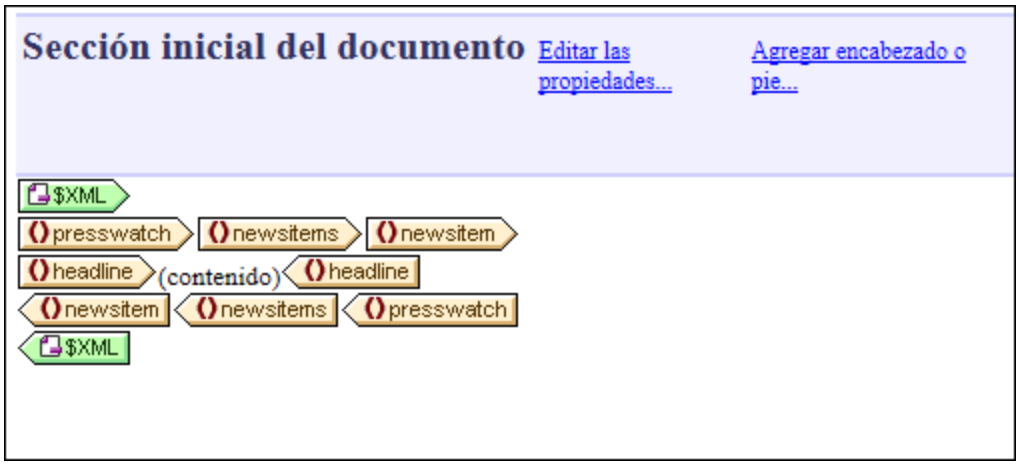
Siga estos pasos para insertar contenido en el diseño SPS a partir de elementos:

1. En la ventana Estructura del esquema 48 expanda el árbol hasta ver los secundarios del elemento `newsitem` (*imagen siguiente*).

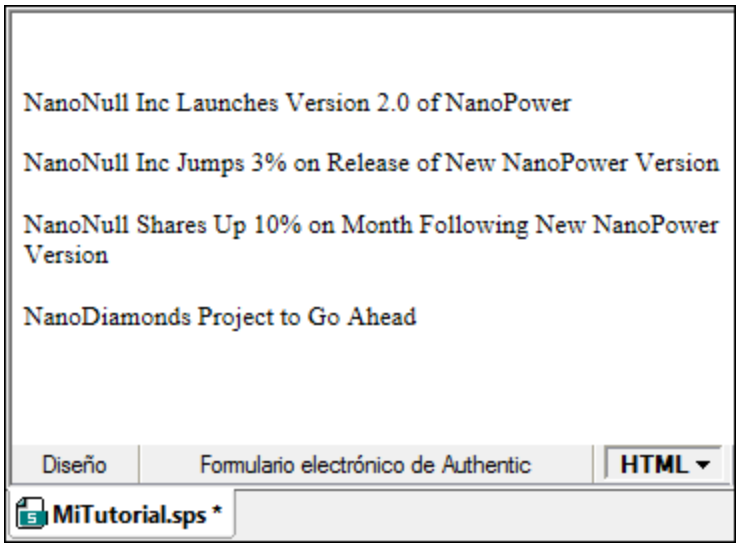


2. Seleccione el elemento `headline` (observe que el tipo de datos del elemento aparece al pasar el cursor por encima, *imagen anterior*). Arrastre el elemento hasta la vista 41 **Diseño** 41 y, cuando el puntero se convierta en punto de inserción, suelte el elemento en la plantilla principal.
3. Al soltar el elemento aparece un menú contextual. Seleccione la opción **Crear contenido**. Las etiquetas de apertura y cierre del elemento `headline` se insertan en el punto donde soltó el elemento y contienen el marcador de posición `(contenido)`. Las etiquetas del elemento `headline` están rodeadas por las etiquetas de apertura y cierre de los elementos antecesores de `headline` (*imagen siguiente*).

4. Ponga los elementos en diferentes líneas (pulsando la tecla **Entrar**) tal y como muestra la imagen siguiente.



Haga clic en la pestaña **HTML** para ver la vista previa en formato HTML [39] (*imagen siguiente*). La vista previa HTML muestra el contenido de los elementos secundarios `headline` del elemento `newsitem`, cada una en forma de cadena de texto.



Mire también la vista previa de la vista [38] **Authentic** [38] y el resultado **RTF**.

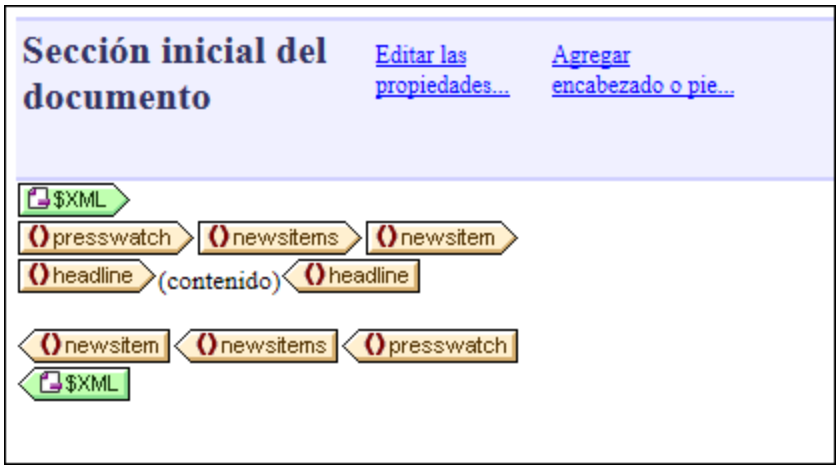
Nota: también puede crear el contenido de un nodo de esta forma: (i) Haga clic en el icono **Insertar contenido** de la barra de herramientas [755] **Insertar elemento de diseño** [755], (ii) haga clic en la posición del diseño donde desea insertar el contenido y (iii) en el cuadro de diálogo que aparece seleccione el nodo que desea usar para crear el contenido.

Insertar contenido de atributos

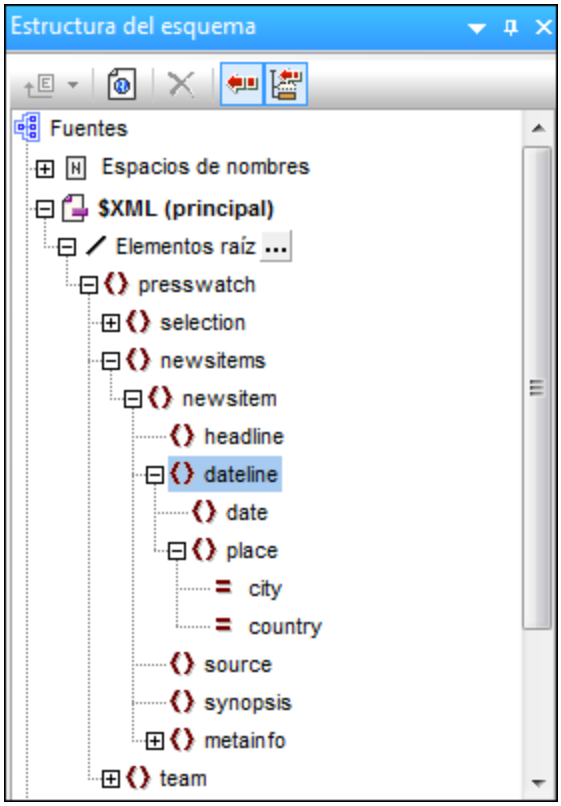
Cuando se crea contenido en el diseño a partir de elementos, el contenido de sus atributos no se inserta automáticamente. Es necesario arrastrar el nodo de atributo hasta el diseño para que el valor del atributo se incluya en los resultados. Siga estas instrucciones:

 © 2016-2022 Altova GmbH

1. Ponga el cursor después de la etiqueta de cierre del elemento `headline` y pulse **Entrar**. Esto inserta una línea nueva (*imagen siguiente*).

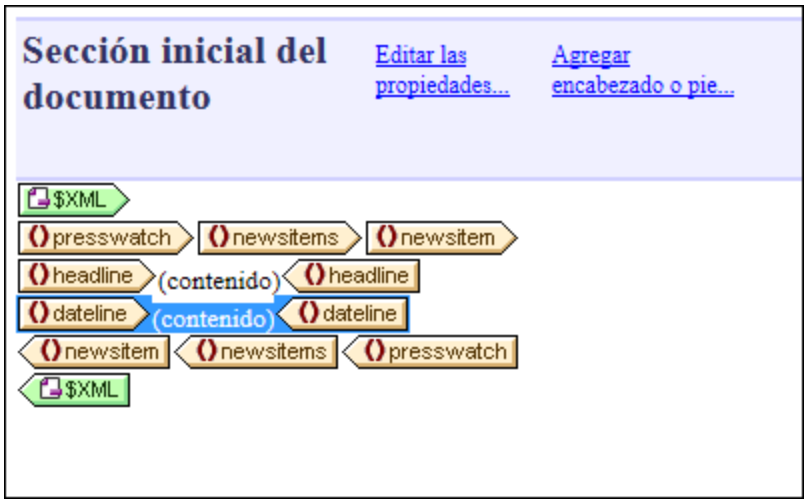


2. En la ventana Estructura del esquema, expanda el elemento `dateline` (*imagen siguiente*).

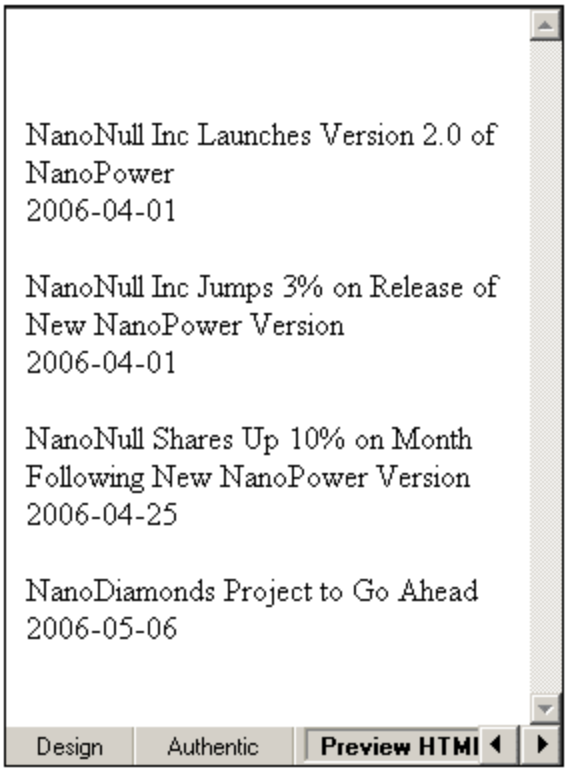


Observe que el elemento `dateline` tiene dos elementos secundarios (`date` y `place`) y que el elemento `place` tiene dos atributos: `city` y `country`.

3. Arrastre el elemento `dateline` hasta el diseño y suéltelo al principio de la línea vacía recién creada (*imagen siguiente*).



4. Cambie a la vista previa en HTML 39 y observe con atención el resultado generado por el elemento `dateline` (*imagen siguiente*).

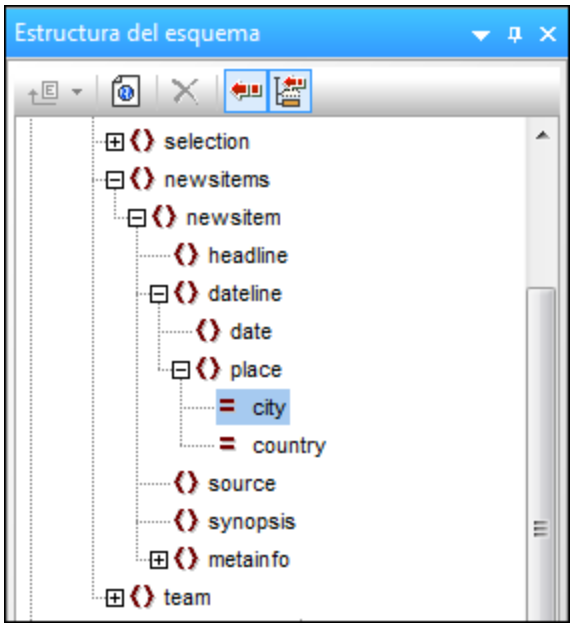


Observe que aunque el resultado incluye el contenido de los secundarios `date` de los elementos `dateline`, no incluye el contenido de los secundarios `place` de `dateline`. Esto se debe a que los datos de `place` están en los atributos del elemento `place` (en los atributos `city` y `country`) y el contenido de los atributos no se incluye en el resultado cuando se procesa el elemento primario del atributo.

5. En la vista **Diseño**, haga clic en el comando de menú **Authentic | Agregar selector de fecha automáticamente** 855 para desactivar la inserción automática de selectores de fecha 492. (El icono no tiene resaltado el contorno cuando está desactivado.) Este paso es necesario si no quiere que se

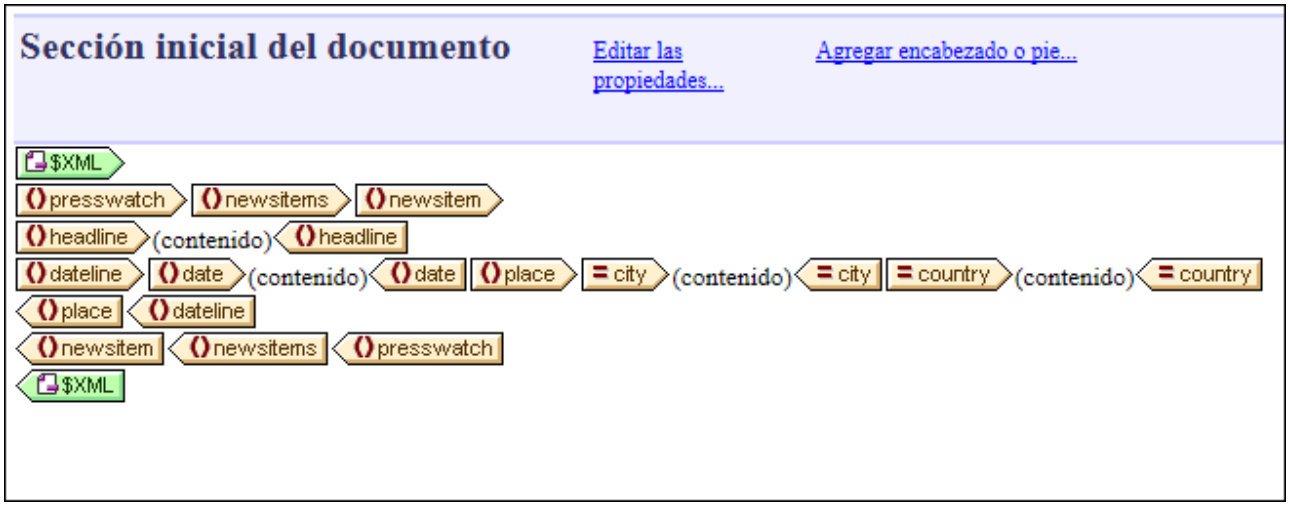
inserte un selector de fecha automáticamente cuando inserte nodos de tipo `xs:date` o `xs:dateTime` en el diseño (que es lo que haremos en el paso siguiente). Arrastre el elemento `date` desde la ventana Estructura del esquema 48 y suéltelo (creando contenido) entre las etiquetas de apertura y cierre del elemento `dateline`.

6. Seleccione el atributo `city` del elemento `dateline/place` (*imagen siguiente*) en la ventana Estructura del esquema 48.



7. Arrastre el nodo de atributo `@city` hasta la vista 37 **Diseño** 37 y suéltelo (creando contenido) justo después de la etiqueta final del elemento `date`.
8. Arrastre el nodo de atributo `@country` hasta la vista 37 **Diseño** 37 y suéltelo (creando contenido) justo después de la etiqueta final del atributo `@city`.

Cuando termine, el SPS tendrá este aspecto:

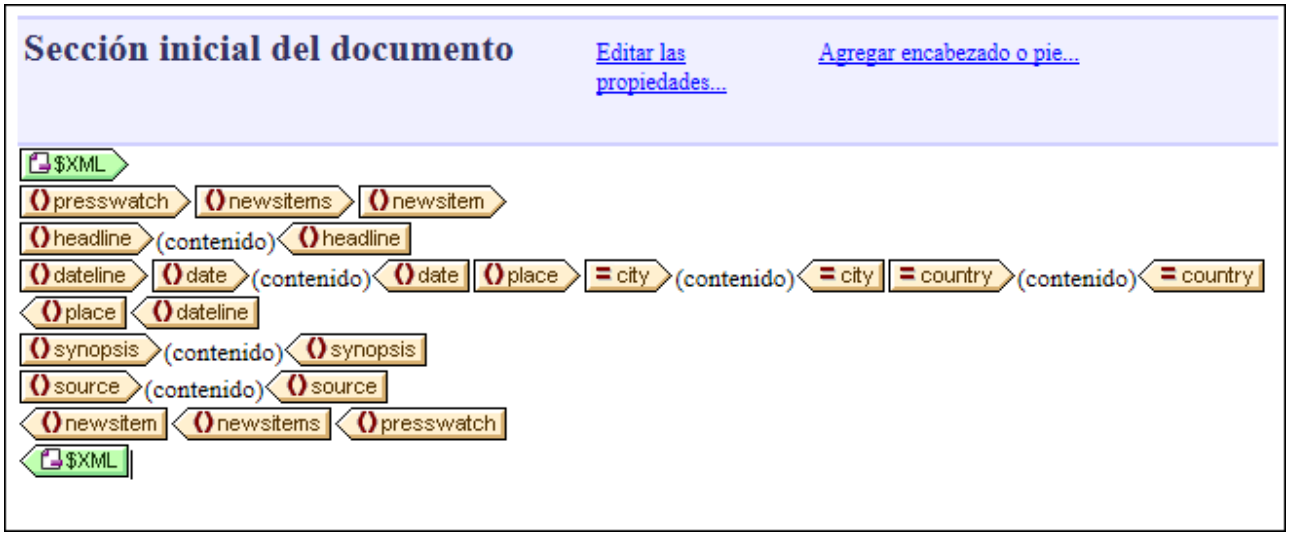


Y esta es la vista previa en HTML 39 del diseño:

Observe que los valores de los atributos `@city` y `@country` ahora están en el resultado.

Agregar más contenido dinámico

El contenido de los elementos y atributos de la fuente de datos XML se puede insertar en cualquier parte del diseño usando el método que acabamos de describir. Para terminar este apartado del tutorial, añada los elementos `synopsis` y `source` en el diseño hasta conseguir este aspecto:



Observe que el elemento `synopsis` se colocó antes del elemento `source`, lo cual se corresponde con el orden de los elementos en el esquema. Después de añadir los elementos `synopsis` y `source` en el diseño, mire la

 © 2016-2022 Altova GmbH

vista previa HTML 39 para ver el resultado. Es importante tener en cuenta que el orden que tienen los nodos en la plantilla principal 32 es el orden que tendrán los documentos de salida (consulte el apartado Plantillas y fragmentos de diseño 265 para obtener más información).

Otro aspecto importante es con qué forma se crea un nodo en el diseño. En la vista previa HTML 39 puede observar que todos los nodos incluidos en el diseño se enviaron al documento de salida como cadenas de texto. Sin embargo, los nodos también se pueden representar en forma de tabla o de cuadro combinado, por ejemplo. En este apartado, especificó como formato de salida el formato cadena de texto porque al insertar los nodos seleccionó la opción **Insertar contenido**. En el apartado Usar condiciones 98 aprenderá a crear cuadros combinados a partir de nodos. En el apartado Usar plantillas globales y el resto de contenidos 106 aprenderá a crear tablas (dinámicas) a partir de nodos.

Guarde el archivo de diseño antes de pasar al apartado siguiente.

3.3 Insertar contenido estático

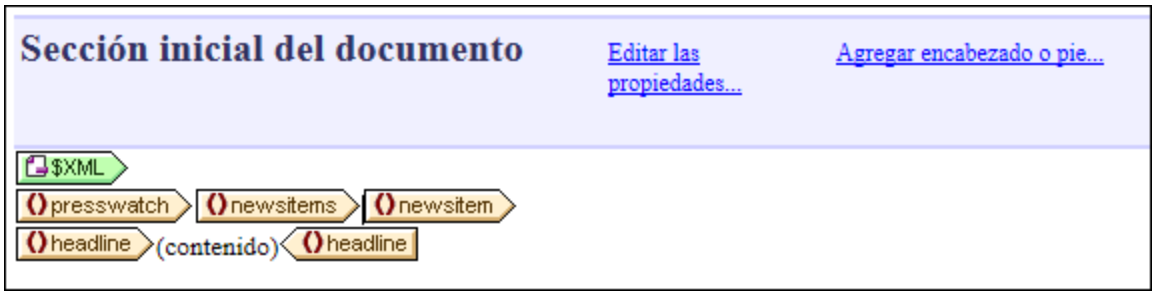
El contenido estático es el contenido que el usuario introduce/inserta en el diseño directamente (a diferencia del contenido dinámico, que procede del archivo XML de origen). En el diseño SPS puede poner una gran variedad de componentes estáticos. En esta parte del tutorial aprenderá a insertar estos componentes estáticos:

- Imágenes 80
- Líneas horizontales 82
- Texto 82

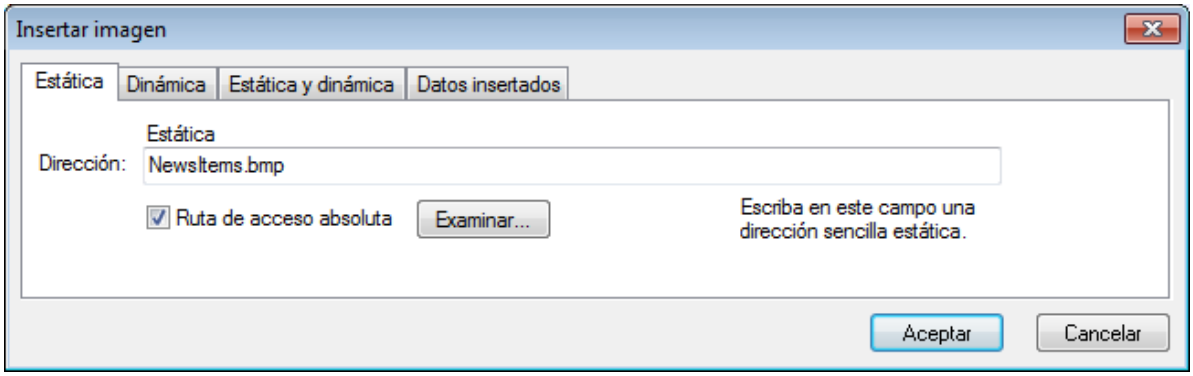
Insertar imágenes estáticas

La imagen estática que vamos a insertar está en la carpeta (Mis) Documentos 34: `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\QuickStart\NewsItems.BMP`. El objetivo es usar esta imagen como cabecera del documento. Para insertar esta imagen como cabecera del documento:

1. Ponga el cursor entre las etiquetas de inicio de los elementos `newsitems` y `newsitem` (*imagen siguiente*).

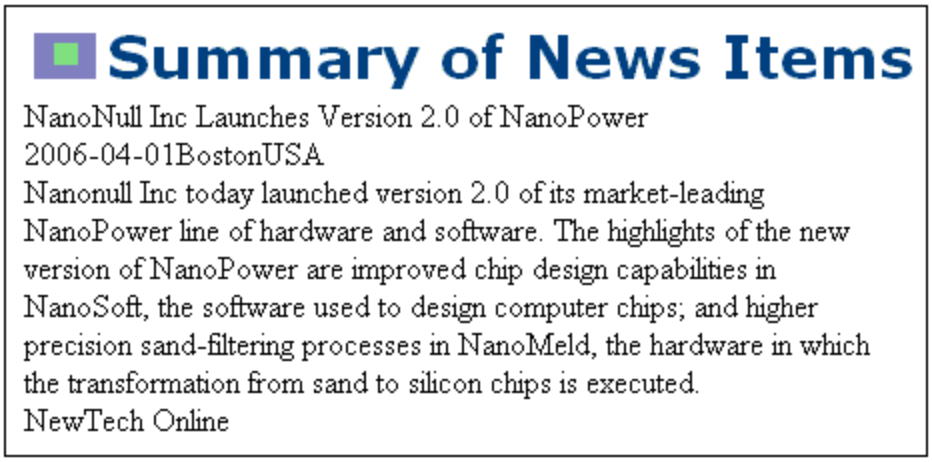


   Observe que el cursor está dentro del elemento `newsitems` pero fuera del elemento `newsitem`. Por tanto, en el documento de salida la imagen se incluirá una sola vez, al principio del procesamiento del elemento `newsitems` (porque en el esquema solamente se define un elemento `newsitems`).
2. Haga clic con el botón derecho y seleccione el comando **Insertar | Imagen** 820. Aparece el cuadro de diálogo "Insertar imagen" (*imagen siguiente*).



3. En la pestaña *Estática*, marque la casilla *Ruta de acceso absoluta* y haga clic en el botón **Examinar**. Busque el archivo `NewsItems.BMP` y selecciónelo.
4. Para terminar haga clic en **Aceptar**.

La vista previa HTML del documento tendría este aspecto:



Insertar líneas horizontales

Vamos a insertar dos líneas horizontales. Inserte la primera línea horizontal entre la cabecera y el cuerpo del documento:

1. Ponga el cursor justo después de la imagen estática que acaba de insertar.
2. Haga clic con el botón derecho y seleccione el comando **Insertar | Línea horizontal** 822. Se inserta una línea horizontal.

Defina estas propiedades para la línea horizontal:

1. Seleccione la línea en la vista 37 **Diseño** 37. En la ventana Propiedades 59 seleccione el componente **`línea`** (en la columna **Propiedades para**) y después seleccione el grupo de propiedades `HTML`.
2. Asigne las propiedades `color` y `size` para la línea.
3. Seleccione la línea en la vista 37 **Diseño** 37. En la ventana Estilos 58 seleccione el componente línea (en la columna **Estilos para**) y después seleccione el grupo de propiedades `cuadro`.
4. Defina una propiedad `margin | bottom` con el valor `12pt`.
5. Mire el resultado en la vista previa 39 **HTML** 39.

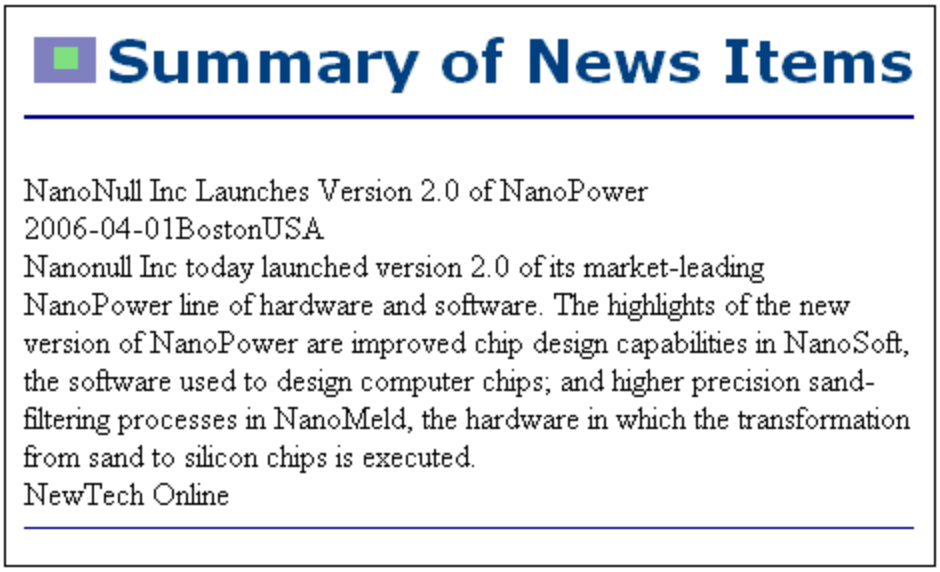
Ahora inserte la segunda línea horizontal al final de cada elemento `newsitem`. Es decir, ponga el cursor justo después de la etiqueta de cierre del elemento `newsitem`. Puede cambiar el grosor de la línea cambiando el valor de la propiedad `size` (en la ventana Propiedades seleccione la propiedad `HTML | size` con valor `3`, por ejemplo).

Insertar texto estático

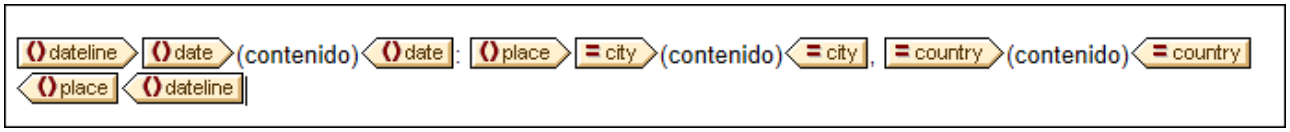
Observe que ya insertó texto estático en el diseño. Cuando pulsó la tecla **Entrar** para crear líneas nuevas (en el apartado anterior Insertar contenido dinámico 73), se añadieron espacios en blanco (texto estático) en el diseño. En el siguiente paso añadiremos unos cuantos caracteres de texto estático.

Por ahora el diseño SPS tiene este aspecto en la vista previa HTML:

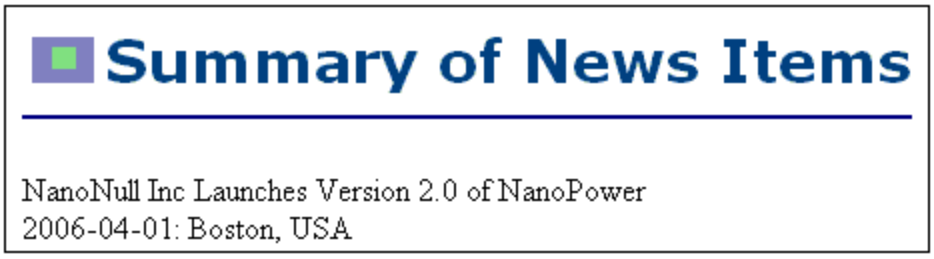
 © 2016-2022 Altova GmbH



Observe que en el resultado del elemento `dateline` el contenido del elemento `date` y los atributos `place/@city` y `place/@country` aparecen uno junto al otro sin espacios. Podemos añadir los espacios como texto estático: ponga el cursor después del elemento `date` y escriba dos puntos (:) y un espacio. Ahora escriba una coma y un espacio después del atributo `@city` (*imagen siguiente*).

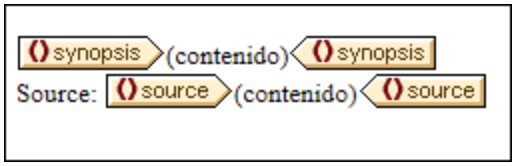


Esta parte del diseño tendría este aspecto en la vista previa HTML:



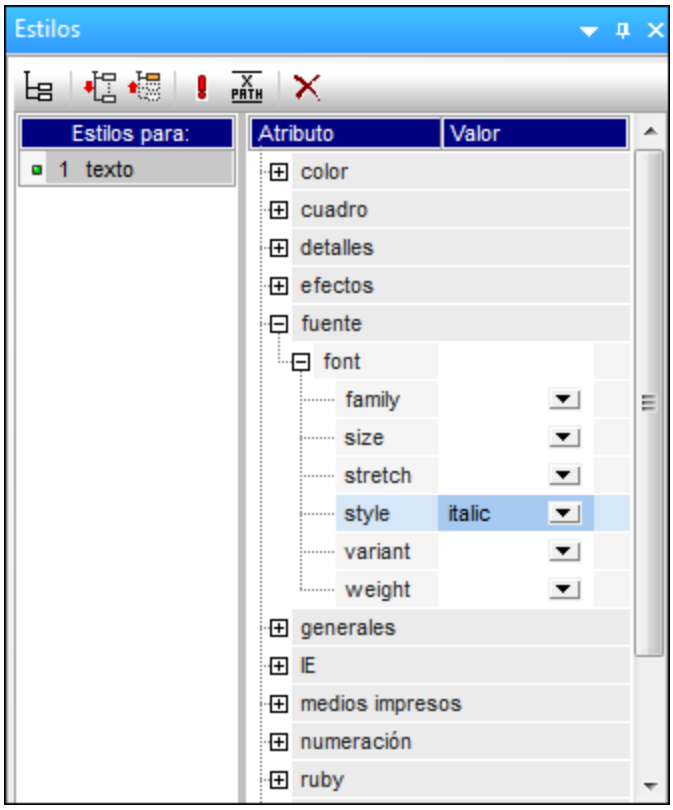
Observe los dos puntos, el espacio y la coma que aparecen en el elemento `dateline`. Todos estos caracteres son elementos de texto estático introducidos directamente en el diseño.

Ahora añada un elemento de texto estático nuevo: escriba la cadena de texto "`Source:` " justo antes de la etiqueta de inicio del elemento `source` (*imagen siguiente*).

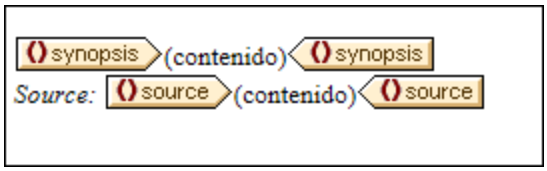


Aplicar formato al texto estático

Para aplicar formato al texto estático, seleccione el texto y defina propiedades de estilo local. Por ejemplo, seleccione la cadena de texto "`Source:`" que acaba de insertar. Observe que en la ventana Estilos 58 (*imagen siguiente*) está seleccionado el componente `1 texto`. Expanda el grupo de propiedades `fuente`, seleccione la propiedad `font | style` y después elija el valor `italic` de la lista desplegable.

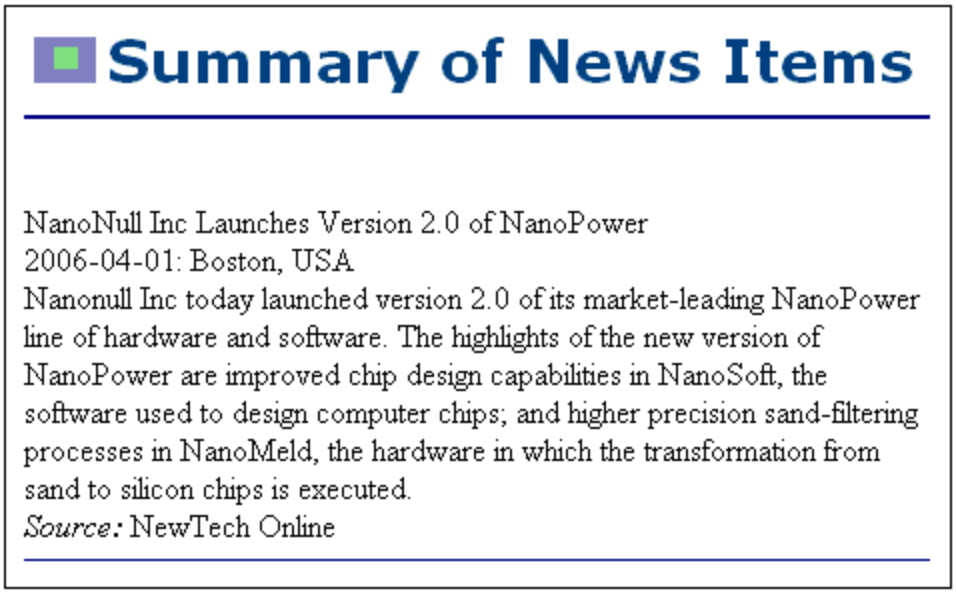


Ahora el texto estático (es decir, la cadena "`Source:`") aparece en cursivas en el diseño:



Ahora la vista previa HTML del diseño tiene este aspecto:

 © 2016-2022 Altova GmbH



Si le parece que no hay suficiente espacio vertical entre el elemento `source` y la línea horizontal que separa los dos elementos `newsitem`, vuelva a la vista **Diseño** e inserte una línea en blanco entre el elemento `source` y la línea horizontal (pulsando la tecla **Entrar**).

Después guarde el archivo.

En este apartado aprendió a insertar contenido estático y a aplicarle formato. En el apartado siguiente aprenderá más cosas sobre cómo aplicar formato a los componentes del diseño con ayuda de principios y propiedades CSS.

3.4 Aplicar formato al contenido

StyleVision ofrece un mecanismo de aplicación de estilos 390, potente y a la vez flexible basado en CSS. Este mecanismo de aplicación de estilos tiene estas características:

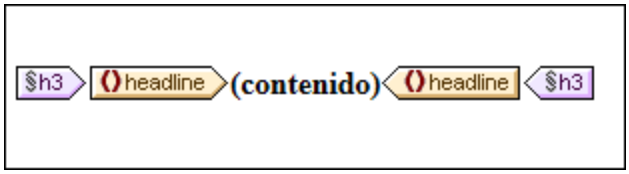
- Es posible definir reglas de estilo CSS tanto para componentes en bloque como para componentes inline.
- Los formatos predefinidos 375 son componentes de bloque que tienen estilos propios y que se pueden usar como contenedores para grupos de componentes que deben tratarse como un solo bloque. Los estilos inherentes de dichos formatos predefinidos se pueden reemplazar con estilos definidos localmente en cada componente. Este mecanismo es similar al principio de estilos en cascada de CSS.
- Es posible declarar atributos de clase en los componentes del diseño y usar la clase como selector de reglas de estilo externas 391 o globales 395.
- Puede especificar estilos en tres niveles. Estos niveles son, en orden ascendente de prioridad: (i) reglas de estilo de hojas de estilos externas 391, (ii) reglas de estilo globales 395 y (iii) reglas de estilo locales 398. No obstante, tenga en cuenta que ciertos tipos de selectores de las reglas de estilo externas y globales (como los selectores basados en nombres `h1`, `a`, `img`, etc.) solamente se aplican en la vista **Authentic** y en el formato de salida HTML, no en RTF. Las reglas que tienen selectores de clase se aplican a formatos de salida HTML, RTF, PDF y Word 2007+.

En este apartado aprenderá a:

- Asignar formatos predefinidos 86
- Asignar un atributo de clase a un componente 87
- Definir estilos en una hoja de estilos CSS externa 88 y añadir esta hoja de estilos al Repositorio de estilos del diseño SPS
- Definir reglas de estilo globales 89
- Definir estilos locales para un grupo de componentes de diseño 91
- Definir estilos locales para un solo componente 91

Asignar formatos predefinidos

Podemos asignar un formato predefinido 375 a un componente para darle el estilo inherente del formato predefinido 375. En la vista **Diseño** seleccione el elemento `headline` y después seleccione el comando de menú **Incluir en | Párrafo especial | Título de tercer nivel (h3)** (o utilice el cuadro combinado **Formatos predefinidos** de la barra de herramientas). Las etiquetas de formato predefinido se crean alrededor del elemento `headline` (*imagen siguiente*).

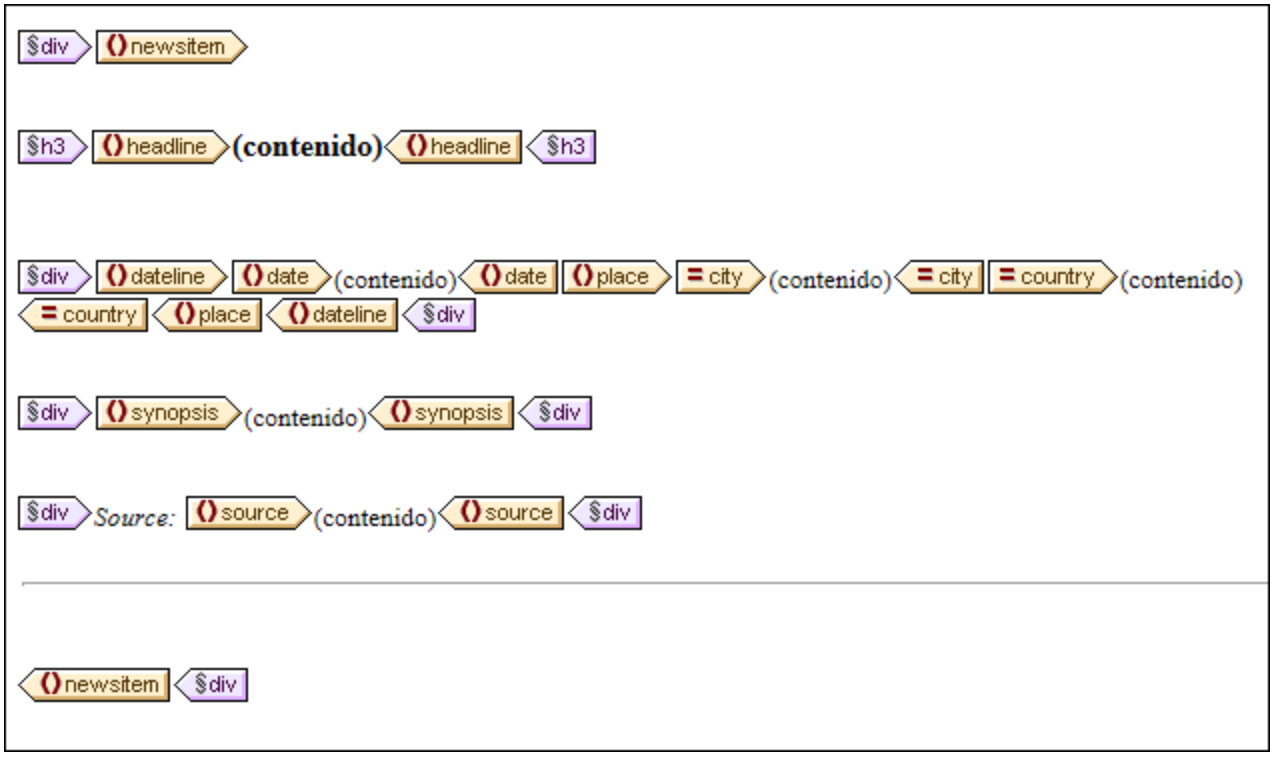


Observe cómo cambian las propiedades `font` del contenido y que se añade espacio vertical antes y después del formato predefinido. Estos valores son inherentes al formato predefinido `h3` (título de tercer nivel).

También puede usar formatos predefinidos para agrupar componentes de diseño y aplicarles formato en bloque o asignarles propiedades inline a todo el grupo de componentes. El formato predefinido más práctico en este caso es el formato `div`, que crea un bloque sin espacios adicionales. En el diseño SPS asigne el formato

 © 2016-2022 Altova GmbH

predefinido `div` (contenedor genérico de bloque) a los nodos `newsitem`, `dateline`, `synopsis` y `source` (*imagen siguiente*). Observe que el texto estático "`Source: `" también se incluye en el componente `div` que incluye al elemento `source` y que todo el elemento `newsitem` está incluido en un componente `div`.

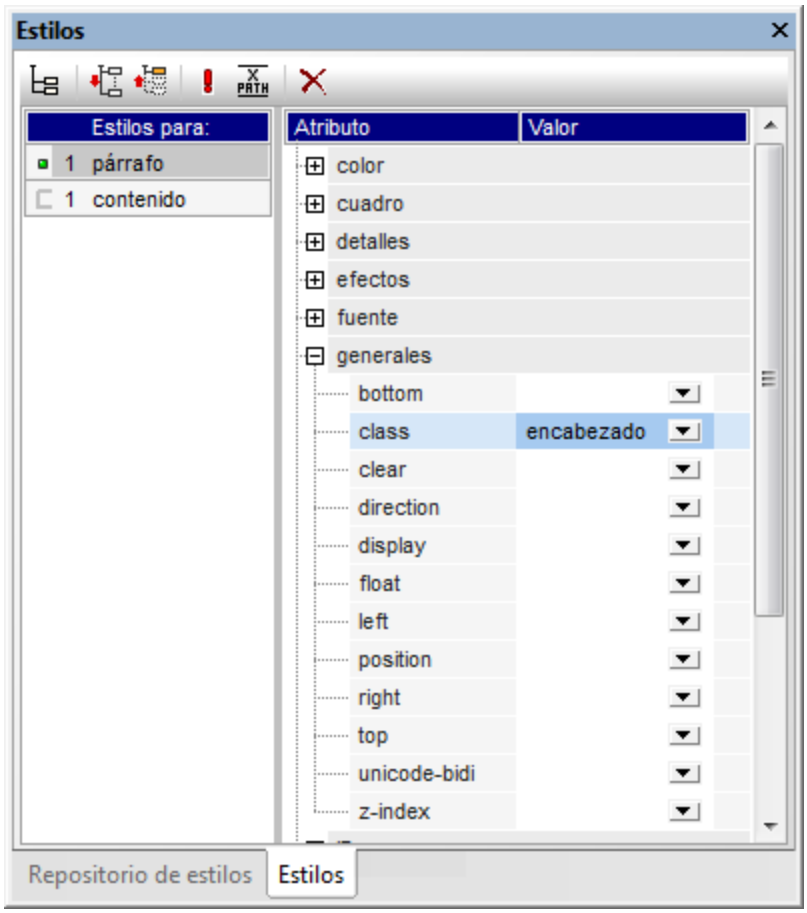


Bien, ya terminó de agrupar componentes en bloques `div`. Más adelante 91 aprenderá a asignar estilos a estos bloques de componentes.

Asignar atributos de clase a componentes

El mecanismo de StyleVision permite definir una regla de estilo para una clase de componentes. Por ejemplo, puede definir un conjunto de propiedades comunes para todos los encabezados (por ejemplo, una familia de fuentes y un color concretos). Para ello es necesario dos cosas: (i) asignar los componentes que deben tener un conjunto de propiedades comunes a una clase y (ii) definir las propiedades de estilo de dicha clase.

En la vista **Diseño** seleccione la etiqueta `h3`. En la ventana Estilos seleccione **`1 párrafo`** (para seleccionar el formato predefinido) y el grupo de propiedades `generales`. Expanda el grupo de propiedades `generales` y después haga doble clic en el campo *Valor* de la propiedad `class` y escriba el texto `encabezado` en el campo de valor.



Esta instancia concreta del formato predefinido `h3` tiene asignada una clase llamada `encabezado`. Cuando defina propiedades de estilo para la clase `encabezado` (estilos de una hoja de estilos externa o estilos SPS globales), estas propiedades se aplicarán a todos los componentes del diseño SPS que sean de clase `encabezado`.

Agregar una hoja de estilos CSS externa al repositorio de estilos

Las reglas de estilo de hojas de estilos CSS externas se pueden aplicar a componentes del diseño SPS. Sin embargo, antes es necesario añadir las hojas de estilos externas al repositorio de estilos. De lo contrario no podremos aplicar sus reglas a los componentes de nuestro diseño. En la ventana Repositorio de estilos 55 (de la vista **Diseño**):

1. Seleccione la entrada **`Externos`**.
2. Haga clic en el botón **Agregar** de la barra de herramientas de la ventana Repositorio de estilos 55. Aparece el cuadro de diálogo "Abrir".
3. Busque el archivo `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\QuickStart\QuickStart.css`, que está en la carpeta (Mis) Documentos 34, y haga clic en **Abrir**.

La hoja de estilos se añade al repositorio de estilos y contiene, entre otras, estas reglas:

```
.encabezado   {
                    font-family: "Arial", sans-serif;
                    font-weight: bold;
```

 © 2016-2022 Altova GmbH

```
                    color: red;
                   }

h3                 {
                    font-size: 12pt;
                   }
```

Las reglas de estilo para la clase `encabezado` y el elemento `h3` se agrupan y generan este resultado HTML para el elemento `headline`.

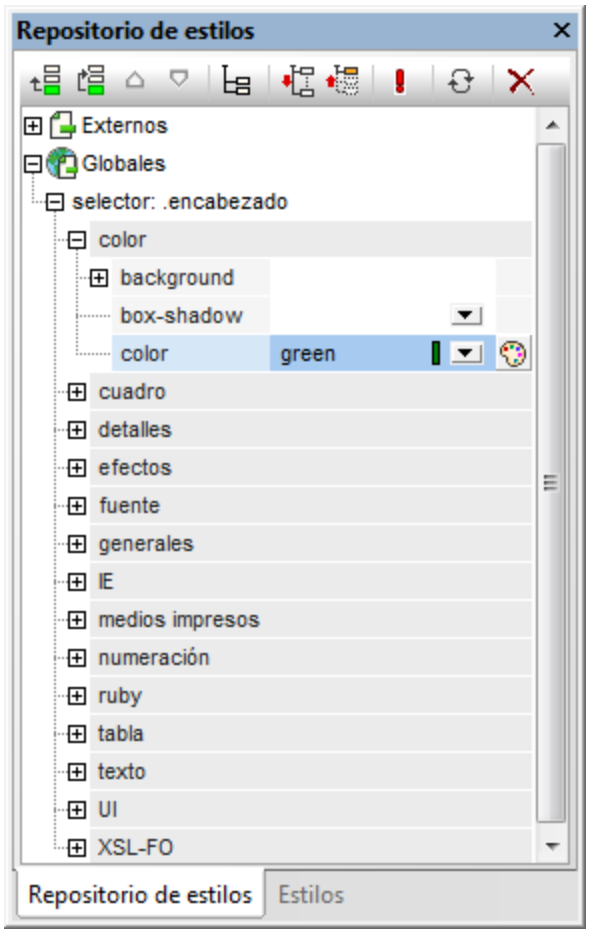
NanoNull Inc Launches Version 2.0 of NanoPower

2006-04-01: Boston, USA
Nanonull Inc today launched version 2.0 of its market-leading NanoPower line of hardware and software. The highlights of the new version of NanoPower are improved chip design capabilities in NanoSoft, the software used to design computer chips; and higher precision sand-filtering processes in NanoMeld, the hardware in which the transformation from sand to silicon chips is executed.
Source: NewTech Online

Definir reglas de estilo globales

El mecanismo de StyleVision permite definir reglas de estilo globales 395 para todo el diseño SPS mediante el uso de selectores CSS. Las reglas se definen en la ventana Repositorio de estilos 55 directamente. Para crear una regla de estilo global para la clase `header`:

1. En la ventana Repositorio de estilos 37 de la vista Diseño 37 seleccione la entrada **`Globales`**.
2. Haga clic en el botón **Agregar** de la barra de herramientas de la ventana. Se crea una regla vacía para el selector comodín (`*`), que aparece resaltado.
3. Escriba `.encabezado` en lugar del selector comodín.
4. Expanda el grupo de propiedades `color` y seleccione el valor `green` en la lista desplegable (*imagen siguiente*).



Si una regla de estilo global define una propiedad que también está definida en la hoja de estilos externa (en este caso la propiedad `color`), el valor de la propiedad de la regla global tiene prioridad sobre la de la hoja de estilos externa. En la vista previa HTML puede ver que el contenido del titular aparece ahora en color verde. Las demás definiciones de propiedad de la hoja de estilos (las que no fueron reemplazadas por propiedades de la regla de estilo global) se conservan: `font-family` y `font-weight`.

Nota: puesto que la regla de estilo global usa el selector de clase, esta regla también se aplica a formatos de salida RTF además de a la vista **Authentic** y a formatos de salida HTML.

 © 2016-2022 Altova GmbH

Definir de una vez estilos ocales para varios componentes

El mecanismo de StyleVision permite definir estilos locales para varios componentes de una sola vez. En el diseño puede especificar que el contenido de texto de los elementos `newsitem` tenga la fuente `Arial`. Para ello haga clic en el componente `div` que rodea al elemento `newsitem` y en la ventana Estilos 58, en la columna **Estilos para:**, seleccione **`1 párrafo`**. Después en el grupo de propiedades `fuente`, asigne el valor `Arial` a la propiedad `font-family`. Esta propiedad será heredada por los cinco formatos predefinidos descendientes.

Ahora, en la vista **Diseño**, seleccione los tres componentes `div` que rodean a los nodos `dateline`, `synopsis` y `source` (mantenga pulsada la tecla **Mayús** mientras hace clic en los componentes `div`). En la ventana Estilos 58 seleccione la entrada **`3 párrafos`** y después, en el grupo de propiedades fuente, asigne el valor `10pt` a la propiedad `font-size`. (El componente `h3` no se seleccionó porque ya tiene una propiedad `font-size` con valor `12pt`.)

Por último, en la vista **Diseño**, seleccione el componente `div` que rodea al elemento `dateline`. En la columna **Estilos para:** de la ventana Estilos 58 seleccione **`1 párrafo`**. En el grupo de propiedades fuente, asigne el valor `bold` a la propiedad `font-weight` y el valor `italic` a la propiedad `font-style`. En el grupo de propiedades `color`, asigne el valor `gray` a la propiedad `color`. Este sería el formato de salida del elemento `dateline`.

2006-04-01: Boston, USA

Observe que el estilo definido para el componente `div` también se aplicó al texto estático que está dentro del componente `div` (es decir, a los dos puntos y a la coma).

Definir estilos locales para un componente

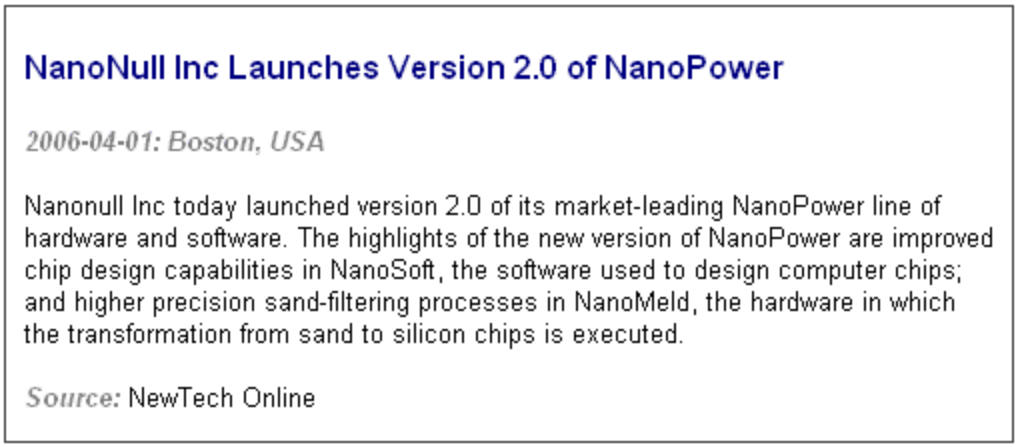
Los estilos locales definidos para un componente sustituyen a los demás estilos definidos en niveles superiores del SPS para dicho componente. Veamos un ejemplo: en la vista **Diseño** seleccione el elemento `headline` y asígnele el color `navy` (propiedad `color` del grupo de propiedades `color` de la ventana Estilos). La propiedad definida localmente (`color:navy`) reemplaza al estilo global definido para la clase `.encabezado` (`color:green`).

Seleccione el componente `div` que rodea al elemento `source`. En la ventana Estilos 58 seleccione **`1 párrafo`** en la columna **Estilos para:** y después asigne el valor `gray` a la propiedad `color` (del grupo de propiedades `color`). En el grupo de propiedades `fuente`, asigne el valor `bold` a la propiedad `font-weight`. Estos valores se aplican al texto estático. Recuerde que en el último párrafo asignamos el valor `italic` a la propiedad `font-style` del texto estático "`Source: `". Las nuevas propiedades (`font-weight:bold` y `color:gray`) se aplican junto con la propiedad `font-style:italic`.

Ahora, en la vista **Diseño**, seleccione el marcador de posición `(contenido)` del elemento `source`. En la columna **Estilos para:** de la ventana Estilos seleccione **`1 contenido`** y asigne el valor `black` a la propiedad `color` (del grupo de propiedades `color`). En el grupo de propiedades `fuente` asigne el valor normal a la propiedad `font-weight`. Las nuevas propiedades se aplican al marcador de posición `(contenido)` del elemento `source` y reemplazan a las propiedades definidas para el componente `div` (*imagen siguiente*).

Últimos retoques al formato

Para terminar de dar formato a esta sección del diseño, seleccione el componente `div` que rodea al elemento `synopsis`. En el cuadro combinado Formatos predefinidos 375 de la barra de herramientas seleccione la opción **p (párrafo)**. Esto aplica a todo el bloque los estilos inherentes del elemento HTML `p`. Ahora la vista previa HTML del diseño tiene este aspecto:



Ahora guarde el archivo de diseño.

© 2016-2022 Altova GmbH

3.5 Usar cálculos automáticos

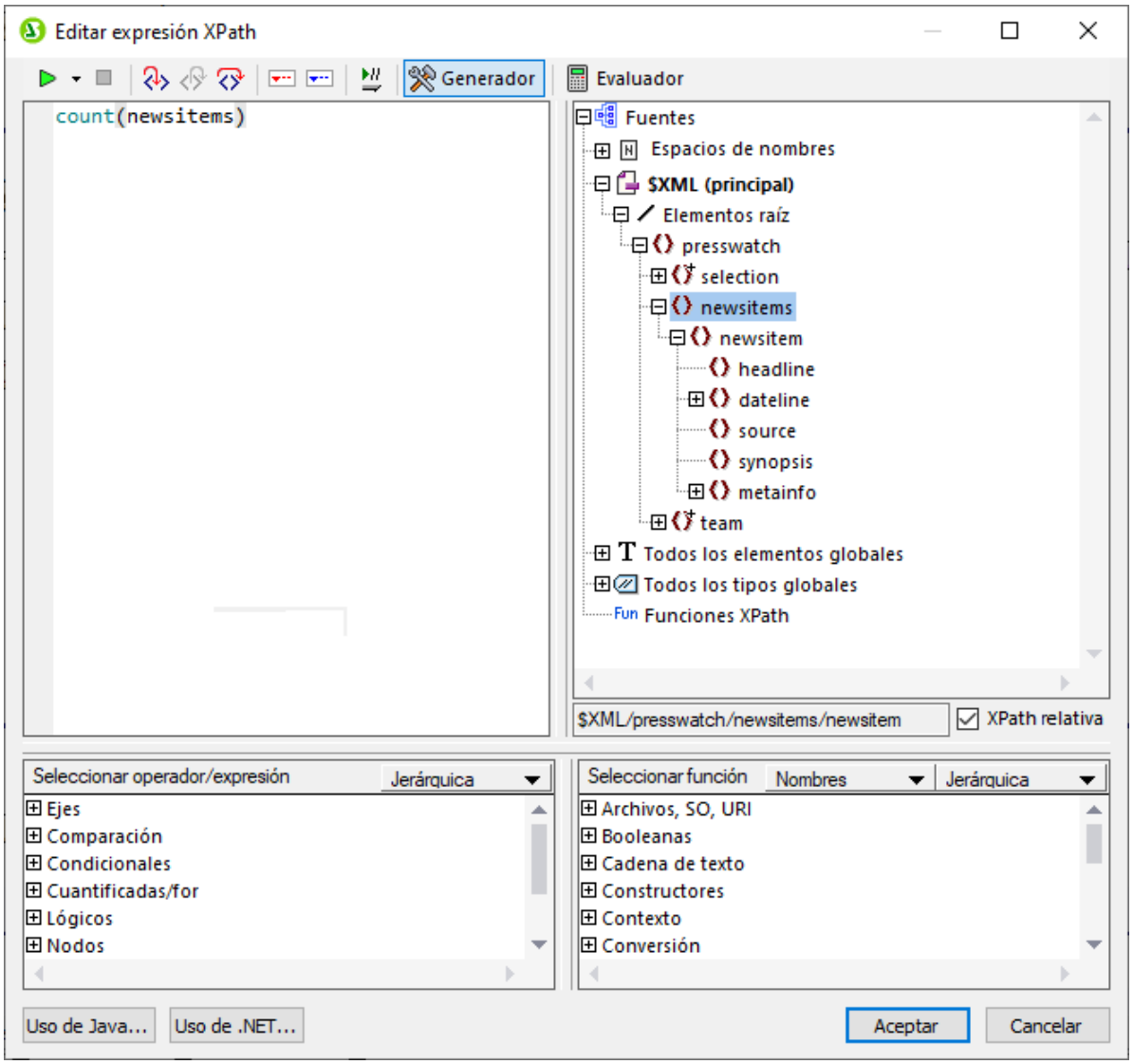
Los cálculos automáticos 293 son un potente mecanismo que permite obtener información adicional sobre los datos XML disponibles. En este apartado del tutorial añadirá dos datos al diseño: el número total de noticias (elementos `newsitem`) y el período de tiempo cubierto por las noticias del documento XML. Estos datos no están disponibles en el documento XML explícitamente pero se pueden calcular o manipular a partir de los datos disponibles.

Contar los nodos newsitem

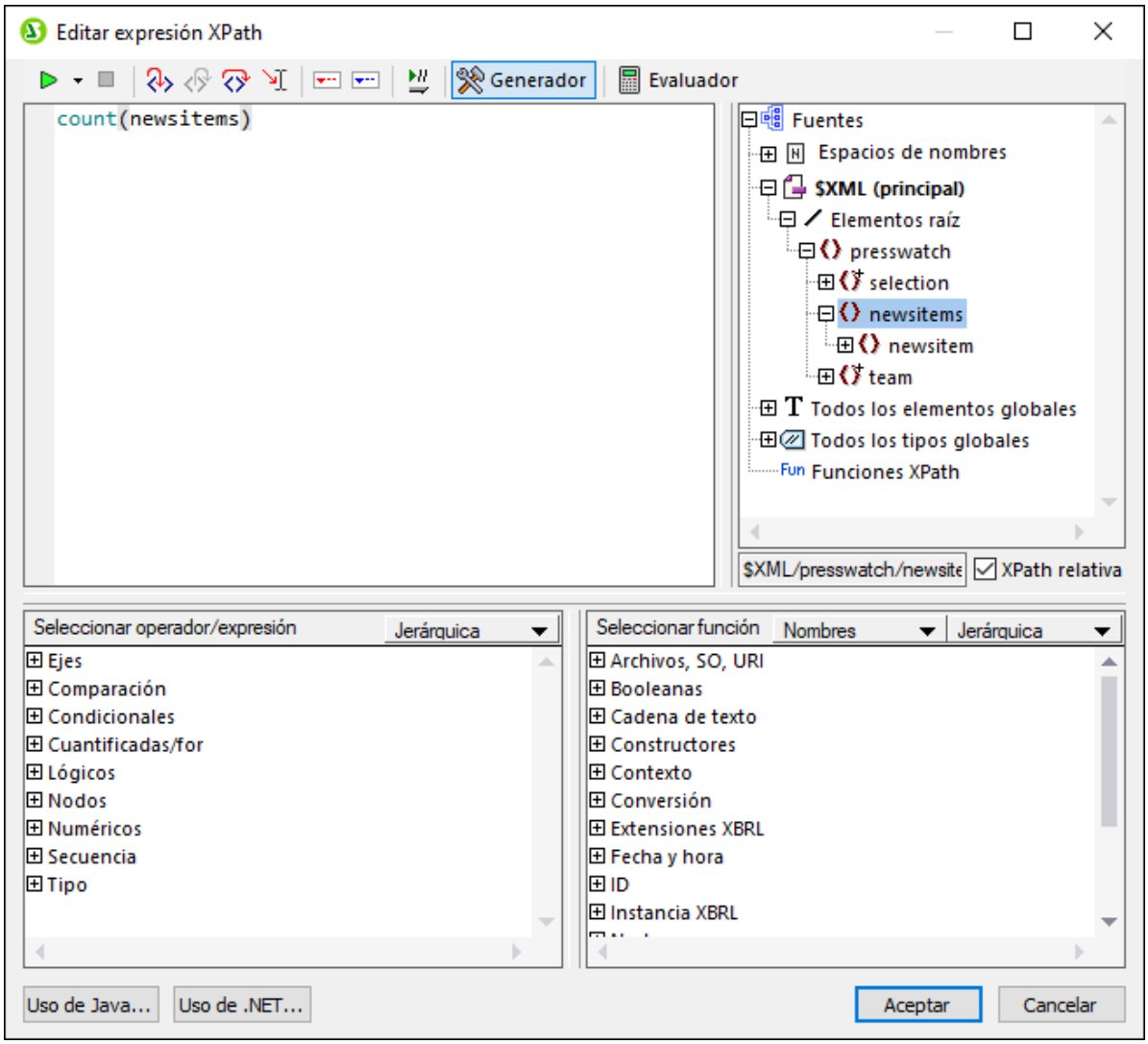
Para calcular cuántas noticias hay en el documento XML (cuántos elementos `newsitem` hay):

1. Primero debe crear espacio suficiente para una línea de texto estático. En esta línea insertaremos también el cálculo automático. Utilice la tecla **Entrar** para añadir líneas nuevas e inserte una línea horizontal bajo el espacio recién creado (*imagen siguiente*).

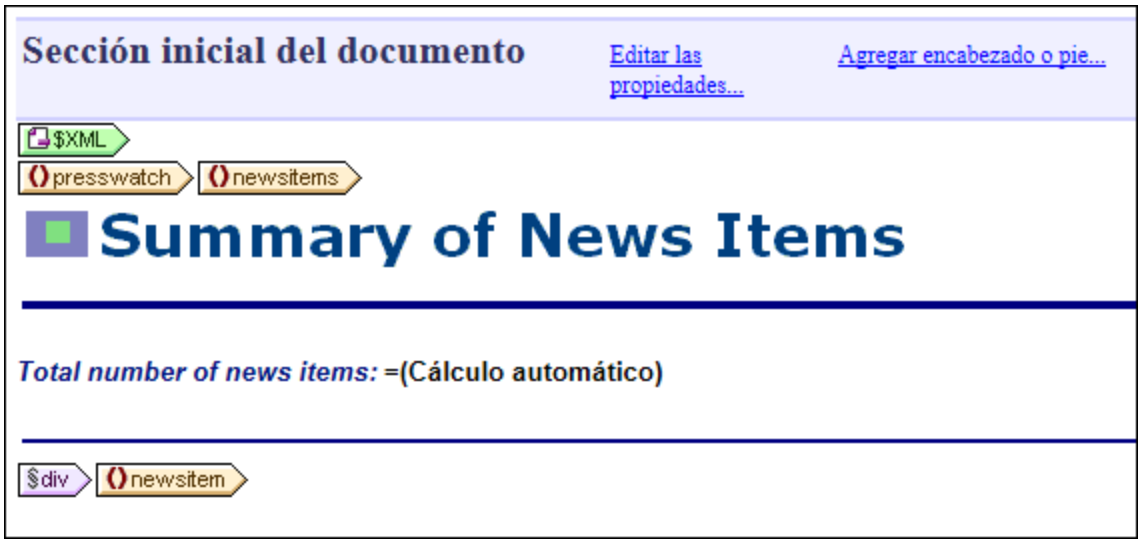
2. Escriba el texto estático "`Total number of news items:` " (*número total de noticias*) tal y como muestra la imagen anterior.
3. Aplique el estilo local que prefiera al texto estático. Para ello siga las instrucciones del apartado anterior Aplicar formato al contenido 91.
4. Ponga el cursor justo después de los dos puntos y seleccione el comando **Insertar | Cálculo automático | Valor**. Se abre el cuadro de diálogo Editar expresión XPath 729 (*imagen siguiente*). Si lo prefiere, también puede hacer clic con el botón derecho y seleccionar el mismo comando en el menú contextual.



5. En el panel donde aparece el árbol del esquema observe que el nodo de contexto es `newsitems`, el cual aparece resaltado. Ahora escriba la expresión `count(newsitem)` en el recuadro superior del diálogo o genere la expresión usando los dos paneles de ayuda situados a la derecha. Por ejemplo, haga doble clic en la función `count` (situada en el grupo de funciones **`Secuencia`**) para insertarla en el recuadro superior. Después ponga el cursor dentro del paréntesis de la función y haga doble clic en el nodo `newsitem` del árbol del esquema. Para ver qué resultado devuelve la expresión, haga clic en el botón **Evaluador**. El resultado de la evaluación aparece en el panel *Resultados*, situado en la parte inferior derecha (*imagen siguiente*). Para más información sobre cómo usar este cuadro de diálogo, consulte el apartado Editar expresiones XPath 729.

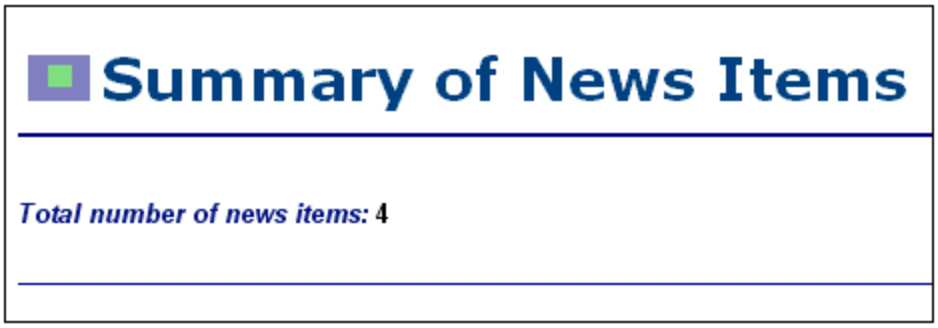


6. Para terminar haga clic en **Aceptar** y el cálculo automático se inserta en la posición del cursor (*imagen siguiente*).
7. Ahora aplique el formato que prefiera al cálculo automático usando estilos locales 91.



© 2016-2022 Altova GmbH

Ahora la vista previa HTML del diseño tiene este aspecto:

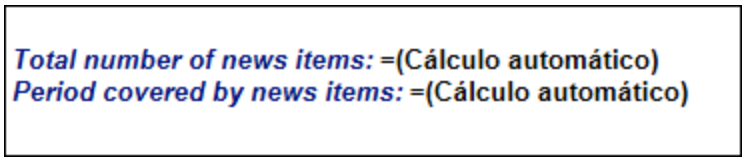


Ver el período de tiempo cubierto por las noticias

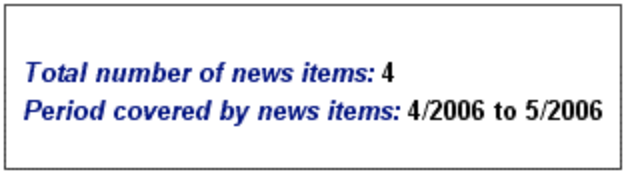
El período de tiempo total cubierto por todas las noticias se puede calcular si obtenemos la fecha de la noticia más antigua y la fecha de la noticia más reciente. Para ello podemos usar las expresiones XPath que aparecen más abajo. La primera expresión XPath da como resultado el contenido del nodo `date`. La segunda expresión genera los valores de mes (`month`) y año (`year`) del nodo `date`. Puede usar una de estas dos expresiones:

- `concat(min(//date), ' to ', max(//date)).`
- `concat(month-from-date(min(//date)), '/', year-from-date(min(//date)), ' to ', month-from-date(max(//date)), '/', year-from-date(max(//date)))`

En la vista **Diseño** inserte el texto estático y el cálculo automático tal y como muestra la imagen siguiente. Después aplique los estilos locales que prefiera.



Ahora la vista previa HTML del diseño tiene este aspecto:



No olvide guardar el archivo de diseño.

3.6 Usar condiciones

Si abre el archivo `QuickStart.xml`, observará que cada elemento `newsitem` tiene un elemento secundario llamado `metainfo`, que a su vez contiene elementos secundarios llamados `relevance`. En el diseño SPS puede crear un cuadro combinado que tenga una lista desplegable con los valores de uno de los elementos `relevance`. Cuando el usuario de la vista **Authentic** seleccione una opción (un valor) de la lista desplegable del cuadro combinado, dicho valor se podrá enviar a un nodo del documento XML. Además podemos usar una condición para probar cuál es el valor seleccionado por el usuario (buscando dicho nodo) y aportar el procesamiento adecuado (opción de presentación) para cada valor seleccionado por el usuario.

En este apartado del tutorial aprenderá a crear una plantilla condicional que muestre solamente las noticias (los elementos `newsitem`) cuyo elemento `relevance` coincida con la opción seleccionada por el usuario.

El proceso sería el siguiente:

1. Se crea un cuadro combinado donde el usuario de la vista **Authentic** puede seleccionar el valor del nodo `byrelevance`. Los valores de la lista desplegable del cuadro combinado se obtienen con ayuda de una expresión XPath, que compila dinámicamente los valores del nodo `relevance`.
2. Se inserta una condición alrededor del elemento `newsitem`. Esta condición selecciona todos los elementos `newsitem` que tengan un elemento `relevance` cuyo contenido coincida con el contenido del nodo `byrelevance`. El contenido rodeado por la rama de una condición se conoce como *plantilla condicional*.
3. Dentro de la plantilla condicional, se enumeran todos los nodos `relevance` del elemento `newsitem`.
4. Se resalta el elemento `relevance` (en la lista de elementos `relevance`) que coincide con el elemento `byrelevance`. Para ello se crea una condición que permita seleccionar dichos elementos `relevance` y después se aplica formato especial a la plantilla condicional.
5. En la condición para el elemento `newsitem`, se inserta una rama que selecciona todos los elementos `newsitem`.

Primer paso: crear el cuadro combinado para seleccionar los diferentes valores del nodo

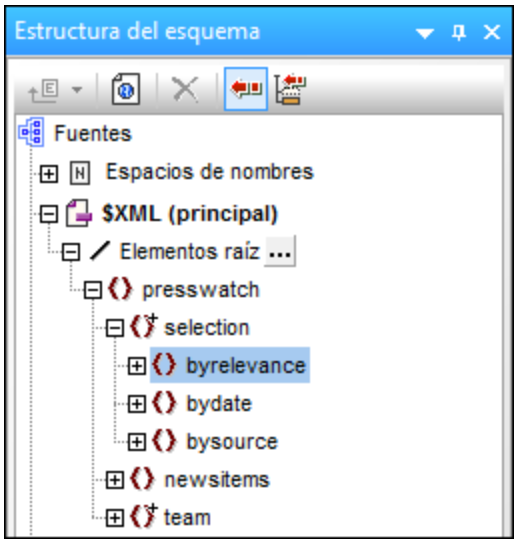
En el documento XML el nodo que contiene las opciones que el usuario puede seleccionar es el nodo `/presswatch/selection/byrelevance`. Este es el nodo a partir del cual crearemos el cuadro combinado. Esto se hace así:

1. Inserte el texto estático "`Select by relevance: `" (seleccionar por relevancia) justo después del segundo cálculo automático 93 (*imagen siguiente*).

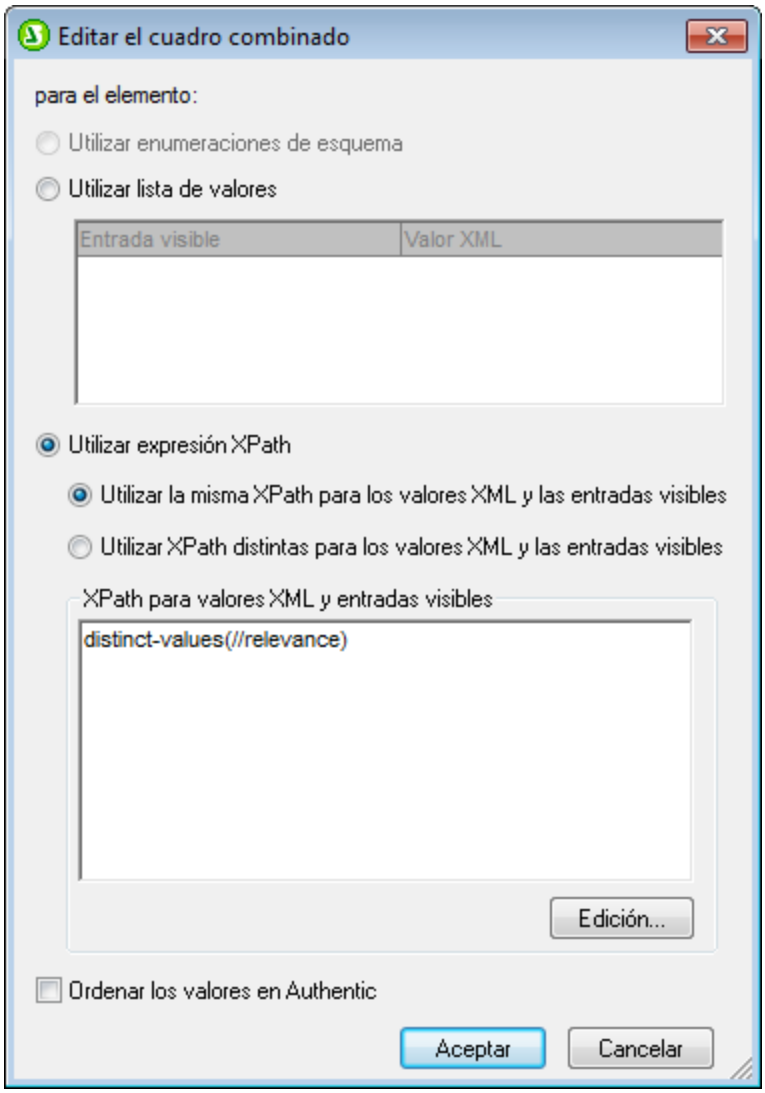
   Total number of news items: **=(Cálculo automático)**
   Period covered by news items: **=(Cálculo automático)**
   Select by relevance:

2. Arrastre el nodo `byrelevance` desde la ventana Estructura del esquema 48 (*imagen siguiente*) y suéltelo justo después del texto estático que escribió en el paso anterior.

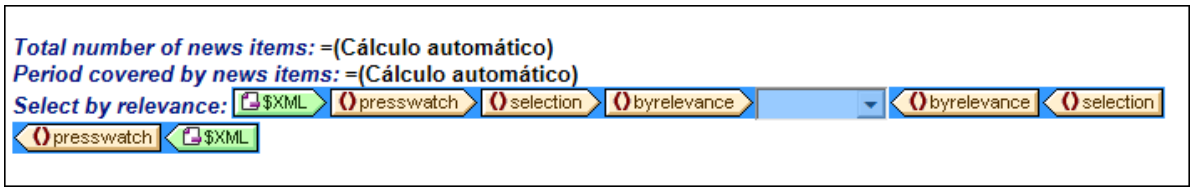
 © 2016-2022 Altova GmbH



3. En el menú contextual que aparece seleccione el comando **Crear cuadro combinado**. Se abre el cuadro de diálogo "Editar el cuadro combinado":

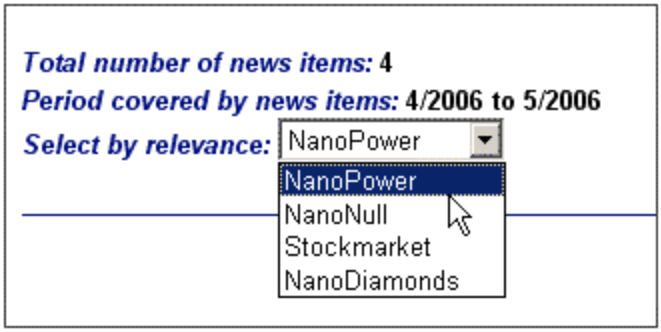


4. Seleccione el botón de opción *Utilizar expresión XPath* y después la opción *Utilizar la misma XPath para los valores XML y las entradas visibles*. En el cuadro de edición *XPath para los valores XML y entradas visibles* escriba esta expresión XPath: `distinct-values(//relevance)`. Esta expresión XPath selecciona los valores únicos de todos los elementos `relevance` del documento XML.
5. Para terminar haga clic en **Aceptar**. El cuadro combinado se inserta y el diseño tiene este aspecto:



6. Cambie a la vista [38] **Authentic** [38]. Haga clic en la flecha del cuadro combinado y observe que la lista desplegable contiene los valores únicos de todos los nodos `relevance` (*imagen siguiente*). Compare la lista con el documento XML. Se trata de una lista dinámica que crecerá a medida que se añadan nuevos valores a los elementos `relevance` del documento XML.

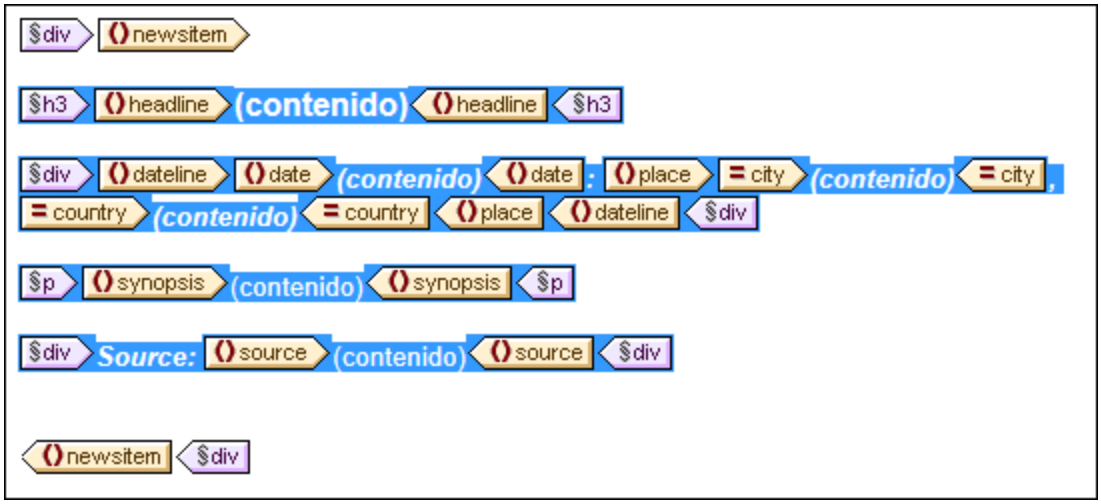
© 2016-2022 Altova GmbH



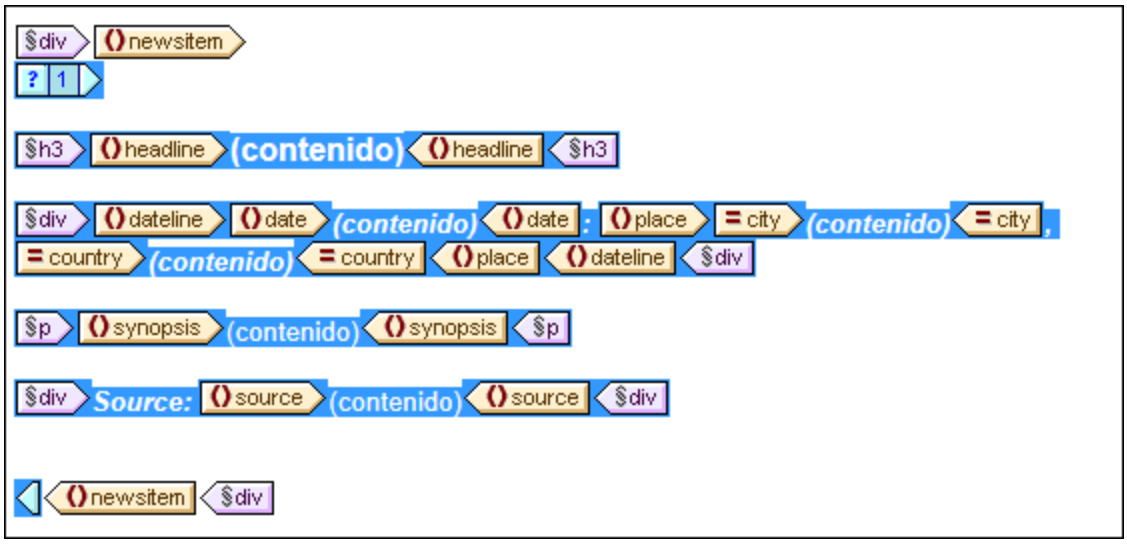
Segundo paso: insertar una condición para mostrar las noticias de la relevancia seleccionada

La condición selecciona los elementos `newsitem` cuyo elemento `metainfo/relevance` tenga el mismo valor que el valor seleccionado por el usuario (y enviado al elemento `/presswatch/selection/byrelevance`). Siga estos pasos para insertar la condición:

1. Seleccione el contenido de la parte `newsitem` del diseño que debe formar parte de la condición (*imagen siguiente*).



2. Seleccione el comando de menú (o el comando del menú contextual) **Incluir en | Condición** 840. Se abre el cuadro de diálogo Editar expresión XPath 729.
3. Introduzca la expresión `metainfo/relevance=/presswatch/selection/byrelevance`. Esta expresión devuelve el resultado `true` si el valor del descendiente `metainfo/relevance` del elemento `newsitem` actual coincide con el valor del elemento `/presswatch/selection/byrelevance` (el valor seleccionado por el usuario).
4. Haga clic en **Aceptar**. La condición se crea alrededor del contenido del elemento `newsitem` (*imagen siguiente*).



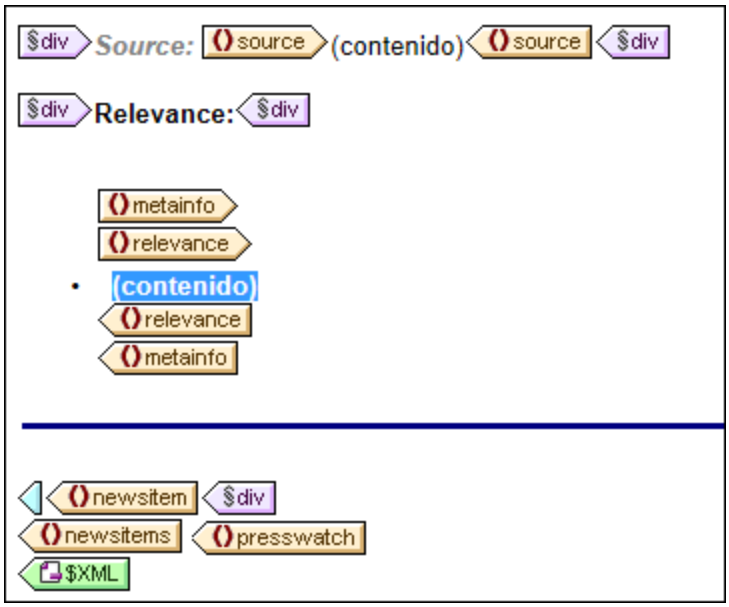
Observe que en esta condición solamente hay una rama. El documento de salida incluirá solamente las noticias (elementos `newsitem`) para las cuales la condición obtenga el resultado `true`. Por tanto, en este caso la condición funciona como un filtro. Más adelante aprenderá a añadir otra rama a la condición.

Insertar una lista a partir del nodo **`relevance`**

Para mostrar los nodos `relevance` de cada elemento `newsitem`, siga esta instrucciones:

1. Cree un poco de espacio vertical debajo del componente `div` que rodea al elemento `source` y también dentro de la etiqueta final de la plantilla condicional.
2. Escriba el texto estático "`Relevance:`" (*Relevancia*) y cree a su alrededor un formato predefinido `div` (resalte el texto estático e inserte el formato predefinido).
3. Arrastre el elemento `relevance` desde el nodo **`Elementos raíz`** de la ventana Estructura del esuqema 48 y suéltelo en el diseño, justo debajo del texto estático `Relevance:`.
4. En el menú contextual que aparece al soltar el nodo en el diseño seleccione la opción **Numeración y viñetas** para crear una lista a partir del nodo. Después seleccione el formato de lista que desea usar.
5. Aplique el formato y el estilo que prefiera al contenido de la lista. Cuando termine, el diseño debería tener este aspecto:

 © 2016-2022 Altova GmbH

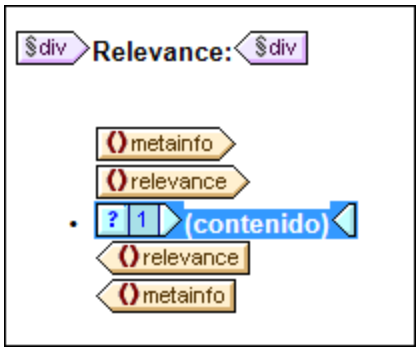


Ahora, en la vista **Authentic**, compruebe qué ocurre si selecciona diferentes valores de `relevance`. Use el cuadro combinado para seleccionar los valores.

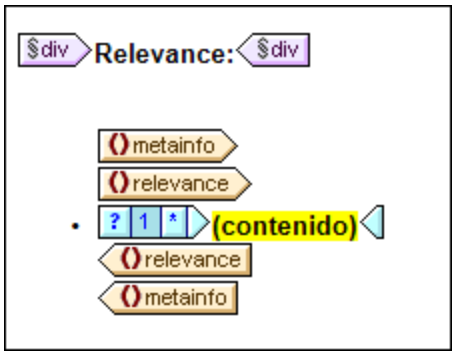
Poner el elemento `relevance` seleccionado en negrita

Algunos elementos `newsitem` tienen más de un elemento `relevance`. En estos casos podemos optimizar el diseño para resaltar el elemento `relevance` que coincide con la opción seleccionada por el usuario.

1. Seleccione el elemento `relevance` en la vista **Diseño**.
2. Inserte una condición formada por la expresión XPath `.=/presswatch/selection/byrelevance`. Esto crea una condición con una sola rama (*imagen siguiente*) que selecciona los elementos `relevance` que coinciden con el elemento `byrelevance`.



3. Seleccione el marcador de posición `(contenido)` y asígnele el formato local `bold` (en la ventana Estilos, en el grupo de propiedades `fuente`) y el formato local `background-color:yellow` (en el grupo `color`).
4. Haga clic con el botón derecho en la condición y seleccione **Copiar rama** en el menú contextual.
5. Aparece el cuadro de diálogo Editar expresión XPath 729. Marque la casilla *Otherwise* que aparece en la esquina superior derecha del cuadro de diálogo.
6. Haga clic en **Aceptar** y se crea la nueva rama (`Otherwise`). Esta rama de la condición selecciona todos los elementos `relevance` que no coinciden con el elemento `byrelevance`.



7. Observe que el contenido de la rama `Otherwise` es una copia de la primera rama, pero el marcador de posición `(contenido)` aparece en negrita y el color de fondo es amarillo. Quite este formato del marcador de posición `(contenido)`.

Llegados a este punto, el diseño contiene una condición con dos ramas (cada una con su plantilla condicional) que lleva a cabo esta prueba en todos los elementos `relevance` del documento XML:

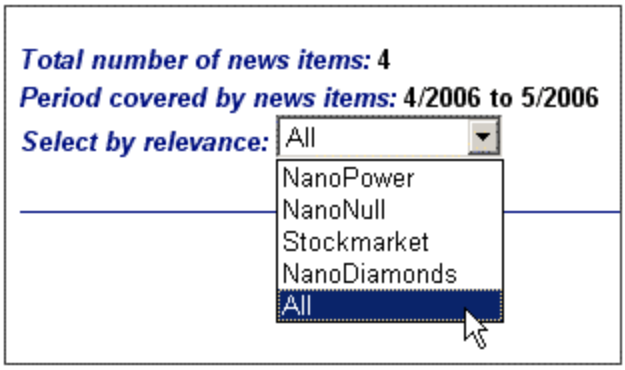
- Si el contenido del elemento `relevance` es idéntico al contenido del elemento `/presswatch/selection/byrelevance`, entonces el contenido de `relevance` aparece en negrita y con color de fondo amarillo.
- De lo contrario (`otherwise`), el contenido de `relevance` aparece con el formato normal.

Pruebe un poco estas condiciones en la vista **Authentic**.

Modificar el cuadro combinado e insertar otra rama condicional

En el cuadro combinado donde el usuario de la vista **Authentic** selecciona un valor de `byrelevance`, no hay ninguna opción que permita seleccionar todos los elementos `newsitem`. Para incluir esta opción:

1. Seleccione el cuadro combinado en la vista **Diseño**.
2. En la ventana Propiedades expanda el grupo de propiedades `cuadro combinado` y haga clic en el botón **Editar** de la propiedad `Valores de entrada del cuadro combinado`.
3. Aparece el cuadro de diálogo Editar el cuadro combinado 194. Cambie la expresión XPath `distinct-values(//relevance)` por la expresión `distinct-values(//relevance), 'All'`. Esta expresión añade la cadena `All` a la secuencia de elementos que devuelve la expresión XPath.
4. Mire otra vez la lista desplegable del cuadro combinado en la vista **Authentic** (*imagen siguiente*).



A partir de ahora si el usuario selecciona la opción **All**, este valor (`All`) se envía al nodo `/presswatch/selection/byrelevance`. La idea es que cuando el nodo `byrelevance` contenga el valor `All`, el documento de salida muestre todas las noticias (todos los elementos `newsitem`).

La condición que muestra la plantilla `newsitem` tiene una sola rama con la expresión `metainfo/relevance=/presswatch/selection/byrelevance`. Como ningún nodo `metainfo/relevance` tiene el valor `All`, el documento de salida no mostrará ninguna noticia cuando `All` sea el valor del nodo `byrelevance`. Lo que tenemos que hacer ahora es crear otra rama para la condición. Esta nueva rama buscará un valor de `All`. Al crear una plantilla `newsitem` dentro de esta rama, conseguiremos ver todas las noticias si la prueba da `true` como resultado. Para ello:

1. Seleccione la condición `newsitem` en la vista **Diseño**.
2. Haga clic con el botón derecho en la condición y seleccione **Copiar rama** en el menú contextual.
3. Aparece el cuadro de diálogo Editar expresión XPath. 729 Introduzca la expresión `/presswatch/selection/byrelevance='All'`.
4. Haga clic en **Aceptar** y se crea la rama nueva.

El contenido de esta nueva rama es una plantilla de la primera rama. Lo que hace la nueva rama es generar la plantilla de las noticias si el usuario selecciona la opción **All** en el cuadro combinado.

Ahora no olvide guardar el archivo de diseño.

3.7 Usar plantillas globales

Las plantillas globales [31] sirven para especificar el procesamiento de un elemento de forma global. Esto permite utilizar las reglas de la plantilla global (definidas en una posición) en varias posiciones de la hoja de estilos. Puede usar las plantillas globales de dos formas distintas:

- Puede copiar las reglas de la plantilla global en la plantilla local.
- Una plantilla local (de la plantilla principal) puede enviar el procesamiento de ese nodo a la plantilla global. Una vez ejecutada la plantilla global, el procesamiento continua en la plantilla principal. En este caso diríamos que se invoca a la plantilla global o que la plantilla global se utiliza desde la plantilla principal.

StyleVision ofrece dos mecanismos para invocar a una plantilla global desde la plantilla principal:

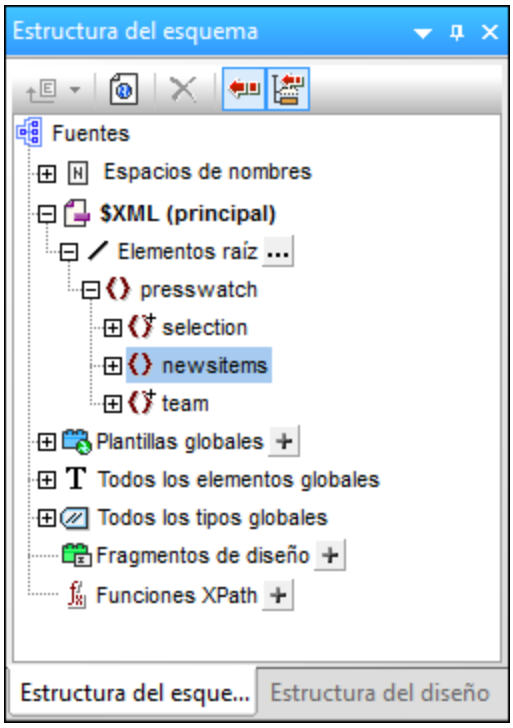
- Una plantilla local puede hacer referencia a una plantilla global.
- Una instrucción `(resto de contenido)` de la plantilla principal puede aplicar plantillas a los elementos descendientes del elemento actual (es decir, al resto de contenidos del elemento actual). Si una plantilla global existe para uno de los elementos descendientes, la plantilla global se le aplica a dicho elemento. De lo contrario, se aplica la plantilla integrada. La plantilla integrada procesa los elementos secundarios y presenta el contenido de texto de los elementos. Como resultado, se presenta el contenido de texto de todos los elementos descendientes. No olvide que el resultado **no incluye** el valor de los atributos.

En este apartado del tutorial aprenderá a crear un diseño para la plantilla de los miembros del equipo de la redacción usando la instrucción resto de contenido y una plantilla global para el elemento global [31] `member`.

Insertar la instrucción **`resto de contenido`**

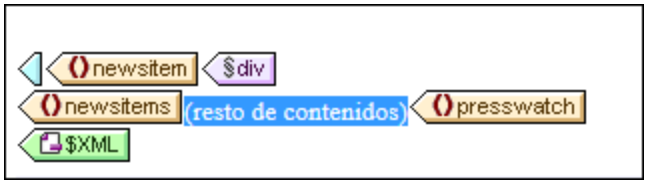
En la imagen siguiente aparece la estructura general del esquema.

© 2016-2022 Altova GmbH



El elemento de documento `presswatch` tiene tres secundarios: (i) `selection`, (ii) `newsitems` y (iii) `team`. La plantilla principal creada en los apartados anteriores procesa el elemento `/presswatch`. Dentro del elemento `presswatch` solamente se procesa el elemento `newsitems`. Los elementos `selection` y `team` no se procesan dentro del elemento `presswatch` (aunque `selection` se procesó dentro del elemento `newsitems`). Por tanto, si introducimos la instrucción `resto de contenido` dentro de `presswatch`, también se procesarán los elementos `selection` y `team`.

Inserte la instrucción `resto de contenido` en el diseño: ponga el cursor entre las etiquetas finales de los elementos `newsitems` y `presswatch` y seleccione el comando de menú (o el comando del menú contextual) **Insertar | Resto de contenido** 813. Se inserta el marcador de posición `(resto de contenido)` en la posición del cursor (*imagen siguiente*).



Si abre la vista previa HTML podrá ver esta cadena de texto:

Esta cadena es el resultado que se obtiene al aplicar las plantillas integradas a los elementos `selection` y `team`. La plantilla integrada para elementos procesa los elementos secundarios. La plantilla integrada para nodos de texto genera el texto del nodo de texto. Al combinar estas dos plantillas integradas obtenemos el contenido de texto de todos los nodos descendientes de los elementos `selection` y `team`. El texto `All` viene de `selection/byrelevance` y aparece antes del texto de salida de los nodos `first`, `last`, `email`,

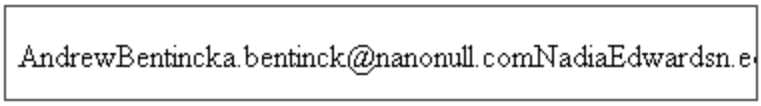
descendientes del nodo `team/member`, en el orden en el que aparecen en el documento XML. Observe que el atributo `id` del elemento `member` no se incluye en el resultado (porque es un atributo y, por tanto, no se considera elemento secundario de `member`).

Crear una plantilla global para `selection`

Como no necesitamos incluir el contenido de `selection` en el resultado, crearemos una plantilla global vacía para `selection`, para que su contenido no se procese. Esto se hace de la siguiente manera:

1. Haga clic con el botón derecho en el nodo `selection` en la ventana Estructura del esquema 48 de la vista **Diseño**.
2. En el menú contextual que aparece seleccione el comando **Convertir en plantilla global o quitarla**. Se crea una plantilla global para el elemento `selection` (*imagen siguiente*).

3. Haga clic en el marcador de posición `(contenido)` de la plantilla global y pulse la tecla **Suprimir** del teclado. El contenido del marcador de posición se elimina.
4. Abra la vista previa HTML y observe que el texto `All` ya no aparece en la línea de texto generada por las plantillas integradas (*imagen siguiente*).



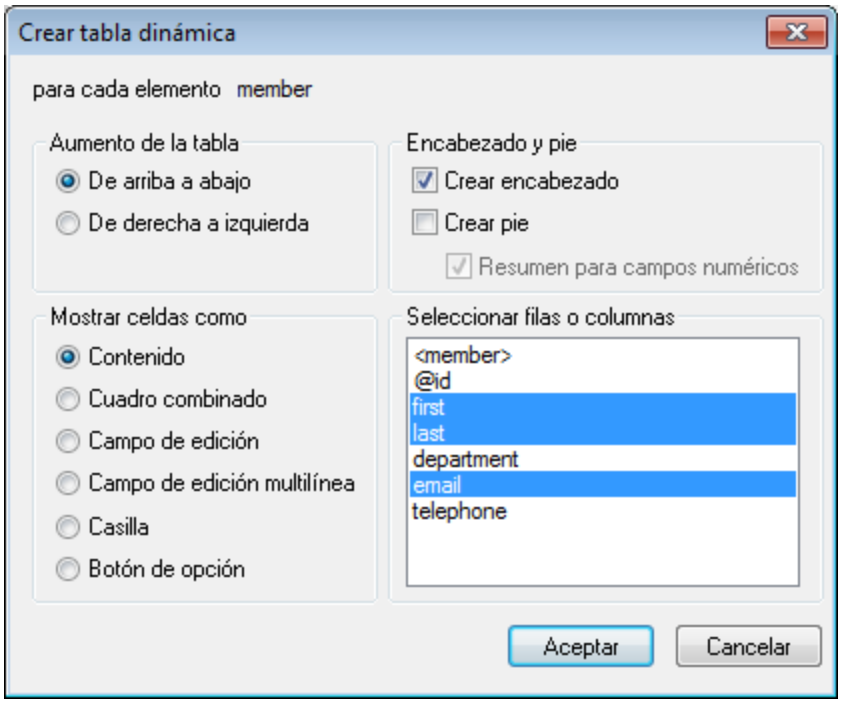
Como la plantilla global para `selection` está vacía, los elementos secundarios de `selection` no se procesan.

Crear una plantilla global para `team/member`

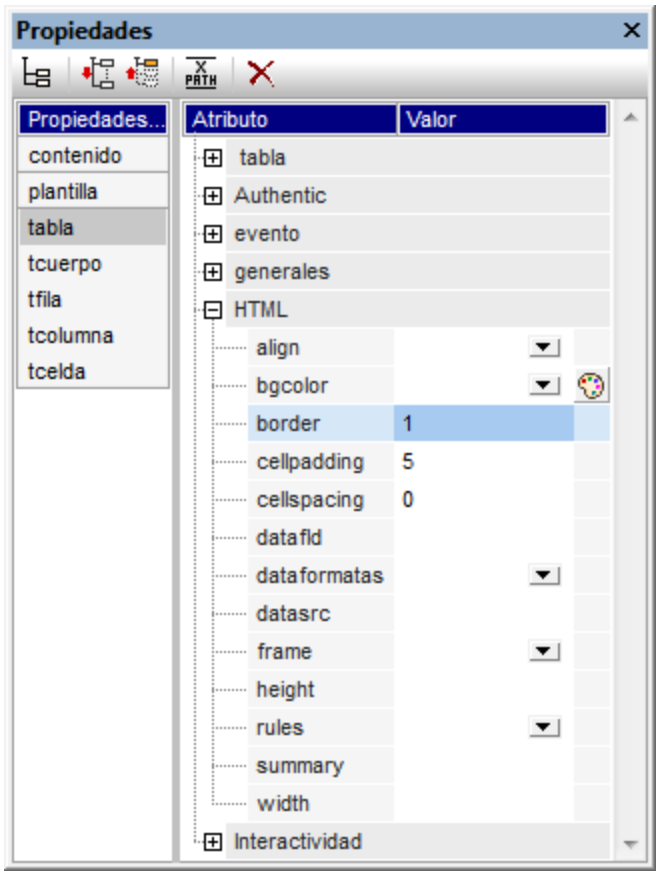
El objetivo es crear una tabla que muestre los datos de los miembros del equipo de control de prensa. Esta tabla se creará en una plantilla global para el elemento `team`. Esto se hace de la siguiente manera:

1. Cree una plantilla global para el elemento `team` (clic con el botón derecho en `team` en la lista **`Todos los elementos globales`** de la ventana Estructura del esquema y seleccione **Convertir en plantilla global o quitarlo**). Esta será la plantilla global del equipo de control de prensa.
2. Expanda el elemento `team` en la lista **`Todos los elementos globales`** y arrastre su elemento secundario `member` hasta la plantilla global `team` en el diseño.
3. Al soltar el elemento en la plantilla global `team` del diseño aparece un menú contextual. Seleccione el comando **Crear tabla**. Aparece el cuadro de diálogo "Crear tabla dinámica" (*imagen siguiente*).

 © 2016-2022 Altova GmbH



4. En el panel *Seleccionar filas o columnas*, anule la selección de `@id`, `department` y `telephone` (*imagen anterior*) y haga clic en **Aceptar**. La tabla dinámica se crea en el diseño.
5. Ponga el cursor en una celda del cuerpo de la tabla y seleccione **`tabla`** en la columna **Propiedades para:** de la ventana Propiedades 59. Después defina las propiedades de tabla que muestra la imagen siguiente.



6. Si quiere puede definir más propiedades en las ventanas Propiedades y Estilos. Por ejemplo, puede dar un color de fondo a la fila de encabezado. Para ello ponga el cursor en la fila de encabezado, seleccione **`tfila`** en la columna **Estilos para:** de la ventana Estilos y asigne un valor a la propiedad `background-color` (del grupo `color`). También puede editar los encabezados, que son cadenas de texto estático. Para terminar, si el marcador de posición `(contenido)` del elemento `team` aún está en la plantilla global, elimínelo.

Ahora la vista previa HTML del diseño tiene este aspecto:

First	Last	Email
Andrew	Bentinck	a.bentinck@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com
John	Edwards	j.edwards@nanonull.com
Janet	Ashe	j.ashe@nanonull.com

© 2016-2022 Altova GmbH

3.8 ¡Eso es todo!

¡Enhorabuena! En este tutorial aprendió los aspectos más importantes del proceso de creación de diseños SPS:

- Cómo crear la estructura 73 del documento (plantilla principal 73 y plantillas globales 106).
- Cómo insertar contenido dinámico 73 y estático 80 en el diseño usando diferentes componentes SPS dinámicos y estáticos.
- Cómo usar estilos CSS 73 de hojas de estilos externas 88, de reglas de estilo globales 89 y de reglas de estilo locales 91.
- Cómo usar cálculos automáticos 93 para derivar información adicional de los datos XML disponibles.
- Cómo usar condiciones 98 para filtrar datos XML y obtener resultados diferentes dependiendo de los valores de los datos XML.
- Cómo usar plantillas globales 108 y la instrucción resto de contenido 106.

Para más información sobre estas funciones y características, consulte el apartado correspondiente de estas cinco grandes secciones:

- Archivo SPS: contenido 134
- Archivo SPS: estructura 219
- Archivo SPS: características avanzadas 292
- Archivo SPS: presentación 374
- Archivo SPS: funciones adicionales 443

Estas secciones también describen muchas otras características de StyleVision que no aparecen en el tutorial de introducción rápida.

¿Qué hacer con el diseño SPS?

Cuando termine de diseñarlo, podrá usar el archivo SPS de varias maneras:

- Para editar documentos XML en la vista **Authentic** de XMLSpy o de Authentic Desktop. (Las ediciones Enterprise y Professional contienen una pestaña de vista previa **Authentic**, a la que le faltan algunas características, como los ayudantes de entrada y los iconos de estado del texto.) Estos dos productos ofrecen una vista **Authentic** completamente funcional, donde puede probar las características de los ayudantes de entrada. Para editar el archivo `QuickStart.xml` en la vista **Authentic** de XMLSpy o de Authentic Desktop, asigne el archivo XML al diseño `MiTutorial.sps` y cambie a la vista **Authentic**.
- Puede generar hojas de estilos XSLT para transformar el archivo XML en documentos HTML y texto. Las hojas de estilos XSLT se generan con el comando **Archivo | Guardar archivos generados** 782 o desde StyleVision Server. Pruebe a generar hojas de estilos XSLT a partir del diseño `MiTutorial.sps` y después use las hojas de estilos para transformar el archivo XML `QuickStart.xml`.

4 Resumen de características

Objetivos

Los documentos SPS creados con StyleVision tienen dos usos principales:

- Controlar la presentación de documentos XML en la vista **Authentic** y habilitar la entrada de datos en documentos XML o BD desde la interfaz de la vista **Authentic**.
- Generar hojas de estilos XSLT para formatos de salida HTML, texto y RTF.

De este modo puede usar los diseños SPS para habilitar la edición de documentos XML y generar documentos de salida en formato HTML, texto y RTF a partir del documento XML editado. Además, las hojas de estilos XSLT generadas se pueden usar para transformar otros documentos XML basados en el mismo esquema que el archivo SPS.

Pasos para crear un archivo SPS

A continuación resumimos los pasos necesarios para crear un archivo SPS nuevo.

1. Asigne un esquema 48 al archivo SPS vacío recién creado. El esquema puede ser: (i) un archivo de esquema (DTD o esquema XML); (ii) un esquema XML generado a partir de una BD (*solo en las ediciones Enterprise y Professional*); (iii) un esquema basado en una taxonomía XBRL (*solo en la edición Enterprise*); (iv) un esquema definido por el usuario (creado directamente en StyleVision). El esquema se asigna en la ventana Vista general del diseño 45. Si lo prefiere, también puede crear un SPS nuevo con un esquema directamente (con el comando **Archivo | Nuevo**).
2. Asigne un archivo XML de trabajo 48 al archivo SPS. El archivo XML de trabajo 30 aporta los datos XML procesados por el SPS a la hora de generar la vista previa de Authentic y de los documentos de salida. El archivo XML de trabajo 30 se asigna en la ventana Vista general del diseño 45 y permite obtener una vista previa de los resultados en StyleVision.
3. Seleccione la versión XSLT que desea usar 115. Para poder generar resultados en texto, la versión XSLT debe ser XSLT 2.0 o XSLT 3.0.
4. Seleccione la opción de compatibilidad de Internet Explorer 116 que concuerde con la versión de IE instalada en su equipo.
5. El documento SPS se diseña en la vista 37 **Diseño** 37 gracias a los diferentes componentes de diseño. El proceso de diseño 114 consiste en crear la estructura del documento y definir propiedades de presentación 374. Si quiere imprimir los resultados 419, también puede especificar propiedades de formato de impresión 419.
6. Pruebe los resultados en la vista Authentic y la vista previa de los resultados. Si necesita hacer cambios en el diseño, hágalos y vuelva a probar el resultado del archivo SPS.
7. Si los necesita, genere 121 archivos XSLT y archivos de salida 121.
8. También puede asignar un archivo de plantilla XML. Este archivo de plantilla XML 113 ofrece datos de partida para un documento XML nuevo que se puede editar en la vista **Authentic** gracias al SPS.
9. El archivo SPS se implementa y se distribuye 118 entre los usuarios de la vista **Authentic**.

© 2016-2022 Altova GmbH

4.1 Archivos SPS y fuentes de datos

Crear un archivo SPS nuevo

Para crear un documento SPS nuevo seleccione una opción del comando **Archivo | Nuevo (Ctrl+N)** 761 o haga clic en el icono **Diseño nuevo** de la barra de herramientas Estándar 759. Se crea un documento SPS nuevo y se abre en la vista **Diseño**. El documento nuevo recibe el nombre provisional `SPS1.sps`, `SPS2.sps`, etc. dependiendo de la posición del archivo SPS en la secuencia de documentos creados desde que se inició la aplicación.

Después de crear el documento SPS nuevo es necesario asignarle archivos fuente.

Asignar archivos fuente al diseño SPS

A los diseños SPS se les puede asignar tres tipos de archivos fuente:

- Esquemas fuente 113
- Archivos XML de trabajo 113
- Archivos de plantilla XML 113

Estas asignaciones de archivos fuente se hacen en la ventana Vista general del diseño 45 y las instrucciones sobre cómo hacerlo están en el apartado Vista general del diseño 45 de esta documentación. A continuación enumeramos los aspectos más importantes de cada tipo de archivo fuente.

Esquemas fuente

Para poder crear la estructura del archivo SPS es necesario asignarle un esquema fuente. Los esquemas fuente se asignan desde la ventana Vista general del diseño 45. El esquema puede ser un archivo de esquema XML (archivos `.xsd`), un esquema XML generado a partir de un archivo XML, un esquema XML generado a partir de una BD, una DTD o un esquema definido por el usuario. Por cada esquema puede asignar un archivo XML de trabajo 113 opcional y un archivo de plantilla XML 113 opcional.

Nota: si quiere añadir un espacio de nombres a un archivo SPS o a una hoja de estilos XSLT generada a partir de un SPS, el espacio de nombres debe añadirse al elemento `schema` de nivel superior del esquema XML en el que se basa el SPS.

Archivo XML de trabajo

El archivo SPS puede tener asociado un archivo XML de trabajo 30. La función del archivo XML de trabajo 30 es aportar los datos XML de origen para las vistas previas de resultados de StyleVision y, por tanto, debe ser válido con respecto al esquema al que está asociado. El archivo XML de trabajo 30 se asigna desde la ventana Vista general del diseño 45.

Archivo de plantilla XML

El archivo SPS puede tener asociado un archivo de plantilla XML 30. La función del archivo de plantilla XML 30 es aportar los datos de partida del documento XML nuevo que se crea cada vez que el archivo SPS se abre en la vista Authentic 27 de un producto de software de Altova que incluya esta vista. El archivo de plantilla XML 30 debe ser válido con respecto al esquema al que está asociado y se asigna al SPS desde la ventana Vista general del diseño 45.

4.2 Crear el diseño

En el diseño SPS puede especificar:

1. Qué contenido 114 (del documento XML o de la BD) debe incluirse en el resultado. También puede insertar contenido en el SPS directamente para incluirlo en el resultado.
2. Qué estructura debe tener el resultado 114.
3. Qué propiedades de presentación (formato) 114 deben aplicarse a cada parte del resultado.

Contenidos del resultado

El contenido del resultado puede proceder de varias fuentes:

1. El documento XML, de la BD a los que se aplica el archivo SPS. El contenido del documento XML 31 se incluye en el diseño SPS arrastrando los nodos de datos XML desde la ventana Estructura del esquema 48 y soltándolos en la posición de destino del diseño.
2. Un documento XML externo al que puede acceder la aplicación (es decir, StyleVision o un producto de Altova con la vista Authentic 27).
3. Puede usar la función doc() de XPath 2.0 en un cálculo automático para acceder al contenido de archivos XML externos. Los documentos XML a los que se accede con la función doc() en una expresión XPath no tienen por qué asignarse al diseño SPS mediante asignaciones de esquemas fuente 48.
4. El propio diseño SPS. Puede insertar texto y otros contenidos (como imágenes y tablas) en el SPS directamente usando el teclado y otras funciones gráficas de StyleVision. Este contenido es independiente del documento XML.
5. Datos (del origen XML) dinámicos manipulados mediante expresiones XPath. Estas manipulaciones se suelen hacer con ayuda de cálculos automáticos 293.
6. Para los resultados en formato HTML puede generar contenidos con ayuda de funciones JavaScript 497 .

Estructura del resultado

En el diseño SPS puede controlar la estructura del resultado 265: (i) mediante el uso de plantillas: la estructura del resultado se especifica en una plantilla básica 265 (la plantilla principal 265 de StyleVision) y puede ser independiente de la estructura del documento XML; (ii) mediante declaraciones: se declaran reglas de plantilla para varios nodos 265 (las plantillas globales 265 de StyleVision) y, por tanto, se genera un resultado que sigue la estructura del documento XML o (iii) una combinación de ambos enfoques. En la vista **Diseño**, puede combinar una plantilla principal 265 con varias plantillas globales 265 para obtener la estructura deseada del documento de salida. Además puede usar archivos SPS modulares 251 y fragmentos de diseño 278 para poder estructurar el archivo SPS con mayor flexibilidad.

Presentación (formato) del resultado

En la vista **Diseño** puede aplicar propiedades de presentación a los componentes del diseño mediante estilos CSS. Los estilos se pueden definir localmente en el componente, para selectores declarados a nivel de documento y para selectores HTML declarados en una hoja de estilos CSS externas. Además puede aplicar a los componentes de diseño ciertos elementos HTML usando formatos predefinidos 375. Para más información sobre cómo especificar propiedades de presentación consulte la sección Archivos SPS: presentación 374.

© 2016-2022 Altova GmbH

4.3 Versiones XSLT y XPath

Un archivo SPS es básicamente una hoja de estilos XSLT y por eso es necesario establecer la versión XSLT de cada archivo SPS: 1.0, 2.0 o 3.0. Para ello haga clic en el icono pertinente de la barra de herramientas: , o . La opción elegida determinará:

- Cuál de los tres motores XSLT de StyleVision se utiliza para las transformaciones. StyleVision tiene motores XSLT 1.0, XSLT 2.0 y XSLT 3.0.
- Qué características XSLT (1.0, 2.0 o 3.0) están disponibles en la interfaz y están permitidas en el archivo SPS. Por ejemplo, XSLT 3.0 utiliza XPath 3.0, que es un lenguaje mucho más potente que XPath 1.0 (usado en XSLT 1.0) o XPath 2.0 (usado en XSLT 2.0). Además algunas características del SPS, como las tablas de contenido, solamente están disponibles con XSLT 2.0 y XSLT 3.0.

Nota: para poder generar resultados en formato texto la versión XSLT debe ser XSLT 2.0 or XSLT 3.0.

Transformación XSLT

Las transformaciones XSLT de StyleVision se utilizan para (i) generar vistas de resultados 39 en la interfaz y (ii) generar y guardar archivos de resultados 121 (HTML, texto y RTF) desde la interfaz gráfica de la aplicación 782 y desde StyleVision Server. El motor XSLT utilizado para transformaciones (el motor XSLT 1.0, 2.0 o 3.0 de Altova) corresponde a la versión XSLT seleccionada en el diseño SPS.

Funciones XSLT en la interfaz gráfica

Las funciones propias de cada versión XSLT dependen sobre todo del uso de la versión XPath adecuada (XPath 1.0 para XSLT 1.0, XPath 2.0 para XSLT 2.0 o XPath 3.0 para XSLT 3.0). Las expresiones XPath se usan mucho en StyleVision (sobre todo en características como los cálculos automáticos 293 y las plantillas condicionales 303) y hay mecanismos de la interfaz que necesitan (y que ayudan a generar) expresiones XPath para funcionar. Las funciones de la versión XPath adecuada aparecen en la interfaz automáticamente dependiendo de la versión XSLT seleccionada.

4.4 Compatibilidad con Internet Explorer

Para poder ver el diseño SPS en la vista **Diseño** y en las pestañas de vista previa de StyleVision es necesario tener instalado Internet Explorer (IE) en el equipo. StyleVision es compatible con estas versiones de IE:

- Internet Explorer 5.5 o superior
- Internet Explorer 6.0 o superior tiene una mayor compatibilidad con XML.
- Internet Explorer 9 o superior ofrece características adicionales, como compatibilidad con más formatos de imagen y estilos CSS nuevos. Si quiere usar estas características en su diseño, le recomendamos utilizar IE9.

Características de IE9 compatibles con StyleVision

StyleVision es compatible con estas características de IE9 (o superior):

- Formatos de imagen compatibles: TIFF, JPEG XR y SVG. (Los documentos SVG deben estar en formato XML y deben estar en el espacio de nombres SVG.) Las versiones anteriores de IE no son compatibles con estos formatos de imagen. Para ver una lista de formatos de imagen y sus formatos de salida compatibles consulte el apartado Tipos de imágenes y formato de salida 185.
- Compatibilidad con los nuevos estilos CSS (incluidos los estilos CSS3) que aparece a continuación. La aplicación de estos estilos se limita a la vista **Authentic** y al formato de salida HTML.

  - `background-clip`
  - `background-origin`
  - `background-size`
  - `box-sizing`
  - `box-shadow`
  - `border-radius (border-*-radius)`
  - `font-stretch`
  - `ruby-align`
  - `ruby-overhang`
  - `ruby-position`
  - `overflow-x, overflow-y`
  - `outline (outline-color, outline-style, outline-width)`
  - `text-align-last (partial)`
  - `text-overflow (partial)`

- Compatibilidad con la nueva función de longitud CSS `calc()`
- Compatibilidad con las nuevas funciones de color CSS `rgba()`, `hsl()` y `hsla()`
- Compatibilidad con las nuevas unidades de longitud CSS `rem`, `vw`, `vm`, `vh` y `ch`
- Los elementos HTML5 compatibles con IE9 se pueden insertar en el diseño como elementos definidos por el usuario 151.

Vista Diseño y versiones de Internet Explorer

Puede configurar la vista **Diseño** para una versión concreta de IE. Esto se hace en el cuadro de diálogo Propiedades 787 (**Archivo | Propiedades** 787) 787, donde puede indicar con qué versión de IE debe ser compatible la vista **Diseño**. La versión seleccionada tiene estos efectos:

- Todos los estilos CSS que pueden ser representados por la versión de IE seleccionada estarán disponibles automáticamente en la ventana Estilos de StyleVision. (Recuerde, que si selecciona la

 © 2016-2022 Altova GmbH

versión IE9, es obligatorio tener IE9 instalado. De lo contrario no podrá usar los estilos CSS compatibles con IE9 en la interfaz de diseño.) Por ejemplo, si tiene IE9 instalado y selecciona la versión IE9 en el cuadro de diálogo "Propiedades", los estilos CSS3 compatibles con IE9 estarán disponibles en la ventana Estilos.

- Los elementos HTML correspondientes a la versión de IE seleccionada se pueden insertar en el diseño como formatos predefinidos 137 o como elementos definidos por el usuario 151. El elemento HTML se representará correctamente en la vista **Authentic** y **HTML** tal y como se represente dicho elemento en la versión de IE instalada. Por ejemplo, si tiene IE9 instalado y selecciona la versión IE9 en el cuadro de diálogo "Propiedades", los elementos HTML5 compatibles se representarán correctamente en la vista **Authentic** y **HTML**.

Configurar la vista Diseño para una versión de IE concreta

Para configurar la vista **Diseño** para una versión de IE concreta, seleccione el comando de menú **Archivo | Propiedades**. En el cuadro de diálogo "Propiedades" que aparece, abra la pestaña *Formato de salida* y seleccione la versión IE pertinente. Para más información consulte el apartado Archivo | Propiedades 787 de la *Referencia del usuario*.

Compatibilidad de diseños SPS antiguos con IE9

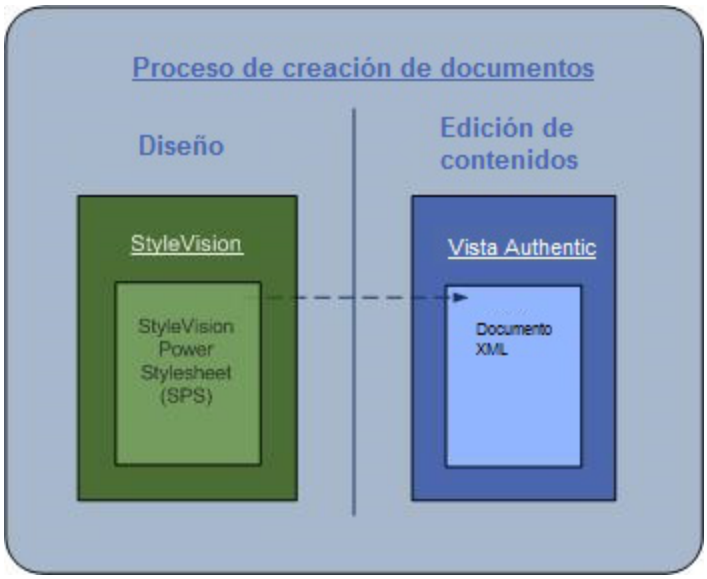
Si abre un diseño SPS creado para una versión anterior de IE y tiene instalada la versión IE9 o una versión superior en el equipo, StyleVision detectará la versión nueva y le preguntará si desea cambiar la configuración de compatibilidad a la versión IE9. Si actualiza la versión de compatibilidad podrá utilizar características adicionales en la vista Diseño (ver más arriba). El aspecto que muestra el documento en la vista Diseño, en la vista **Authentic** y en la vista previa **HTML** no cambiará excepto en el caso de las **columnas de tablas**, que en IE9 se tratan de forma diferente. Si cambia a la versión de compatibilidad IE9, asegúrese de que las columnas de tabla se generan bien. Si no es así, modifique las propiedades de las columnas o cambie la versión de compatibilidad de IE a la versión seleccionada previamente en el cuadro de diálogo Propiedades 787.

4.5 Archivos SPS y la vista Authentic

Uno de los usos principales de los documentos SPS es controlar la entrada de datos y la presentación de documentos XML en la vista Authentic [27], una vista de documentos disponible en los productos de Altova. Con ayuda de la vista Authentic los usuarios que no están familiarizados con XML pueden introducir y editar datos del documento XML con facilidad y sin introducir errores.

El proceso de creación de documentos y de edición en la vista Authentic tiene dos fases:

- ***Diseño de documentos:*** la vista Authentic del documento XML, que es una vista gráfica, se diseña en StyleVision. El documento de diseño es un archivo SPS. Este archivo SPS no sólo procesa el documento XML para su presentación en la vista Authentic y en otros formatos de salida, sino que además ofrece mecanismos para **introducir datos en el archivo XML o en la BD** desde la vista Authentic**.**



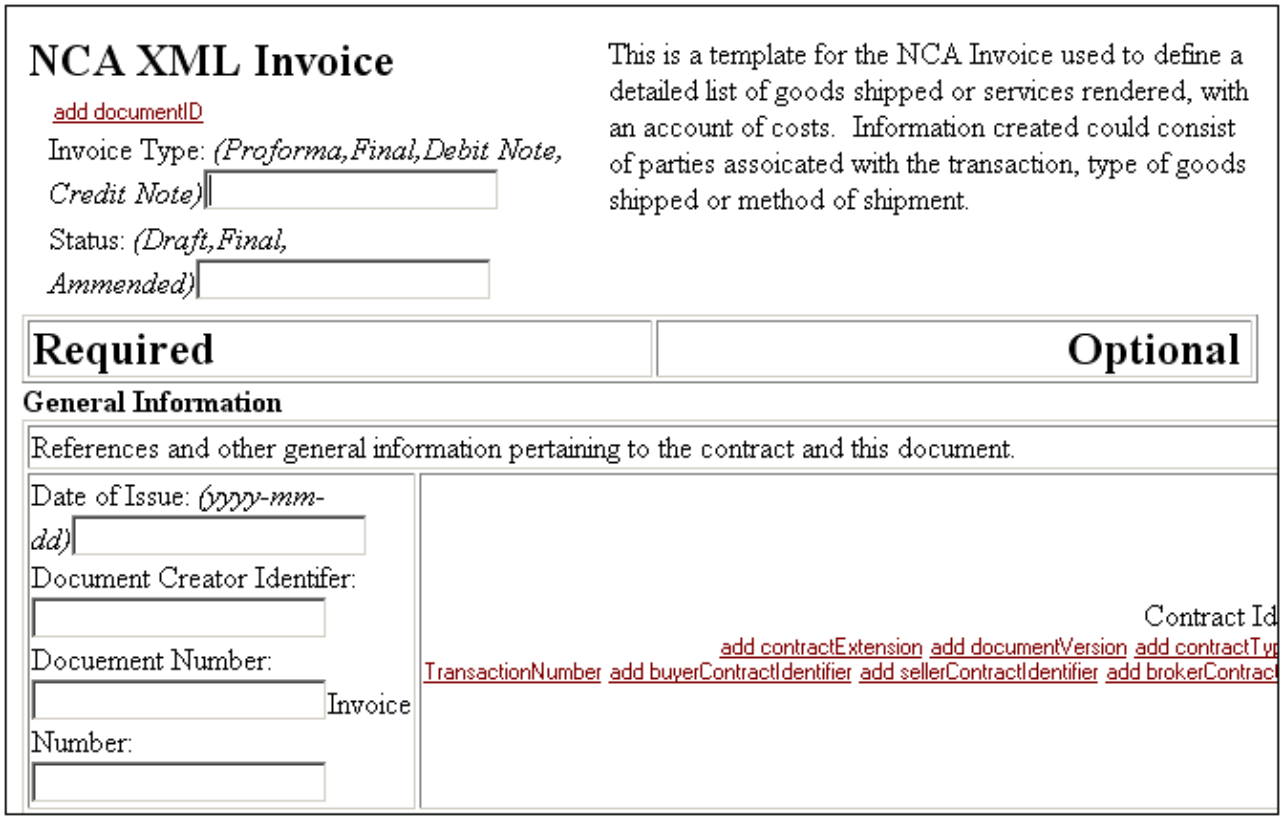
- ***Edición de contenidos:*** este archivo SPS creado en la fase de diseño de documentos se vincula al documento XML que se debe editar. (El documento XML debe ser válido con respecto al esquema en el que se basa el SPS.) El documento XML vinculado al SPS se presenta de forma gráfica en la vista Authentic de los productos de Altova. Cuando se crea un documento XML nuevo, puede asignarle un SPS y después editarlo en la vista Authentic con ayuda de la plantilla de documento (archivo de plantilla XML [30]) y los controles especificados en el SPS. Si abre un documento XML ya existente y le asigna un SPS, los datos del documento pueden verse y editarse en la vista Authentic según el diseño del SPS.

El usuario de la vista Authentic no necesita conocer el archivo XML ni el esquema utilizados para el documento. La vista Authentic del documento facilita la edición de contenidos y, por tanto, es obligación del diseñador del SPS crear una vista Authentic sencilla para el usuario final. Para más información sobre la vista Authentic, consulte la documentación de la vista Authentic del Manual del usuario de XMLSpy o Authentic Desktop.

 © 2016-2022 Altova GmbH

Archivos SPS para esquemas de estándares sectoriales

Los programas de Altova que incluyen la vista Authentic vienen con varios archivos SPS de muestra para varios esquemas de estándares sectoriales. Gracias a estos esquemas el usuario puede crear un documento XML inmediatamente en la vista Authentic. Por ejemplo, la imagen siguiente muestra parte de la vista Authentic del estándar DocBook.



Si quiere puede personalizar los SPS para los estándares sectoriales, disponibles en la carpeta `Examples/IndustryStandards` de la carpeta de su aplicación de Altova.

4.6 Sincronizar StyleVision y Authentic

En cada versión de StyleVision se añaden nuevas características que aumentan la potencia y la flexibilidad de la vista Authentic [27]. Sin embargo, es importante tener en cuenta que:

- Los archivos SPS creados con versiones más recientes de StyleVision pueden ser incompatibles con versiones anteriores de la vista Authentic.
- Las características nuevas de un archivo SPS (si el archivo se creó con una versión más reciente de StyleVision) solamente pueden ser interpretadas por una versión de la vista Authentic igual de reciente o posterior.

Por tanto, si usa una versión más reciente de StyleVision para crear un archivo SPS, deberá sincronizar sus productos de Altova que incluyen la vista Authentic [27] con esta versión de StyleVision. Esto significa que si, por ejemplo, utilizó **StyleVision 2008 Release 2** para crear un archivo SPS, entonces debe usar **Authentic Desktop 2008 Release 2** (o cualquier otro producto de la misma versión que tenga la vista Authentic) para editar correctamente este archivo SPS.

Recuerde que las versiones más recientes de los productos que incluyen la vista Authentic [27] son capaces de interpretar archivos SPS creados con versiones previas de StyleVision.

¿Qué pasa si un archivo SPS ya implementado se modifica con una versión más reciente de StyleVision?

Imagine que implementa un archivo SPS entre varios usuarios de la vista Authentic y posteriormente se añaden características nuevas al SPS con una versión más reciente de StyleVision. En este caso el desarrollador deberá sincronizar la implementación siguiendo estos pasos:

1. El desarrollador obtiene una clave de licencia para la nueva versión de la vista Authentic.
2. El desarrollador prueba los cambios hechos en el SPS usando la nueva versión de StyleVision y de la vista Authentic.
3. La versión nueva del producto que incluye la vista Authentic [27] se distribuye a todos los usuarios de la vista Authentic.
4. Ahora se puede distribuir el SPS modificado a los usuarios de la vista Authentic.

© 2016-2022 Altova GmbH

4.7 Archivos generados

En StyleVision puede generar hojas de estilos y archivos de salida con el comando **Archivo | Guardar archivos generados** 782 o con StyleVision Server. Además, si solamente quiere validar o transformar XML mediante XSLT, puede hacerlo con RaptorXML(+XBRL) Server 703 directamente y sin necesidad de llamar a StyleVision.

Con StyleVision puede generar estos archivos:

- Hojas de estilos XSLT basadas en el diseño SPS. Puede generar hojas de estilos XSLT para documentos de salida en formato HTML, texto y RTF.
- Archivos de salida generados al procesar el archivo XML de trabajo 30 asignado al SPS con las hojas de estilos XSLT generadas a partir del SPS.

El marcado de los resultados/documentos de salida está en el SPS. Los datos para los resultados/documentos de salida están en el documento XML o en la BD. Es la hoja de estilos XSLT la que combina el marcado con los datos para generar los resultados/documentos de salida. Tanto las hojas de estilos XSLT como los resultados pueden verse en las vistas de resultados 39 de StyleVision.

Nota: si quiere añadir un espacio de nombres a un SPS o a una hoja de estilos generada con un SPS, el espacio de nombres debe añadirse en el elemento `schema` de nivel superior del esquema XML en el que se basa el SPS.

Sitio web de Altova: Informe de datos XML

Documentos de salida

A continuación destacamos aspectos importantes de los documentos que puede generar con StyleVision:

- ***Hojas de estilos para HTML y documentos de salida HTML:*** (1) el formato y el diseño de los documentos HTML generados con StyleVision es idéntico al que tienen en la vista previa HTML de la aplicación y casi idéntico al de la vista Authentic del documento XML. (2) Los dispositivos de entrada de datos (campos de entrada de texto, casillas de verificación, etc.) del archivo HTML no permiten introducir datos. El objetivo de estos dispositivos de entrada de datos es introducir datos XML en la vista Authentic y no se pueden usar para introducir datos en el documento HTML, aunque se transformen en elementos HTML gráficos equivalentes.
  Sitio web de Altova: Conversión de datos XML en HTML

- ***Hojas de estilos para RTF y documentos de salida RTF:*** (1) el diseño RTF necesita ciertas definiciones para medios impresos. Estas definiciones (diseño de la portada, paginación, etc.) se especifican en la ventana Propiedades 59 y en la ventana Estructura del diseño 52. (2) Si se usaron dispositivos de entrada de datos en el SPS, estos dispositivos se representan, siempre que sea posible, como elementos gráficos en la página RTF. Si un dispositivo de entrada de datos no se puede simular bien como elemento gráfico (p. ej. las casillas de verificación), se utiliza otra presentación.

- ***Hojas de estilos para texto y documentos de salida en formato texto:*** como el texto plano no usa formato, las propiedades de estilo del diseño no se pasan a los documentos de salida en formato texto.

Documentos RTF de salida

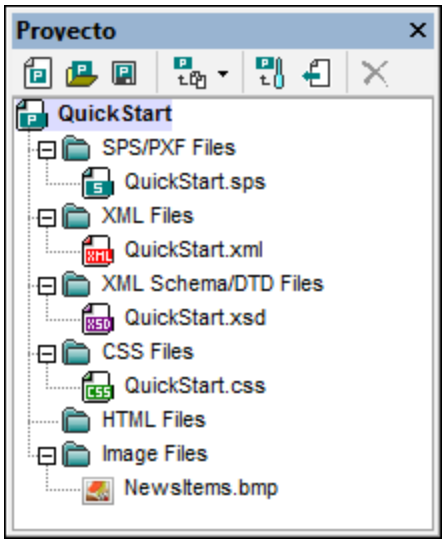
Los documentos de salida RTF se generan a partir del archivo XML en un solo paso, procesando el documento XML con la hoja de estilos XSLT-para-RTF generada con el archivo SPS. Las propiedades del documento de salida RTF se definen en el archivo SPS y puede obtener una vista previa en la pestaña de la vista **RTF**. Para obtener el archivo RTF debe generarlo con el comando **Archivo | Guardar archivos generados** 782 o desde StyleVision Server.

Nota: si hay un problema con la vista previa incrustada, StyleVision intenta abrir el documento de vista previa en una aplicación externa (como MS Word o Adobe Reader) y emite un mensaje de error sobre la vista previa incrustada. Si el documento de vista previa se abre en una aplicación externa, será necesario cerrar la aplicación externa antes de volver a generar el documento de salida temporal. De lo contrario, aparece un mensaje diciendo que el archivo está siendo utilizado por otro proceso. Antes de cerrar el diseño SPS debe cerrar también la aplicación externa. De lo contrario, StyleVision no podrá cerrar el documento de salida temporal dado que la aplicación externa bloquea el documento.

 © 2016-2022 Altova GmbH

4.8 Proyectos de StyleVision

Puede reunir todos los archivos relacionados en un archivo de proyecto en la ventana Proyecto (*imagen siguiente*). La idea es poder acceder con facilidad a los archivos que componen el proyecto mientras se diseña el archivo SPS. Por ejemplo, puede arrastrar un archivo SPS desde la ventana Proyecto hasta la ventana Estructura del esquema y crear un módulo con él. O puede arrastrar un archivo de imagen hasta el diseño y crear una imagen estática con él. O puede arrastrar una hoja de estilos CSS hasta la ventana Repositorio de estilos y añadirla al diseño como hoja de estilos externa.



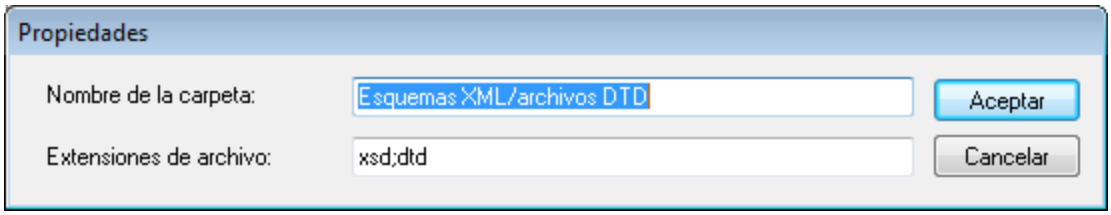
Crear y guardar el proyecto

Puede crear un proyecto nuevo con el comando **Proyecto | Proyecto nuevo**. Un proyecto nuevo contiene por defecto varias carpetas para tipos de archivo distintos (*imagen anterior*). Puede asignar tipos de archivo a cada carpeta desde el cuadro de diálogo "Propiedades" de la carpeta. Para dar un nombre a un proyecto nuevo debe guardarlo (con la extensión `.svp`). Una vez guardado puede cambiar el nombre del proyecto cambiando el nombre del archivo de proyecto en su ubicación en el sistema de archivos.

Carpetas de proyecto

Puede añadir carpetas nuevas tanto a la carpeta de proyecto principal como a las carpetas y subcarpetas de la carpeta de proyecto principal, creando tantos niveles de carpetas como quiera. Puede añadir tres tipos de carpetas: (i) carpetas de proyecto, (ii) carpetas externas (buscando y seleccionado una carpeta del sistema de archivos) y (iii) y carpetas web externas (añadiendo una dirección URL). Todas estas carpetas se añaden a la carpeta de proyecto principal con estos comandos del menú **Proyecto**: (i) **Agregar carpeta de proyecto al proyecto...**, (ii) **Agregar carpeta externa al proyecto...** y (iii) **Agregar carpeta web externa al proyecto...**. Para añadir estas carpetas a una carpeta o subcarpeta del proyecto, haga clic con el botón derecho en la carpeta y seleccione el comando pertinente en el menú contextual que aparece.

Después puede asignar tipos de archivos a la carpeta en su cuadro de diálogo "Propiedades" (*imagen siguiente*). Para abrir este cuadro de diálogo haga clic con el botón derecho en la carpeta y seleccione el comando **Propiedades**.

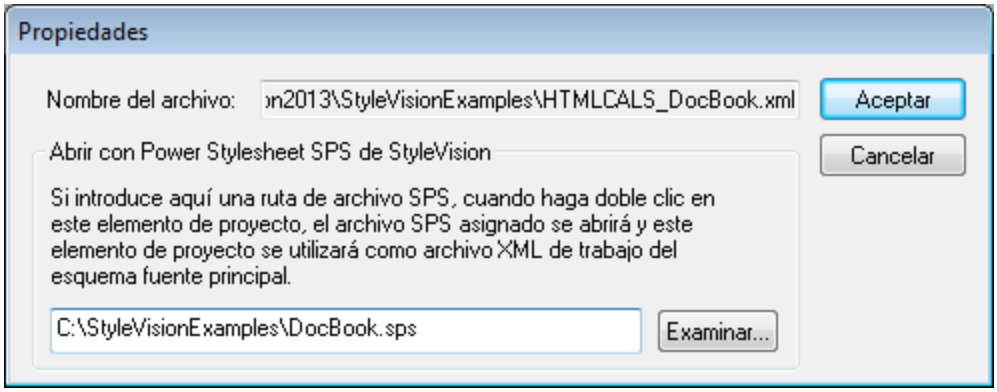


En este cuadro de diálogo puede editar el nombre de la carpeta y las extensiones de archivo asignadas a la carpeta (es obligatorio separar con punto y coma las extensiones de archivo). Los nombres de carpeta de proyecto también se pueden editar seleccionando la carpeta en la ventana Proyecto, pulsando **F2** y editando el nombre. Si una carpeta tiene asignados tipos de archivos (extensiones) y añade archivos al proyecto con el comando **Agregar archivos al proyecto**, a la carpeta se añaden automáticamente los archivos que tengan dichas extensiones. Si varias carpetas tienen asignadas las mismas extensiones de archivo, el archivo se añade a la primera de dichas carpetas. Puede cambiar el orden de las carpetas en la ventana Proyecto mediante operaciones de arrastrar y colocar. Sin embargo, las carpetas de primer nivel (es decir, el nivel inmediatamente inferior a la carpeta de proyecto principal) tienen este orden: (i) carpetas de proyecto, (ii) carpetas externas y (iii) carpetas web externas.

Archivos de proyecto

Puede añadir archivos tanto a la carpeta de proyecto principal como a sus carpetas y subcarpetas. Los archivos se añaden con estos comandos del menú **Proyecto**:

- ***Agregar archivos al proyecto:*** cada archivo seleccionado se añade a la primera carpeta que tenga definido el tipo de extensión del archivo.
- ***Agregar recurso global al proyecto:*** se añade un archivo a través de un recurso global.
- ***Agregar URL al proyecto:*** se añade un archivo por medio de su URL (que se define en el cuadro de diálogo "Agregar URL al proyecto").
- ***Agregar archivo activo al proyecto:*** el archivo (SPS) activo se añade a la primera carpeta de la ventana Proyecto que tenga definido el tipo de extensión `.sps`.
- ***Agregar archivo activo y relacionados:*** el archivo (SPS) activo y sus archivos relacionados (como los esquemas, archivos XML de trabajo, CSS, imágenes estáticas, etc.) se añaden al proyecto y en las carpetas correspondientes (según el tipo de extensión definido para cada carpeta). Con este comando puede agrupar en un solo proyecto todos los archivos relacionados con un diseño.
- ***Propiedades:*** si selecciona un archivo del proyecto y hace clic en el icono **Propiedades** de la ventana Proyecto o selecciona el comando **Propiedades** del menú contextual, se abre una ventana que muestra la ubicación del archivo. Si el archivo seleccionado es un archivo XML y hace clic en **Propiedades**, puede seleccionar el archivo SPS con el que se debe asociar el XML (*imagen siguiente*).



 © 2016-2022 Altova GmbH

Cuando se asocia un archivo SPS a un archivo XML que está en un proyecto, al hacer doble clic en el XML se abre el archivo SPS asociado al XML (el XML se asigna al SPS como archivo XML de trabajo). Esta característica permite trabajar con más rapidez, al no tener que buscar el SPS ni el archivo XML de trabajo en el sistema de archivos.

Los comandos mencionados más arriba añaden archivos al proyecto principal según las extensiones de tipo de archivo asignadas a las carpetas de la carpeta de proyecto principal. Para agregar archivos a una carpeta concreta, haga clic con el botón derecho en la carpeta y seleccione el comando pertinente en el menú contextual. Dentro de cada carpeta los archivos se enumeran alfabéticamente. No olvide que puede arrastrar archivos de una carpeta a otra. Para ver la ubicación de un archivo, haga clic con el botón derecho y seleccione **Propiedades** en el menú contextual.

Recursos globales

A las carpetas del proyecto puede añadir recursos globales 444 de tipo archivo o de tipo carpeta. Un recurso global de tipo archivo es un alias para un archivo. Un alias puede tener varias configuraciones y cada configuración apunta a un archivo distinto. Así, si se utiliza un recurso global en un proyecto, puede enlazar con uno de los recursos de destino, dependiendo de la configuración que está activa en StyleVision. Del mismo modo, un recurso de tipo global es un alias que puede enlazar con una de las carpetas de destino, dependiendo de la configuración que está activa. Si usa un recurso global de tipo carpeta en el diseño para identificar un archivo (digamos un archivo XML de trabajo o un archivo CSS), el recurso global de tipo carpeta identifica una carpeta solamente. La ruta de la carpeta al archivo necesario debe especificarse. Para más información sobre el uso de recursos globales, consulte Usar recursos globales 455.

Arrastrar y colocar

En la ventana Proyecto puede arrastrar una carpeta dentro de otra carpeta. Los archivos se pueden arrastrar a cualquier carpeta, pero no se puede cambiar de posición dentro de su carpeta actual (porque los archivos se enumeran alfabéticamente). Además, puede arrastrar archivos y carpetas desde el explorador de archivos de Windows hasta la ventana Proyecto.

Usar proyectos

Los archivos, dependiendo del tipo, se pueden usar de varias maneras. El menú contextual de cada archivo de la ventana Proyecto indica qué acciones se pueden realizar con el archivo seleccionado. Además, cuando se arrastra un archivo a una posición donde se puede ejecutar una acción, se abre un menú que contiene varios comandos. En algunos casos, el comando se ejecuta directamente nada más soltar el archivo en la posición de destino. A continuación describimos las acciones que se pueden realizar con cada tipo de archivo.

Archivos SPS

- **Abrir el diseño:** abre el archivo SPS en una nueva ventana de diseño. Este comando está disponible cuando se arrastra un archivo SPS desde la ventana Proyecto hasta la vista **Diseño**.
- **Importar como módulo:** importa el archivo SPS como módulo en el archivo SPS activo. El archivo importado se enumera bajo la entrada **Módulos** de la ventana Vista general del diseño. También puede importar el archivo como módulo si lo arrastra hasta la entrada **Módulos** de la ventana Vista general del diseño.

Archivos XML

- **Editar archivo en XMLSpy***:* abre el archivo XML en XMLSpy.

- **Crear diseño nuevo***:* crea un SPS nuevo. El esquema para el SPS es un esquema XML generado a partir del documento XML. El archivo XML de trabajo del SPS es el archivo XML. También puede arrastrar el archivo hasta la plantilla principal de la vista **Diseño** para usar este comando.
- **Asignar como archivo XML de trabajo**: asigna el archivo XML como archivo XML de trabajo del SPS. Para asignar el archivo como archivo XML de trabajo también puede arrastrar el archivo hasta la entrada **`XML de trabajo`** de la ventana Vista general del diseño.
- **Asignar como archivo de plantilla XML**: asigna el archivo XML como archivo de plantilla XML del SPS. Para agregar el archivo como archivo de plantilla XML también puede arrastrar el archivo hasta la entrada **Plantilla XML** de la ventana Vista general del diseño.

  Nota: al hacer doble clic en un archivo XML se ejecuta una de estas tres acciones (dependiendo de lo que se defina en la pestaña Proyecto del cuadro de diálogo Opciones [894]): (i) el archivo se abre en Altova XMLSpy, (ii) se crea un diseño nuevo basado en el archivo XML o (iii) aparece un aviso pidiendo al usuario que confirme qué acción se ejecuta.

Esquemas XML y archivos DTD

- **Editar archivo en XMLSpy***:* abre el esquema en XMLSpy.
- **Crear diseño nuevo***:* crea un SPS nuevo basado en el esquema seleccionado. Para crear un SPS nuevo con el esquema seleccionado como esquema fuente también puede arrastrar el archivo hasta la plantilla principal de la vista **Diseño**.
- **Asignar como archivo de esquema**: asigna el esquema seleccionado como esquema fuente del archivo SPS activo y reemplaza el esquema fuente actual. Este comando es ideal para cambiar de esquema rápidamente si, por ejemplo, el esquema cambió de ubicación o desea corregir la asignación de esquema. Para usar este comando también puede arrastrar el archivo a la entrada **Esquema** de la ventana Vista general del diseño.

  Nota: al hacer doble clic en un esquema se ejecuta una de estas tres acciones (dependiendo de lo que se defina en la pestaña Proyecto del cuadro de diálogo Opciones [894]): (i) el esquema se abre en Altova XMLSpy, (ii) se crea un diseño nuevo basado en el esquema o (iii) aparece un aviso pidiendo al usuario que confirme qué acción se ejecuta.

Archivos CSS

- **Editar archivo en XMLSpy***:* abre el archivo CSS en XMLSpy.
- **Importar al repositorio de estilos**: añade el archivo CSS a la lista de archivos CSS externos del Repositorio de estilos (la entrada **`Externos`** del repositorio). Para importar el archivo al repositorio de archivos también puede arrastrar el archivo hasta la entrada **`Externos`** de la ventana Repositorio de estilos.

Archivos HTML

- **Editar archivo en XMLSpy***:* abre el archivo HTML en XMLSpy.
- **Abrir:** abre el archivo HTML en el explorador web predeterminado.
- **Crear diseño nuevo**: crea un SPS nuevo, donde puede crear un esquema basado en el documento HTML. También puede arrastrar el archivo HTML hasta la vista **Diseño** para usar este comando.

Archivos de imagen

- Si pasa el cursor del ratón sobre un archivo de imagen en una carpeta de proyecto podrá ver una vista previa de esa imagen (formatos **`.png`**, **`.jpeg`**, **`.gif`**, **`.bmp`**, **`.tiff`** e **`.ico`**). Haga doble clic en la imagen para abrira en el visor/editor de imágenes predeterminado.

 © 2016-2022 Altova GmbH

- **Abrir**: abre el archivo de imagen en el visor/editor de imágenes predeterminado.
- **Insertar imagen en el diseño**: inserta la imagen como imagen estática en el SPS. También puede arrastrar la imagen hasta su posición de destino.

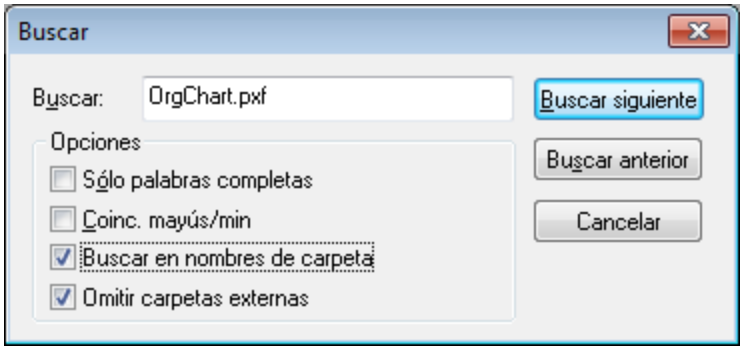
Todos los tipos de archivo

- **Explorar la carpeta contenedora**: abre la carpeta donde está el archivo seleccionado en una ventana del explorador de archivos de Windows.
- **Cortar, Copiar, Pegar, Eliminar**: estos comandos cortan y copian el archivo seleccionado en el portapapeles, pegan el archivo y lo eliminan. Estos comandos también se pueden ejecutar en varios archivos a la vez.
- **Seleccionar todo**: selecciona todos los archivos del proyecto.
- **Propiedades**: abre el cuadro de diálogo "Propiedades", donde aparece la ubicación del archivo.

Buscar en el proyecto

Puede buscar archivos y carpetas en el proyecto usando su nombre o parte del nombre. Si la búsqueda encuentra resultados, los archivos y carpetas que coincidan con el término de búsqueda se resaltan uno por uno.

Para iniciar una búsqueda, seleccione en la ventana Proyecto la carpeta de proyecto donde desea buscar 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 definir las opciones de búsqueda.



Estas son las opciones de búsqueda disponibles en el cuadro de diálogo "Buscar":

- *Sólo palabras completas*: la cadena de búsqueda debe coincidir con una palabra entera del nombre de archivo o carpeta. En los nombres de archivo se considera una palabra completa la parte del nombre que aparece antes y después del punto.
- *Coincidencia de mayúsculas/minúsculas*: la cadena de búsqueda debe coincidir exactamente con el nombre de archivo o carpeta.
- *Buscar en nombres de carpeta*: los nombres de carpeta se añaden en la búsqueda. Si no marca esta opción, solamente se busca en nombres de archivo.
- *Omitir carpetas externas* 806: las carpetas externas se pueden incluir o excluir de la búsqueda. Las carpetas externas son carpetas reales del sistema o de la red (es decir, no son carpetas del proyecto, que se crean dentro del proyecto y no en el sistema de archivos).

Si la búsqueda encuentra resultados, el primer resultado se resalta en la ventana Proyecto. Después puede buscar los demás resultados haciendo clic en los botones **Buscar siguiente** y **Buscar anterior** del cuadro de diálogo "Buscar".

4.9 Catálogos en StyleVision

El mecanismo de catálogos XML permite obtener archivos desde carpetas locales, lo que incrementa la velocidad de procesamiento, además de mejorar la portabilidad de los documentos, ya que solo es necesario cambiar los URIs del archivo de catálogo. Consulte el apartado Funcionamiento de los catálogos 128 para más detalles.

Los productos XML de Altova usan un mecanismo de catálogo para cargar y acceder rápidamente a archivos de uso habitual, como archivos DTD y esquemas XML. Este mecanismo se puede personalizar y ampliar, y viene explicado en los apartados Estructura del catálogo en StyleVision 129 y Personalizar catálogos 131. El apartado Variables para ubicaciones del sistema en Windows 132 incluye las variables de Windows de las ubicaciones comunes del sistema. Estas variables se pueden usar en los archivos de catálogo para localizar carpetas de uso habitual.

Este apartado se divide en varios subapartados:

- Funcionamiento de los catálogos 128
- Estructura del catálogo en StyleVision 129
- Personalizar catálogos 131
- Variables para ubicaciones del sistema en Windows 132

Para más información sobre los catálogos consulte la especificación XML Catalogs specification.

4.9.1 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 StyleVision:

```
<?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:

```
<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 StyleVision 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.

4.9.2 Estructura del catálogo en StyleVision

Al iniciarse, StyleVision 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.
- **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\StyleVision**.
- 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

4.9.3 Personalizar catálogos

Cuando cree entradas en el archivo **CustomCatalog.xml** (o en cualquier otro archivo de catálogo que sea leído por StyleVision), 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 StyleVision 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 StyleVision 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 StyleVision 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 StyleVision 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.

Especificaciones de XML Schema

La información de la especificación XML Schema está integrada en StyleVision 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 StyleVision para sus ayudantes de entrada, en caso de que el usuario quiera crear documentos basados en estas recomendaciones.

4.9.4 Variables para ubicaciones del sistema en Windows

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`
`%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.

© 2016-2022 Altova GmbH

%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.

5 Archivos SPS: contenido

Este apartado describe en detalle los procedimientos clave necesarios para crear y editar los componentes que alojan datos XML en el diseño SPS. Los procedimientos se enumeran más abajo y se describen en los apartados de esta sección. Estos mecanismos se usan para diseñar cualquier tipo de plantilla: principal 265, global 266 o con nombre 278.

- Insertar contenido XML como texto 135. Para insertar datos XML en el diseño arrastre los nodos pertinentes (elemento, atributo, tipo o CDATA) hasta la vista **Diseño** y cree `(contenido)` o `(resto-de-contenido)` a partir de ellos.
- Insertar contenido de MS Word 143
- Plantillas definidas por el usuario 148
- Elementos definidos por el usuario, Bloques de texto XML definidos por el usuario 151
- Trabajar con tablas. 155 El diseñador del SPS, puede insertar tablas en el diseño SPS (tablas estáticas) o crear tablas basadas en subestructuras del documento XML. Además del diseñador del SPS, también puede insertar tablas el usuario de la vista Authentic.
- Crear listas 178. Listas estáticas (la estructura de lista se introduce directamente en el diseño SPS) y listas dinámicas (la subestructura del documento XML crea una lista en la que se pueden ordenar los datos).
- Uso de contenido gráfico 183. Puede insertar contenido gráfico en el diseño SPS utilizando varios métodos para determinar el URI de destino (estático, dinámico, ambos combinados e identificadores URI de entidades sin analizar.
- Usar dispositivos de entrada de datos (o controles de formulario) 190. El usuario de la vista Authentic puede introducir datos XML con ayuda de dispositivos de entrada, como campos de entrada y cuadros combinados. Esto ayuda al usuario y a la vez limita la entrada de datos. Puede crear dispositivos de entrada de datos a partir de los nodos del documento XML.
- Hipervínculos 199.
- Códigos de barras 200.
- Módulos de diseño 204.
- Función 215 **Convertir en** 215. Esta función permite seleccionar otro nodo como coincidencia para una plantilla y permite cambiar el tipo de contenido del nodo.

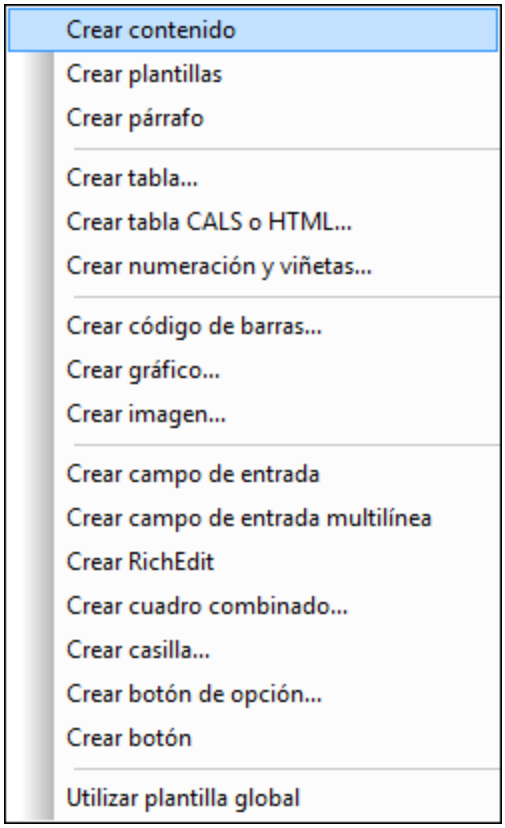
Temas relacionados

- Archivos SPS: características avanzadas 292
- Archivos SPS: otras funciones 443

© 2016-2022 Altova GmbH

5.1 Insertar contenido XML como texto

Para insertar los datos de un nodo del documento XML en el diseño, arrastre el nodo de esquema pertinente desde la ventana Estructura del esquema hasta la vista **Diseño**. Al soltar el nodo de esquema en el diseño aparece un menú donde puede elegir qué tipo de contenido se crea a partir del nodo (*imagen siguiente*).



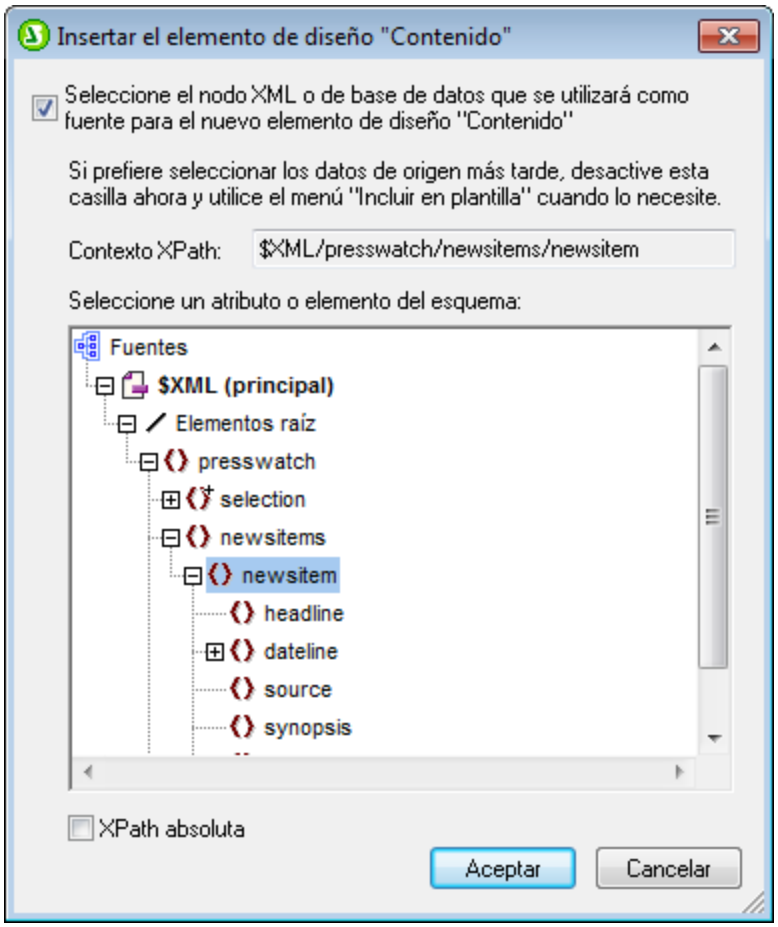
Tipos de nodos de esquema

Hay tres tipos de nodos de esquema que se pueden arrastrar hasta el diseño desde la ventana Estructura del esquema: (i) nodos de elemento, (ii) nodos de atributo y (iii) nodos de tipo de datos.

Usar el icono "Insertar contenido" de la barra de herramientas

El icono **Insertar contenido** de la barra de herramientas Insertar elementos de diseño 755 sirve para insertar el contenido de un nodo en el diseño. Utilice este icono de la siguiente manera:

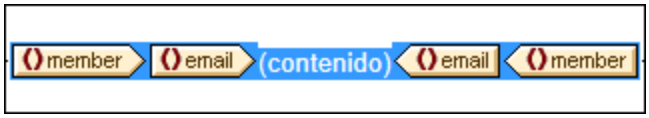
1. Haga clic en el icono **Insertar contenido**.
2. Haga clic en la posición del diseño donde desea insertar el contenido. Aparece el cuadro de diálogo de selección de nodos (*imagen siguiente*).



3. En el cuadro *Contexto XPath* aparece el contexto de la posición del diseño que eligió para insertar el contenido. Seleccione el nodo con el que desea crear contenido.
4. Haga clic en **Aceptar**. Se crea el marcador de posición `(contenido)`. Si el nodo seleccionado no es el nodo de contexto, alrededor del marcador `(contenido)` se crean etiquetas de plantilla adicionales con la ruta de acceso del nodo seleccionado.

Generar el contenido de texto de los nodos

Para dar salida al contenido de texto de un nodo es necesario crear contenido con el nodo. Cuando creamos contenido con un nodo, el diseño tiene este aspecto:

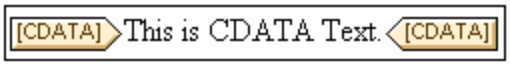


En la imagen anterior se creó contenido a partir del elemento `Email`. El resultado mostrará el contenido de texto de `Email`. Si `Email` tiene elementos descendientes, como `Bold` e `Italics`, el contenido de texto de los elementos descendientes también se generará en el resultado porque forman parte del contenido de `Email`. Recuerde que los nodos de atributo de `Email` no se consideran nodos secundarios y que, por tanto, el contenido de los nodos de atributo no se incluye en el contenido de `Email`. Es decir, los nodos de atributo no se procesan si no se insertan explícitamente en el diseño.

 © 2016-2022 Altova GmbH

Secciones CDATA

Las secciones CDATA del documento XML se procesan y se incluyen en los resultados y en la vista Authentic aparecen señaladas con etiquetas, si el marcado está activado (comando de menú **Authentic | Marcado** 858). También puede insertar secciones CDATA en el documento XML desde la vista Authentic (con ayuda del menú contextual).



Nota: en la vista Authentic no se pueden insertar secciones CDATA dentro de campos de entrada (es decir, dentro de cuadros de texto ni cuadros de texto multilínea). Solamente se pueden insertar dentro de elementos que aparezcan en la vista Authentic como componentes de contenido de texto.

En esta sección

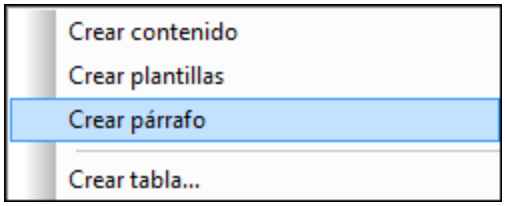
Los apartados de esta sección describen el procedimiento de inserción de contenido XML como texto:

- Cómo marcar con un formato predefinido 137 el contenido de texto de un nodo al insertarlo.
- Cómo afecta la estructura del esquema fuente el uso de la vista Authentic 138.
- Cómo incluir y procesar nodos descendientes que no están incluidos en un nodo explícitamente (Resto-de-contenido) 141.

Nota: puede crear una **regla de plantilla vacía** si elimina el marcador de posición `(contenido)` de un nodo. Las reglas de plantilla vacías sirven para indicar que un nodo no se debe procesar.

5.1.1 Insertar contenido con formato predefinido

El contenido de texto de un nodo se puede insertar directamente con el marcado de uno de los formatos predefinidos de StyleVision. Para ello arrastre el nodo desde la ventana Estructura del esquema y suéltelo en la posición de destino. Al soltarlo aparece un menú contextual donde debe seleccionar **Crear párrafo** (*imagen siguiente*).



Para cambiar de formato predefinido seleccione la etiqueta de formato predefinido y elija otro formato en el cuadro combinado de Formato de la barra de herramientas 750 (*imagen siguiente*) o con el comando de menú **Insertar | Formato**.

También puede cambiar de formato predefinido cambiando el valor de la propiedad `tipo de párrafo` en la ventana Propiedades (grupo de propiedades `párrafo`) o cambiando de tipo de párrafo con el comando de menú contextual 274 **Incluir en | Párrafo especial** 274 de la plantilla del nodo.

Cada tipo de párrafo tiene características de formato propias. Recuerde que el tipo de formato `pre` permite procesar retornos de carro y avances de línea tal y como están en lugar de normalizarlos como espacios en blanco.

Temas relacionados

- Símbolos de la vista Diseño 724
- Formatos predefinidos 375

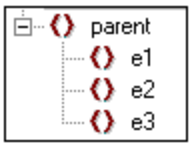
5.1.2 Agregar elementos en la vista Authentic

El modo en que se crean elementos en el diseño determina cómo responde la vista Authentic a ciertas acciones del usuario (p. ej. qué ocurre cuando el usuario pulsa la tecla **Tabulador** o cuando hace clic en el símbolo `Agregar...`). El primer paso consiste en determinar qué elementos se crean en la vista Authentic cuando el usuario añade un elemento. Es decir, cuando el usuario añade un elemento (al hacer clic en el icono **Insertar elemento** del ayudante de entrada Elementos, por ejemplo), ¿qué elementos secundarios se crean automáticamente?

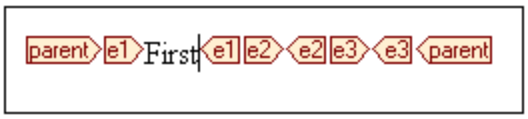
El punto más importante que debe tener en cuenta es que la vista Authentic sigue la estructura definida en el esquema subyacente. Por tanto, para asegurarse de que la vista Authentic implementa correctamente la estructura del esquema, debe tener en cuenta algunas reglas de diseño.

Modelo de contenido inequívoco

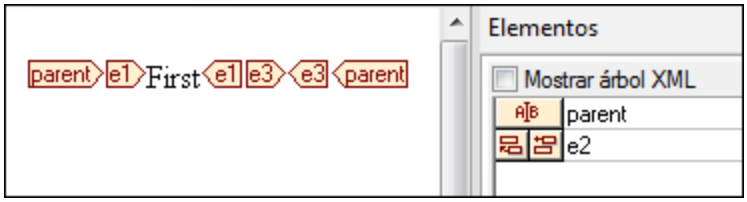
Se entiende que un modelo de contenido es inequívoco cuando está formado por una sola secuencia (con `maxOccurs=1`) de elementos secundarios (sin elementos `choice`, `group`, `substitution`, etc.). Cuando se trabaja con este tipo de modelo de contenido y el usuario añade un elemento, la secuencia de elementos secundarios se conoce inequívocamente y los secundarios se añaden automáticamente. En la imagen siguiente, por ejemplo, los tres elementos secundarios son obligatorios y solamente pueden aparecer una vez.



Cuando se añade el elemento primario `parent` en la vista Authentic, sus elementos secundarios se insertan automáticamente (*imagen siguiente*). Al pulsar la tecla **Tabulador** el usuario pasa al siguiente elemento de la secuencia.



Si el elemento `e2` fuera opcional, al añadir el elemento primario `parent` en la vista Authentic, los elementos `e1` y `e3` se insertarían automáticamente y el elemento `e2` aparecería en el ayudante de entrada Elementos, para que el usuario lo inserte si quiere (*imagen siguiente*). Al pulsar la tecla **Tabulador** en `e1` el usuario pasa a `e3`.



El ejemplo de modelo de contenido anterior es el único caso inequívoco según la vista Authentic. Todos los demás casos se consideran ambiguos y para que la vista Authentic puede aclarar las ambigüedades y mostrar los elementos deseados de forma eficaz, es necesario que el diseño cumpla cierta reglas básicas.

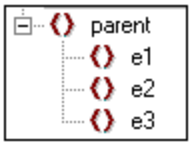
Modelo de contenido ambiguo

Para que la vista Authentic muestre los elementos correctamente mientras se edita un elemento XML, el diseño SPS debe cumplir ciertas reglas.

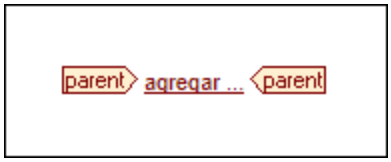
- Los elementos secundarios se presentan en el mismo orden en el que aparecen en el diseño.
- Para que la vista Authentic pueda distinguir unos elementos secundarios de otros, todos deben estar en el diseño en el orden necesario **y dentro de un solo nodo primario**. Si la relación entre los secundarios del mismo nivel debe conservarse en la vista Authentic, no se debe poner cada elemento secundario de un solo primario dentro de varias instancias del nodo primario.

Veamos ejemplos concretos de estas dos reglas:

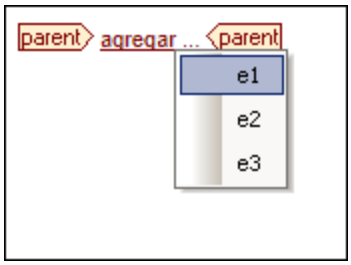
Imaginemos un modelo de contenido de un elemento primario llamado `parent` que está formado por una sola secuencia de elementos secundarios obligatorios. Este modelo de contenido es similar al modelo de contenido inequívoco que explicamos al principio, con una sola diferencia: la secuencia es opcional y, por tanto, el modelo es ambiguo porque la secuencia no es algo seguro. Si crea un documento de diseño como el que aparece en la imagen de abajo, habrá algunas ambigüedades en la vista Authentic.



La vista Authentic del elemento primario `parent` tendría este aspecto (porque la secuencia es opcional):

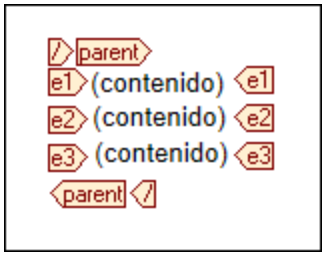


Al hacer clic en `Agregar...` se abre un menú contextual con los tres elementos secundarios:

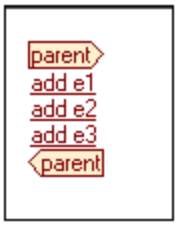


Si selecciona uno de ellos, el secundario elegido se inserta (*imagen siguiente*), pero la vista Authentic no inserta ninguno de los otros dos porque no puede aclara las ambigüedades. La vista tampoco ofrece la posibilidad de insertar los otros dos elementos:

Por tanto, la manera correcta de diseñar este modelo de contenido (según las reglas antes descritas) sería crear explícitamente los nodos necesarios en el orden deseado dentro del nodo primario. El documento de diseño tendría este aspecto:



Observe que los tres elementos secundarios aparecen ahora **dentro de un solo nodo primario** en la vista Authentic:



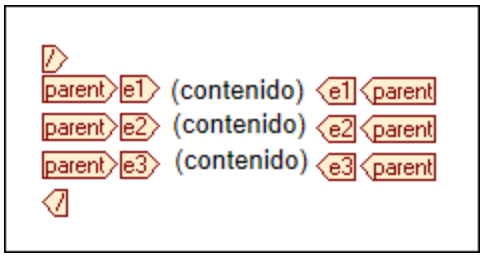
Ahora el usuario de la vista Authentic puede hacer clic en el comando `agregar` del elemento correspondiente para insertar el elemento y su contenido.

Nota:

- Si un elemento puede aparecer varias veces y se siguieron las reglas anteriores, el elemento aparecerá en el ayudante de entrada tantas veces como permita el número máximo de apariciones definido por el esquema (`maxOccurs`).
- Cuando se crea un elemento secundario dentro de un nodo primario distinto (*imagen siguiente*) no sólo se crean relaciones secundario-primario aisladas para cada elemento secundario cuyas instancias se

 © 2016-2022 Altova GmbH

crean así. Además aumenta el tiempo de procesamiento porque el nodo primario debe cruzarse otra vez para poder ubicar cada elemento secundario.



Temas relacionados

- Insertar contenido XML como texto 135
- Uso de la vista Authentic 662

5.1.3 Resto de contenidos

El marcador de posición `resto-de-contenido` aplica plantillas a todos los elementos secundarios restantes del elemento para el que se creó la plantilla. Imaginemos este ejemplo:

- El elemento `primario` tiene 4 elementos secundarios: `secundario1` - `secundario4`.
- En la plantilla del elemento `primario` se definió cómo se deben procesar los elementos `secundario1` y `secundario4`.

Es decir, solamente se procesan los elementos `secundario1` y `secundario4`. Los elementos `secundario2` y `secundario3` no se procesarán. Pero si insertamos el marcador de posición `resto-de-contenido` dentro de la plantilla del elemento `primario`, no solo se procesarán `secundario1` y `secundario4` con las reglas de procesamiento definidas en la plantilla. También se aplicarán plantillas a los elementos `secundario2` y `secundario3`. Si se definieron plantillas globales 265, se usarán las plantillas globales. De lo contrario, se aplicarán las plantillas integradas predeterminadas (para nodos de elemento, atributo y texto).

Importante: tenga cuidado cuando seleccione nodos para el marcador `resto-de-contenido`.

- Tal y como decíamos en el ejemplo, el marcador `resto-de-contenido` selecciona todos los nodos de elemento y de texto secundarios. Incluso los nodos secundarios no válidos del documento XML se procesarán.
- Los atributos de nodo no se seleccionan porque no son nodos secundarios, es decir no están en el eje secundario de XPath.
- Si se usa una plantilla global de un secundario en la plantilla del primario, entonces no se considera que el secundario se haya aplicado localmente. Como resultado, el marcador de posición `resto-de-contenido` también seleccionará dichos secundarios. Sin embargo, si una plantilla global de un secundario se copia localmente, su uso cuenta como uso local y el marcador de posición `resto-de-contenido` no selecciona el secundario.

Nota: puede crear una **regla de plantilla vacía** si elimina el marcador de posición `(contenido)` de un nodo. Las reglas de plantilla vacías sirven para indicar que no se debe procesar algún nodo.

Temas relacionados

- Insertar contenido XML como texto 135
- Símbolos de la vista Diseño 724
- Plantillas y fragmentos de diseño 265

© 2016-2022 Altova GmbH

5.2 Insertar contenido de MS Word

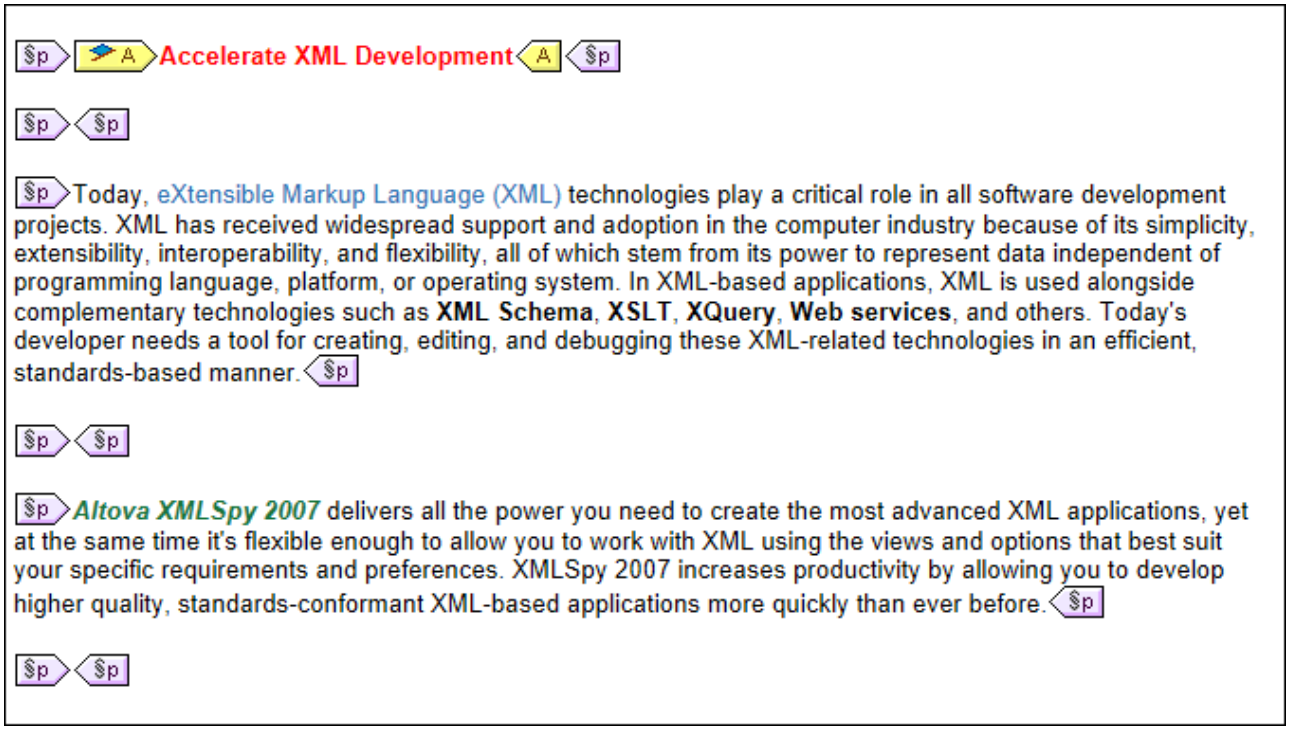
Si Microsoft Word 2007+ está instalado en el equipo, podrá pegar contenido de documentos Word en el diseño en forma de contenido estático. El contenido Word se insertará dentro de los componentes de diseño que correspondan y podrá aplicar las propiedades de formato de texto del contenido Word. Por ejemplo, el contenido de texto que esté en un bloque de párrafo Word se insertará dentro de un componente de tipo Párrafo 137 y el formato del texto se conservará (*imagen siguiente*).

Accelerate XML Development

Today, eXtensible Markup Language (XML) technologies play a critical role in all software development projects. XML has received widespread support and adoption in the computer industry because of its simplicity, extensibility, interoperability, and flexibility, all of which stem from its power to represent data independent of programming language, platform, or operating system. In XML-based applications, XML is used alongside complementary technologies such as **XML Schema**, **XSLT**, **XQuery**, **Web services**, and others. Today's developer needs a tool for creating, editing, and debugging these XML-related technologies in an efficient, standards-based manner.

Altova XMLSpy 2007 delivers all the power you need to create the most advanced XML applications, yet at the same time it's flexible enough to allow you to work with XML using the views and options that best suit your specific requirements and preferences. XMLSpy 2007 increases productivity by allowing you to develop higher quality, standards-conformant XML-based applications more quickly than ever before.

Contenido de Word.



Contenido de Word pegado en el diseño SPS. El formato de texto se conservó y también se aplicó un formato de tipo párrafo adecuado.

Nota: además de contenido de Word, en los diseños de StyleVision también puede pegar *cualquier otro tipo de contenido que se pueda pegar en documentos Word*. Por ejemplo, **tablas de MS Excel** y **contenido de páginas HTML**.

Nota: para crear un archivo SPS que incluya contenido estático de un documento Word entero, utilice el comando **Archivo | Nuevo | A partir de un archivo Word 2007+** 761.

Características Word compatibles

StyleVision es compatible con estas estructuras y formatos Word cuando se copia y pega contenido Word en el diseño:

- Texto con formato
  - *Diferentes fuentes, tamaños, grosores, estilos y decoración, etc.*
  - *Color*
  - *Color de fondo*
  - *Bordes alrededor del texto*
- Párrafos
- Saltos de página
- Línea horizontal
- Hipervínculos
- Marcadores
- Tablas
  - *Rowspans, colspans*
  - *Contenido con formato/enriquecido*

 © 2016-2022 Altova GmbH

  - *Tablas anidadas*
  - *Encabezados y pies*
- Listas y sublistas
  - *Con viñetas: estilos diferentes*
  - *Con enumeración: estilos diferentes*
- Imágenes

5.3 Insertar contenido de MS Excel

Si tiene Microsoft Excel 2007+ instalado en el equipo, puede pegar el contenido de documentos Excel en el diseño SPS como **contenido estático**. El contenido estático se insertará en forma de tablas estáticas y otros componentes de diseño adecuados para ello. Las propiedades de formato se conservarán (*imagen siguiente*). Cada hoja de Excel se inserta en el diseño en una tabla estática diferente.

	A	B
1	09.03	Euclid's Elements
2	11.09	English Phrasal Verbs in Use
3	08.86	Code Book
4	09.80	Foundations and Fundamental Concepts of Mathematics
5	18.38	Style
6	08.72	The English Language
7	18.36	History of Mathematics
8	06.09	QED
9	07.89	Fowler's Modern English Usage
10	05.30	Oxford Guide to Plain English
11	21.88	Rediscover Grammar
12	11.97	How to Solve It
13	21.54	Advanced Learner's Grammar
14	19.02	Macmillan English Grammar in Context
15	22.57	Oxford Style Manual
16	16.44	Proofreading
17	20.51	Taschenbuch Mathematischer Formeln und Moderner Verfahren
18	97.50	Oxford Companion to the Book

Hoja Excel

© 2016-2022 Altova GmbH

§p Sheet1 §p

09.03	Euclid's Elements
11.09	English Phrasal Verbs in Use
08.86	Code Book
09.80	Foundations and Fundamental Concepts of Mathematics
18.38	Style
08.72	The English Language
18.36	History of Mathematics
06.09	QED
07.89	Fowler's Modern English Usage
05.30	Oxford Guide to Plain English
21.88	Rediscover Grammar
11.97	How to Solve It
21.54	Advanced Learner's Grammar
19.02	Macmillan English Grammar in Context
22.57	Oxford Style Manual
16.44	Proofreading
20.51	Taschenbuch Mathematischer Formeln und Moderner Verfahren
97.50	Oxford Companion to the Book

§p §p

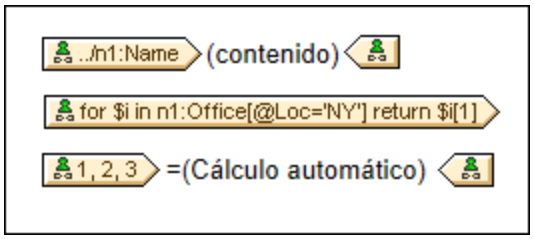
Contenido de Excel importado en un diseño en forma de tabla estática, conservándose el formato del texto.

Nota: además de contenido de Excel, en los diseños de StyleVision también puede pegar *cualquier otro tipo de contenido que se pueda pegar en documentos Excel*. Por ejemplo, contenido de **MS Word** y contenido de **páginas HTML**.

Nota: para crear un archivo SPS que incluya contenido estático de un documento Excel entero, utilice el comando **Archivo | Nuevo | Nuevo a partir de un documento Excel 2007+** 761.

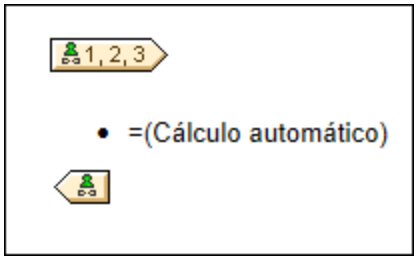
5.4 Plantillas definidas por el usuario

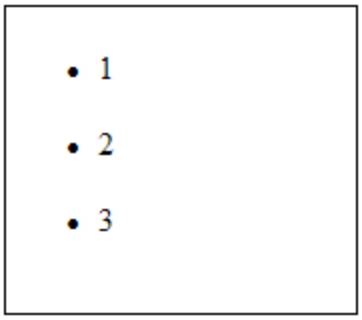
Las plantillas definidas por el usuario son plantillas para elementos generados por una expresión XPath creada por el usuario. Estos elementos pueden ser valores atómicos o nodos. En la imagen siguiente puede ver tres plantillas definidas por el usuario. Observe que las etiquetas de este tipo de plantillas tienen un icono en forma de peón verde. Las plantillas definidas por el usuario son muy prácticas porque ofrecen una gran flexibilidad. Sin embargo, no olvide que el contenido generado por plantillas definidas por el usuario **no se puede editar en la vista Authentic**.



La expresión XPath de las plantillas definidas por el usuario de la imagen hace esto:

- Selecciona un nodo de un esquema fuente. Con una expresión XPath puede alcanzar cualquier nodo de cualquier esquema fuente desde cualquier nodo de contexto. Si StyleVision puede apuntar inequívocamente al nodo especificado, la plantilla definida por el usuario se convierte automáticamente en una plantilla normal, lo cual habilita la edición en la vista Authentic. Si es una plantilla definida por el usuario, el icono de la etiqueta tendrá forma de peón verde.
- Selecciona un nodo que cumple con la condición especificada por la construcción `for` de XPath 2.0 y XPath 3.0. Dichas plantillas no pueden convertirse en plantillas normales porque la construcción `for` no permite a StyleVision resolver el destino inequívocamente con la información de esquema que tiene a su disposición.
- Selecciona una secuencia de valores atómicos `{1, 2, 3}`. Si está permitido crear una plantilla para un valor atómico, no podrá usar el marcador de posición `(contenido)` dentro de dicha plantilla. Esto se debe a que la instrucción `xsl:apply-templates` (que es la que genera el marcador de posición `(contenido)`) solamente se puede aplicar a elementos de nodo (no a valores atómicos). Sin embargo, lo que puede hacer es usar un cálculo automático junto con otro elemento de diseño como una lista. Por ejemplo, la plantilla definida por el usuario que aparece a continuación daría como resultado el texto de la derecha.





Nota: si el archivo SPS utiliza el modo XSLT 1.0, entonces la expresión XPath utilizada debe devolver un conjunto de nodos. De lo contrario se emite un error.

© 2016-2022 Altova GmbH

Ventaja de usar XPath para seleccionar nodos

La ventaja de usar expresiones XPath para seleccionar nodos del esquema (o lo que es lo mismo, plantillas definidas por el usuario) es que la potencia del mecanismo de selección de rutas de XPath se puede aprovechar para seleccionar cualquier nodo o secuencia de elementos, así como para filtrar o establecer condiciones para la selección de nodos. Como resultado, podrá apuntar a ciertos nodos del documento XML para una plantilla determinada. Por ejemplo, la expresión XPath `//Office/Department[@Location="NY"]` seleccionará solamente los nodos `Department` cuyo atributo `Location` tenga el valor `NY`. Para más ejemplos consulte el resto de este apartado.

Nota: si una expresión XPath contiene varios pasos de ruta de acceso, entonces la posición de los paréntesis desempeña un papel importante (sobre todo en el caso de las agrupaciones y ordenaciones). Por ejemplo, la expresión XPath `/Org/Office/Dept` no se procesa igual que la expresión `(/Org/Office/Dept)`. En el caso de la primera expresión, sin paréntesis, el procesador recorre cada paso de la ruta. Para la segunda expresión, con paréntesis, todos los elementos `Dept` de todos los elementos `Office` se devuelven en un solo conjunto de nodos.

Paréntesis	Mecanismo XSLT subyacente	Efecto

No	`<xsl:for-each select="Org">`
	`  <xsl:for-each select="Office">`
	`  <xsl:for-each select="Dept">`
	`  ...`
	`  </xsl:for-each>`
	`  </xsl:for-each>`
	`</xsl:for-each>`
		Cada elemento `Office` tiene sus propios `Dept`. Puede realizar agrupaciones y ordenaciones dentro de cada `Office`.

Sí	`<xsl:for-each`
	`select="/Org/Office/Dept">`
	`  ...`
	`</xsl:for-each>`
		Los elementos `Dept` abarcan todos los elementos `Office` y todos los elementos `Org`.

Tenga en cuesta esta importante diferencia a la hora de realizar agrupaciones y ordenaciones.

Insertar una plantilla definida por el usuario

Para insertar una plantilla definida por el usuario:

1. Haga clic en el icono **Insertar plantilla definida por el usuario** de la barra de herramientas "Insertar elementos de diseño" y después en la posición del diseño donde desea insertar la plantilla. También puede hacer clic con el botón derecho en el diseño y seleccionar el comando **Insertar plantilla definida por el usuario** del menú contextual.
2. Se abre el cuadro de diálogo Editar expresión XPath 729. Escriba la expresión XPath que desea usar y haga clic en **Aceptar**. Observe que el nodo de contexto de la expresión XPath será el nodo donde hizo clic. Se crea una plantilla nodo vacía. A veces se crea un nodo unido. Cuando un nodo está unido, los nodos de instancia a los que se apunta están seleccionados como si estuvieran en el mismo nivel. Cuando el nodo no está unido (sino dividido en varios niveles jerárquicos) la selección del nodo se hace recorriendo cada nodo de instancia de cada nivel. El conjunto de nodos que la expresión devuelve en ambos casos (unido y dividido) es el mismo, a no ser que se especifique un criterio de agrupación o de ordenación. Para más información consulte el apartado Operaciones con plantillas nodo 274.

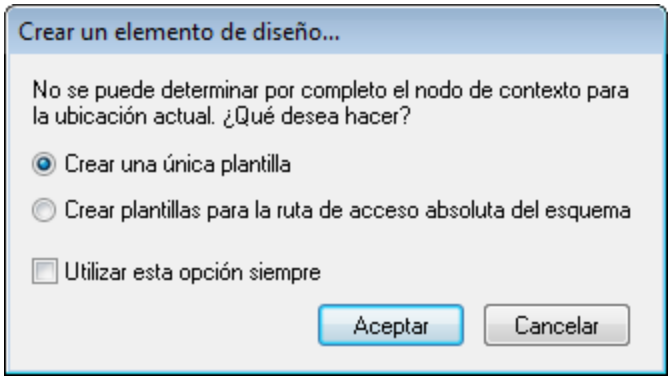
Editar el modelo de coincidencia para la plantilla

La selección de nodos de cualquier plantilla nodo (normal o definida por el usuario) se puede cambiar usando

una expresión XPath y seleccionando otro modelo de coincidencia. Para editar el modelo de coincidencia de una plantilla nodo, haga clic con el botón derecho en la plantilla nodo y seleccione **Editar modelo de coincidencia** en el menú contextual. Se abre el cuadro de diálogo "Editar expresión XPath", donde puede introducir la expresión XPath para seleccionar otro nodo. Después haga clic en **Aceptar**.

Agregar nodos a plantillas definidas por el usuario

Si añade un nodo de esquema a una plantilla definida por el usuario creada para un nodo o secuencia que no se puede colocar en el contexto del esquema fuente, StyleVision no puede conocer el contexto para el nodo nuevo. En este caso aparece un aviso (*imagen siguiente*) preguntando cómo se debe hacer referencia al nuevo nodo: (i) por su nombre (con una ruta de acceso relativa) o (ii) por una ruta de acceso completa empezando por la raíz del esquema fuente.



Este aviso aparece siempre por defecto, pero se puede deshabilitar en la pestaña *Diseño* del cuadro de diálogo "Opciones" (**Herramientas | Opciones** 894).

Temas relacionados

- Archivos SPS: contenido 134
- Operaciones con plantillas nodo 274
- Elementos definidos por el usuario 151

© 2016-2022 Altova GmbH

5.5 Elementos y bloques de texto definidos por el usuario

Los elementos definidos por el usuario 151 y los bloques de texto XML definidos por el usuario 152 sirven para insertar (i) cualquier tipo de elemento y (ii) cualquier tipo de bloque de texto XML en el diseño. Su ventaja es que el diseñador del SPS no tiene por qué limitarse a utilizar elementos XML de los esquemas fuente y elementos de diseño de StyleVision. También pueden crear (i) plantillas para cualquier otro elemento (definidos por el usuario) y (ii) código XML independiente (bloques XML) que crea objetos por separado (como objetos ActiveX, por ejemplo).

Hay una diferencia importante entre los elementos definidos por el usuario y los bloques de texto XML definidos por el usuario. Los elementos definidos por el usuario se crean en el diseño como nodos plantilla para un solo elemento XML (con atributos). Todo el contenido de esta plantilla debe crearse explícitamente. Este contenido puede ser cualquier elemento de diseño de la SPS. Por el contrario, los bloques de texto XML definidos por el usuario no pueden contener elementos de diseño. Se trata de bloques independientes y autónomos. Puesto que los elementos definidos por el usuario están vacíos cuando se crean, no sirven para crear un objeto que necesite líneas de código. Si necesita crear un objeto con código utilice los bloques de texto XML definidos por el usuario.

Nota: la vista Authentic solamente es compatible con los elementos y bloques de texto XML definidos por el usuario creados en los productos de Altova de la edición Enterprise Edition.

5.5.1 Elementos definidos por el usuario

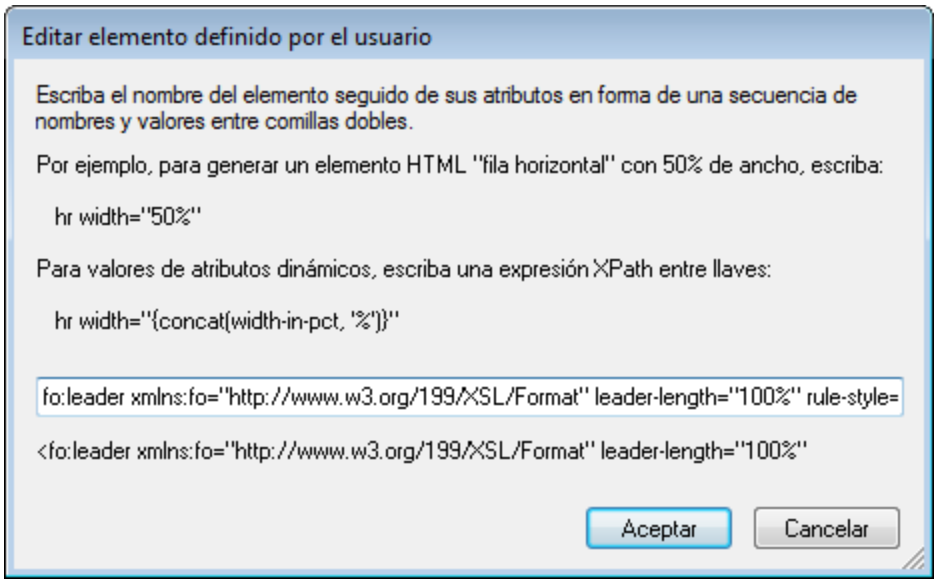
Los elementos definidos por el usuario son elementos que se pueden generar en el resultado del SPS sin necesidad de que dichos elementos existan en los esquemas fuente del SPS. Esto significa que puede insertar cualquier elemento de cualquier espacio de nombres (HTML o XSL-FO) en cualquier posición del diseño. Después puede insertar elementos de diseño SPS dentro de estos elementos.

Nota: la vista Authentic solamente es compatible con los elementos definidos por el usuario creados en los productos de Altova de la edición Enterprise Edition.

Insertar elementos definidos por el usuario

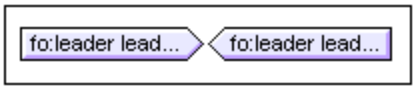
Si quiere usar elementos definidos por el usuario en su diseño, siga estas instrucciones:

1. Haga clic con el botón derecho en la posición del diseño donde desea insertar el elemento definido por el usuario.
2. En el menú contextual seleccione **Insertar elemento definido por el usuario | Elemento definido por el usuario**.
3. Se abre un cuadro de diálogo (*imagen siguiente*). Escriba el nombre del elemento, sus atributos con sus valores y una declaración de espacio de nombres si el documento no incluye ninguna.



En el ejemplo de la imagen anterior, se creó un elemento XSL-FO llamado `leader`. Se le dio el prefijo `fo:`, que está enlazado con la declaración de espacio de nombres `xmlns:fo="http://www.w3.org/1999/XSL/Format"`. El elemento tiene varios atributos, incluidos los atributos `leader-length` y `rule-style`, cada uno con su valor. El elemento, sus atributos y su declaración de espacio de nombres se deben introducir sin los corchetes angulares.

4. Haga clic en **Aceptar** para insertar el elemento en el diseño. El elemento aparece en el diseño en forma de plantilla vacía, con etiquetas de apertura y cierre (*imagen siguiente*).



5. Ahora puede añadir contenido a esta plantilla tal y como haría con cualquier otra. El elemento definido por el usuario puede incluir contenido estático, contenido dinámico del documento XML y otros elementos definidos por el usuario.

Nota: si su elemento definido por el usuario está pensado para un formato de salida particular, inclúyalo dentro de una condición basada en el resultado. Así evitará resultados inesperados en los demás formatos de salida.

Temas relacionados

- Bloques de texto XML definidos por el usuario 152
- Plantillas definidas por el usuario 148

5.5.2 Bloques definidos por el usuario

Un bloque de texto XML definido por el usuario es un fragmento XML que se insertará en el código XSLT generado por el diseño SPS. Se inserta en el diseño SPS en forma de bloque independiente en el cual no se

 © 2016-2022 Altova GmbH

puede añadir ningún elemento de diseño. Este bloque de texto XML se aplicará como código XSLT en la posición de la hoja de estilos donde aparece.

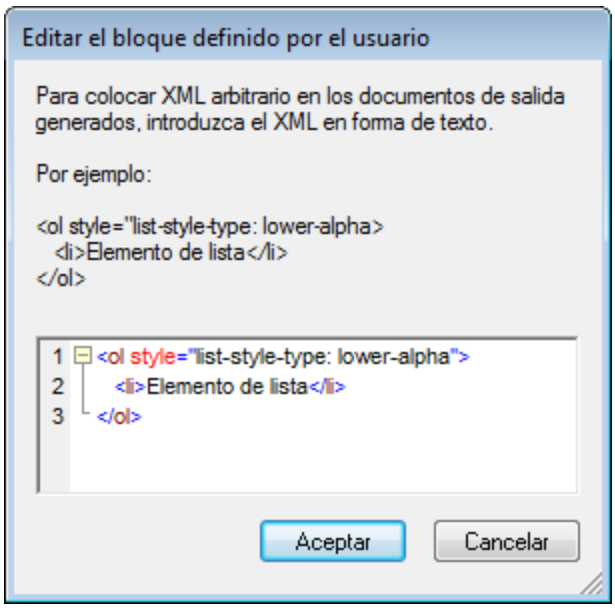
Estos bloques son muy útiles porque permiten al diseñador de la hoja de estilos insertar fragmentos XSLT y código personalizado en el diseño. Por ejemplo, puede insertar un objeto ActiveX dentro de un elemento HTML `SCRIPT`.

Nota: esta característica solamente es compatible con las ediciones Enterprise Edition que incluyen la vista Authentic (es decir, con las ediciones Enterprise Edition de StyleVision, Authentic Desktop, Authentic Browser y XMLSpy).

Insertar bloques de texto XML definidos por el usuario

Siga estas instrucciones para insertar un bloque de texto XML:

1. Haga clic con el botón derecho en la posición del diseño donde desea insertar el bloque definido por el usuario.
2. En el menú contextual seleccione **Insertar elemento definido por el usuario | Bloque definido por el usuario**.
3. Aparece un cuadro de diálogo (*imagen siguiente*), escriba el bloque de texto XML que desea insertar. No olvide que debe tener un formato XML correcto. Si el formato no es correcto, el botón **Aceptar** del diálogo no se habilita.



En el ejemplo de la imagen anterior se añade un bloque de texto XML que genera una lista HTML ordenada.

4. Haga clic en **Aceptar** para insertar el elemento en el diseño. El bloque de texto aparece en el diseño en forma de cuadro de texto.

Nota: si su bloque de texto XML está pensado para un formato de salida particular, inclúyalo dentro de una condición basada en el resultado. Así evitará resultados inesperados en los demás formatos de salida.

Temas relacionados

- Elementos definidos por el usuario 151
- Plantillas definidas por el usuario 148

© 2016-2022 Altova GmbH

5.6 Tablas

En un diseño SPS se pueden utilizar dos tipos de tabla: tablas SPS y tablas CALS/HTML. Estos dos tipos de tabla son muy diferentes y es importante comprender estas diferencias. Esta sección describe en detalle cómo se debe usar cada tipo.

Tablas SPS

Una tabla SPS es un componente de un diseño SPS al que se le da estructura y formato en el diseño. Se puede crear en cualquier parte del diseño y puede crear tantas tablas SPS como quiera.

Las tablas SPS son dispositivos de presentación y se representan usando el vocabulario de presentación de la vista Authentic y del formato de salida. La estructura de una tabla SPS **no está representada por nodos del documento XML**, aunque el contenido propiamente dicho de las celdas de la tabla proceda de nodos del documento XML.

Hay dos tipos de tablas SPS:

- **Tablas estáticas:** las crea paso a paso la persona que diseña el SPS. Después de crear la estructura de la tabla se define el contenido de cada celda por separado. El contenido de las celdas puede proceder de diferentes posiciones de la estructura del esquema e incluso de tipos diferentes. No olvide que las filas de una tabla estática no representan una estructura de datos repetitiva. Por eso este tipo de tabla se denomina *estática*: tiene una estructura fija que no cambia, independientemente del contenido XML.
- **Tablas dinámicas:** están pensadas para estructuras de datos del documento XML que se repiten. Pueden crearse para elementos de esquema que tienen una subestructura (es decir, que tienen como mínimo un atributo o elemento secundario). Cualquier elemento de una subestructura se repite si existe más de una instancia suya. Cada instancia del elemento equivale a una fila de la tabla dinámica y sus elementos o atributos secundarios (todos o algunos) equivalen a las columnas de la tabla. La estructura de una tabla dinámica, por tanto, refleja el contenido del archivo XML y cambia con el contenido de forma dinámica.

Tablas CALS/HTML

El modelo de contenido de una tabla CALS o HTML se define en el documento XML (y por extensión en el DTD o en el esquema) y sigue la correspondiente especificación (CALS o HTML). Después puede especificar en el diseño SPS si la tabla CALS/HTML se procesa o no como tabla. La estructura de datos XML que representa la tabla CALS/HTML en este caso genera marcado de tabla para cada formato de salida. El formato de las tablas CALS/HTML se puede especificar en el documento XML de instancia o en el SPS o en ambos.

Por ejemplo, observe la vista previa HTML de esta tabla HTML.

Name	Phone
John Merrimack	6517890
Joe Concord	6402387

Este sería el fragmento de código HTML para la tabla XML anterior:

```
<table border="1" width="40%">
  <tbody>
```

```
    <tr>
      <td>Name</td>
      <td>Phone</td>
    </tr>
    <tr>
      <td>John Merrimack</td>
      <td>6517890</td>
    </tr>
    <tr>
      <td>Joe Concord</td>
      <td>6402387</td>
    </tr>
  </tbody>
</table>
```

El documento XML original podría ser este:

```
<phonelist border="1" width="40%">
  <items>
    <person>
      <data>Name</data>
      <data>Phone</data>
    </person>
    <person>
      <data>John Merrimack</data>
      <data>6517890</data>
    </person>
    <person>
      <data>Joe Concord</data>
      <data>6402387</data>
    </person>
  </items>
</phonelist>
```

Observe que los nombres de elemento del documento XML no necesitan tener semántica de tabla. Sin embargo, la estructura de la tabla, debe corresponder al modelo de tabla HTML o CALS. Además, tenga en cuenta que:

- Al elemento de columna HTML `<td/>` solamente puede corresponder un elemento XML.
- Una tabla CALS/HTML se puede insertar en cualquier posición del documento XML donde, según el esquema, se permita el elemento que corresponde al elemento `table`.
- En la vista Authentic los datos se introducen directamente en las celdas de la tabla. Estos datos se almacenan como contenido del elemento de tabla CALS/HTML correspondiente.
- Las propiedades de formato de una tabla CALS/HTML pueden venir del documento XML o se pueden especificar en el diseño SPS.

Resumen para diseñadores

Desde el punto de vista del diseñador del SPS, podemos resumir todo esto en tres puntos:

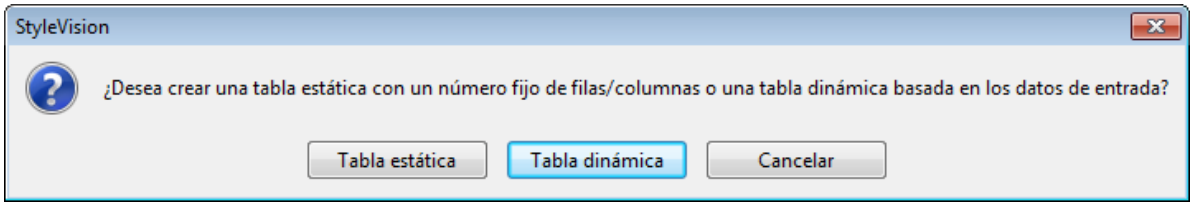
- La **estructura** de una tabla SPS se define en el SPS. Por el contrario, la estructura de una tabla CALS/HTML se especifica en el esquema y debe seguir la del modelo de la tabla CALS/HTML. Sin embargo, los nombres de elemento del esquema pueden ser diferentes de los nombres de elemento de los modelos de tabla CALS o HTML.
- Los elementos **colspan** y **rowspan** de las tablas SPS se especifican en el SPS. Pero en las tablas CALS/HTML, los colspan y rowspan se especifican en el documento XML de instancia.
- El **formato de tabla** de las tablas SPS se especifica en el SPS. El formato de las tablas CALS/HTML se especifica en el documento XML de instancia y/o en el SPS.

© 2016-2022 Altova GmbH

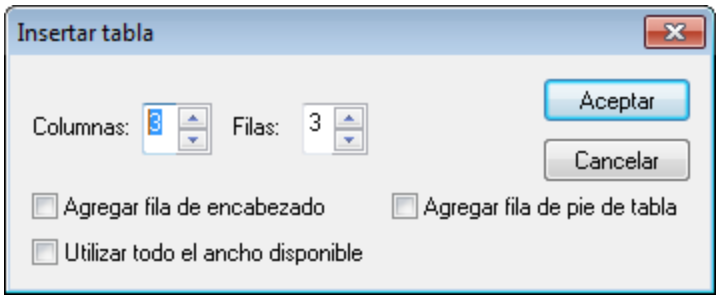
5.6.1 Tablas estáticas

Para crear una tabla estática:

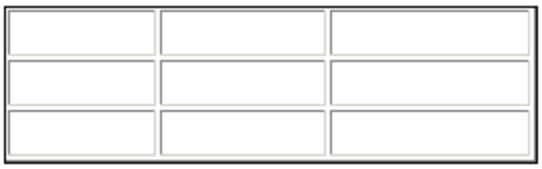
1. Use uno de estos mecanismos: (i) clic en el comando **Tabla | Insertar tabla**, (ii) clic en el comando **Insertar | Tabla** o (iii) clic en el icono **Insertar tabla** de la barra de herramientas "Insertar elementos de diseño".
2. Todos estos comandos abren este cuadro de diálogo:



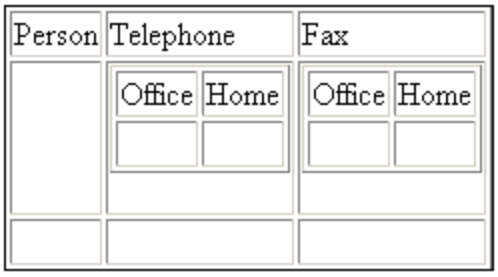
   Haga clic en el botón **Tabla estática**.
3. Se abre el cuadro de diálogo "Insertar tabla" (*imagen siguiente*), donde puede especificar las dimensiones de la tabla y si la tabla debe ocupar todo el ancho disponible.



4. Haga clic en **Aceptar** y se crea una tabla vacía con las dimensiones elegidas (*imagen siguiente*).



5. Ahora puede introducir el contenido en las celdas de la tabla usando las funciones estándar de StyleVision. El contenido de las celdas puede ser texto, elementos arrastrados desde la estructura del esquema u objetos como imágenes y tablas anidadas. Por ejemplo, esta tabla contiene tablas anidadas.



Las tablas estáticas SPS son ideales para organizar datos XML situados al azar en la jerarquía del esquema o para almacenar contenido estático (contenido que no procede de la fuente XML).

Eliminar columnas, filas y tablas

Para eliminar una columna, fila o tabla, ponga el cursor en la columna, fila o tabla que desea eliminar y haga clic en el comando de menú **Tabla | Eliminar columna**, **Tabla | Eliminar fila** o **Tabla | Eliminar tabla**, respectivamente. Si tiene tablas anidadas, estos comandos se aplicarán a la columna, fila y tabla donde se encuentre el cursor.

Iconos de edición de tablas de la barra de herramientas

Los iconos de edición de tablas, que por defecto están en la segunda fila de la barra de herramientas, son accesos directos a los comandos del menú **Tabla**. Estos comandos sirven para insertar, eliminar, editar la estructura y asignar propiedades de formato a la tabla estática. Estos iconos también se pueden usar para editar tablas SPS dinámicas. Pero **no se pueden usar para editar tablas CALS/HTML**, porque el formato de las tablas CALS/HTML 172 se define de otra manera.

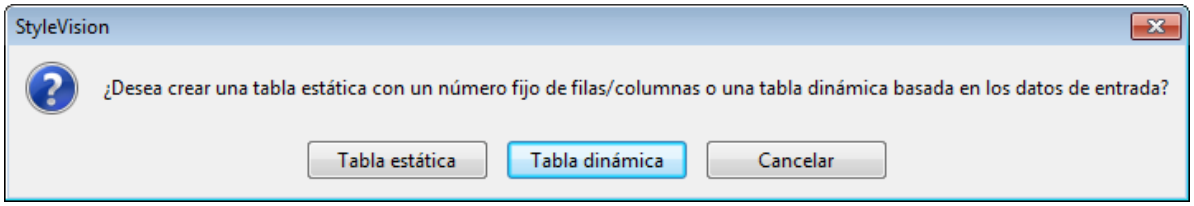
Temas relacionados

- Crear tablas dinámicas 158
- Procesamiento condicional en tablas 162
- Tablas SPS en la vista Diseño 164
- Formato de tablas estáticas y dinámicas 165
- Tablas CALS/HTML 172

5.6.2 Tablas dinámicas

Para insertar una tabla dinámica:

1. Use uno de estos mecanismos: (i) clic en el comando **Tabla | Insertar tabla**, (ii) clic en el comando **Insertar | Tabla** o (iii) clic en el icono **Insertar tabla** de la barra de herramientas "Insertar elementos de diseño".
2. Todos estos comandos abren el cuadro de diálogo que aparece en la imagen siguiente. Si seleccionó el icono **Insertar tabla** de la barra de herramientas, el cuadro de diálogo se abre al hacer clic en la parte del diseño donde desea insertar la tabla.

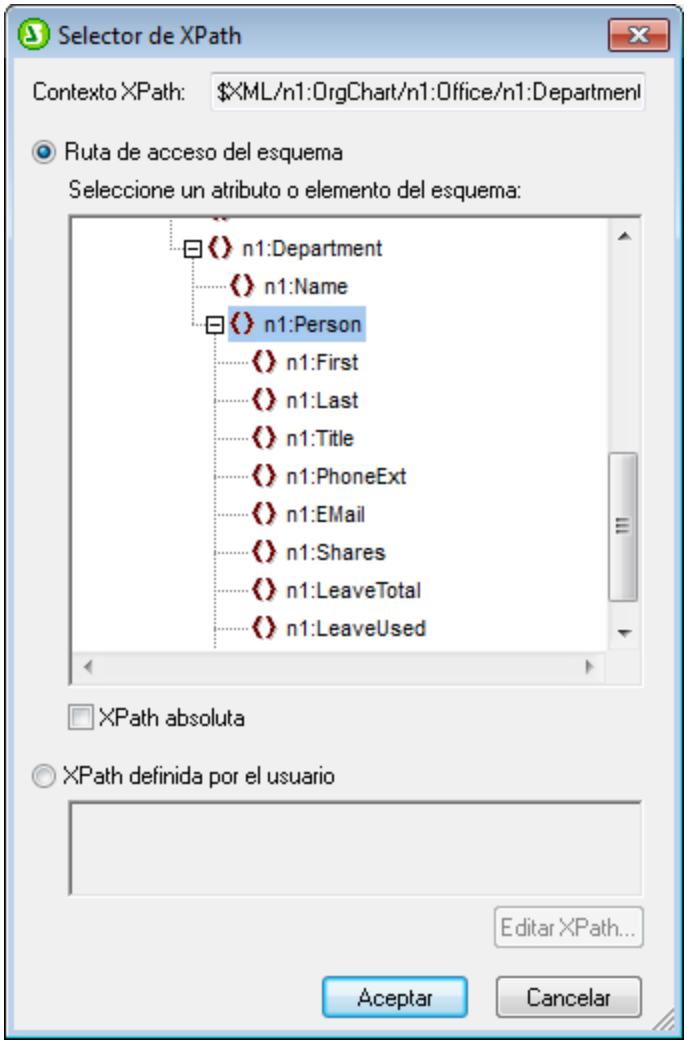


   Haga clic en el botón **Tabla dinámica**.

3. Se abre el cuadro de diálogo "Selector de XPath" (*imagen siguiente*). Observe que en el cuadro *Contexto XPath* aparece el contexto de la posición donde eligió insertar la tabla y no se puede

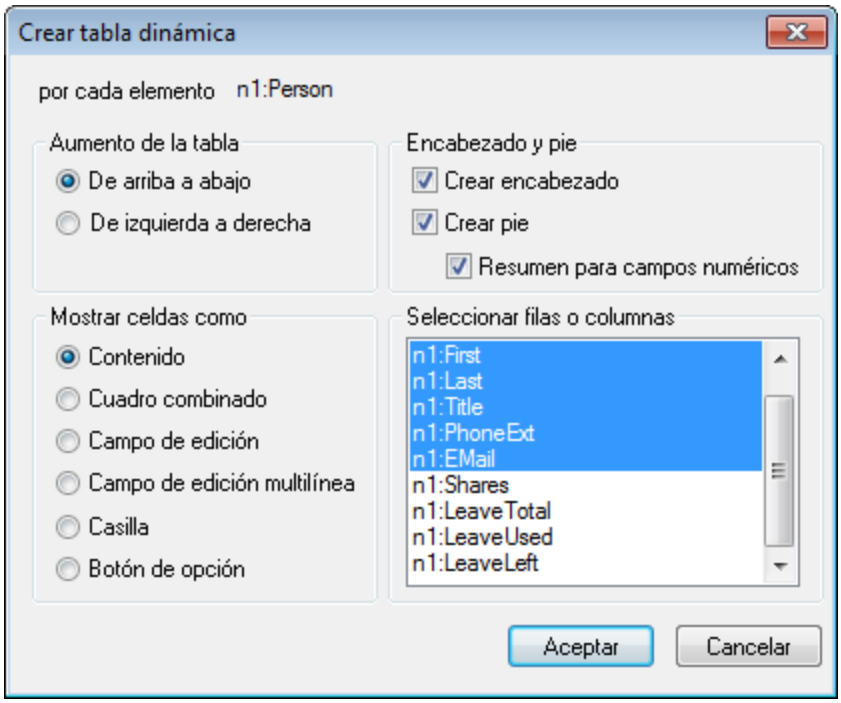
 © 2016-2022 Altova GmbH

cambiar en el cuadro de diálogo. Seleccione el nodo con el que desea crear la tabla dinámica. En la imagen siguiente el nodo de contexto es `n1:Department` y el nodo `n1:Person` se usará para crear la tabla.



Seleccione la opción *XPath definida por el usuario*, si prefiere escribir una expresión XPath para seleccionar el nodo con el que se creará la tabla dinámica.

4. Haga clic en **Aceptar**. Se abre el cuadro de diálogo "Crear tabla dinámica" (*imagen siguiente*).



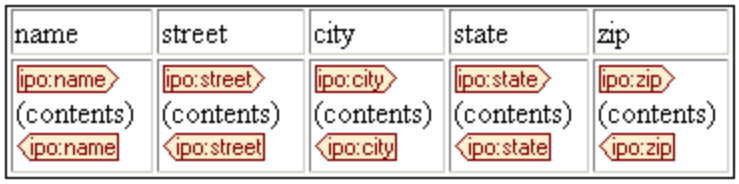
5. Los elementos y atributos secundarios del elemento que arrastró hasta la vista **Diseño** aparecen en el grupo de lista *Seleccionar filas o columnas* y se pueden usar para crear las columnas de la tabla. Desactive los nodos secundarios que no desee utilizar y seleccione los que desea usar para crear columnas. En la imagen anterior, se desactivaron los elementos `Shares`, `LeaveTotal`, `LeaveUsed` y `LeaveLeft`.) El resto de opciones del cuadro de diálogo se explican más abajo. Haga clic en **Aceptar** cuando termine y recuerde que solamente se crean tablas para los elementos y atributos secundarios, no para los descendientes situados en niveles inferiores.

Nota: si usó una *XPath definida por el usuario* para seleccionar el nodo con el que se crea la tabla dinámica, lo más probable es que StyleVision no pueda distinguir a qué nodo está apuntando. Como consecuencia, puede que a veces el cuadro de diálogo "Crear tabla dinámica" no incluya una lista de atributos/elementos secundarios en el grupo *Seleccionar filas o columnas.* En estos casos deberá rellenar a mano la tabla con el contenido de nodo. Este contenido de nodo debe ser elementos/atributos secundarios del nodo que seleccionó y a partir del cual se creó la tabla.

Nota: hay otra manera de crear una tabla a partir de un nodo de esquema: arrastrando el nodo desde la estructura del esquema hasta el diseño y especificando, al soltar el nodo, que se debe crear una tabla a partir de él.

Aumento de la tabla

Este es el aspecto que tiene una tabla cuando aumenta de tamaño *de arriba a abajo*:



 © 2016-2022 Altova GmbH

Y este es el aspecto que tiene cuando aumenta de tamaño de *izquierda a derecha*:

Encabezados y pies de tabla

Las columnas y filas pueden tener encabezados, que equivalen a los nombres de los elementos de columna y fila. Los encabezados de columna se crean encima de cada columna. Los encabezados de fila se crean a la izquierda de una fila. Para incluir encabezados marque la casilla *Crear encabezado.* Si la tabla aumenta de arriba a abajo y crea un encabezado, se crea una fila de encabezado encima del cuerpo de la tabla. Si la tabla aumenta de izquierda a derecha y crea un encabezado, la columna de encabezado se crea a la izquierda del cuerpo de la tabla.

Para incluir pies de tabla marque la casilla *Crear pie*. Los pies de tabla, como los encabezados, se pueden crear tanto para las columnas (debajo de las columnas) como para las filas (a la derecha de una fila). El pie de tabla de columnas o filas numéricas suma todas las columnas o filas si se marcó la casilla *Resumen para campos numéricos*.

En el menú **Tabla** encontrará comandos para unir y dividir las celdas del encabezado y del pie de tabla, así como para insertar, anexar y eliminar filas y columnas. Esto ofrece una gran flexibilidad a la hora de dar estructura a los encabezados y pies de tabla. Además, los encabezados y pies pueden tener cualquier tipo de contenido estático o dinámico, incluso plantillas condicionales y cálculos automáticos.

Nota: los encabezados y pies se deben crear cuando se define la tabla dinámica (marcando las casillas *Crear encabezado* y *Crear pie* del cuadro de diálogo "Crear tabla dinámica". Cuando se anexa o inserta una fila dentro de una tabla dinámica no se crean encabezados ni pies, sino una fila nueva, que son cosas muy diferentes. Con las opciones *Crear encabezado/Crear pie* se añaden encabezados y pies de tabla al principio y al final de la tabla, respectivamente. Si se inserta o anexa una fila, la fila aparece cada vez que aparece el elemento con el que se creó la tabla dinámica.

Tablas dinámicas anidadas

Puede anidar una tabla dinámica dentro de otra si el elemento con el que se crea la tabla anidada es secundario del elemento que se usó para crear la tabla dinámica en la que se anida.

1. Cree la tabla dinámica exterior y cree una columna con el elemento secundario que quiere usar como tabla anidada.
2. Ahora haga clic con el botón derecho en el elemento secundario en la tabla dinámica (en la vista **Diseño**).
3. Seleccione el comando **Convertir en | Tabla**. Aparece el cuadro de diálogo "Crear tabla dinámica".
4. Defina las propiedades de la tabla dinámica anidada.

Para anidar una tabla dinámica en una tabla estática, arrastre el elemento con el que desea crear la tabla dinámica y suéltelo en una celda de la tabla estática. Al soltarlo aparece un menú contextual, donde debe seleccionar el comando **Crear tabla**.

Tablas para elementos con contenido de texto

Para crear columnas (o filas) para elementos secundarios, el elemento a partir del cual se crea la tabla debe tener un nodo de atributo o elemento secundario. Los nodos de texto secundarios no bastan. Si este es su caso, puede crear un elemento secundario llamado `Texto`, por ejemplo, y poner su nodo de texto dentro de los elementos `ElementoDeTabla/Texto`. Ahora podrá crear una tabla dinámica a partir de `ElementoDeTabla`. Esta tabla tendrá una columna para los elementos `Texto`. Cada fila tendrá una celda con el nodo de texto del elemento `Texto` y las filas de la tabla equivaldrán a las instancias del elemento `ElementoDeTabla`.

Contenido de las celdas de la tabla

Al crear una tabla dinámica puede elegir qué tipo de componente de StyleVision se usa para representar el contenido de las celdas. En el ejemplo de la imagen anterior, las celdas de la tabla se crearon en forma de contenido. Es decir, se marcó la opción *Mostrar celdas como contenido* en el cuadro de diálogo "Crear tabla dinámica". Pero también se podrían crear en forma de dispositivo de entrada de datos. Dicho lo cual, es necesario señalar dos cosas:

- La opción seleccionada es una opción global que afecta a todas las celdas de la tabla. Si quiere que alguna celda tenga un aspecto diferente a las demás, deberá editar dicha celda después de crear la tabla: haga clic con el botón derecho en la celda y seleccione **Convertir en** en el menú contextual. Después seleccione el nuevo tipo de contenido.
- Si crea celdas como contenido y el elemento tiene elementos secundarios, el contenido de la celda será una concatenación de cadenas de texto del elemento y de sus descendientes.

Eliminar columnas, filas y tablas

Para eliminar una columna, fila o tabla, ponga el cursor en la columna, fila o tabla que desea eliminar y haga clic en el menú **Tabla | Eliminar columna**, **Tabla | Eliminar fila** o **Tabla | Eliminar tabla**, respectivamente. Si tiene tablas anidadas, estos comandos se aplicarán a la columna, fila y tabla donde se encuentre el cursor.

Iconos de edición de tablas de la barra de herramientas

Los iconos de edición de tablas de la barra de herramientas son accesos directos a los comandos del menú **Tabla**. Estos comandos sirven para insertar, eliminar, editar la estructura y asignar propiedades de formato de la tabla dinámica. Estos iconos también se pueden usar para editar tablas estáticas. Pero **no se pueden usar para editar tablas CALS/HTML**, porque el formato de las tablas CALS/HTML se define de otra manera.

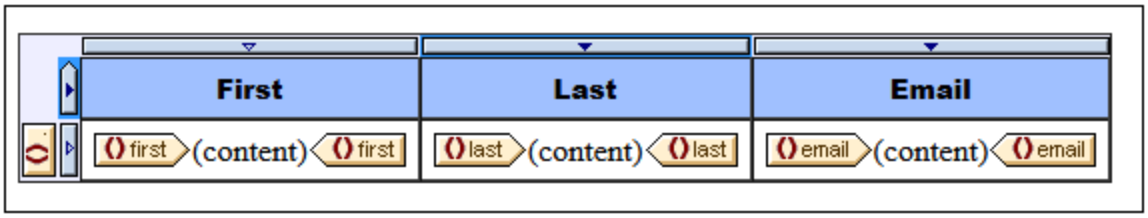
5.6.3 Procesamiento condicional en tablas

Si quiere, puede definir un procesamiento condicional en ciertas columnas y filas de las tablas estáticas y dinámicas, así como en los encabezados de columna y fila. Este procesamiento condicional sirve para mostrar u ocultar la columna, la fila o el encabezado dependiendo de si la condición definida se cumple o no. Si el resultado de la condición es `true`, la columna, la fila o el encabezado se representa en el formato de salida. De lo contrario, se oculta.

Agregar y editar procesamiento condicional

Para añadir procesamiento condicional a columnas, filas o encabezados, haga clic con el botón derecho en el componente elegido y seleccione **Editar instrucción de procesamiento condicional**.

 © 2016-2022 Altova GmbH



Al hacer clic en el comando **Editar instrucción de procesamiento condicional** se abre el cuadro de diálogo Editar expresión XPath 729, donde puede introducir la expresión XPath de la condición. A continuación describimos algunos ejemplos de uso del procesamiento condicional.

- En una columna, fila o tabla, introduzca la expresión XPath `false()` para ocultarla y `true()` para mostrarla.
- Una columna se representa solamente si la suma de todos los valores de la columna supera un valor entero concreto.
- Una columna o fila se representa solamente si dicha columna o fila no tiene ninguna celda vacía.
- Una columna o fila se representa solamente si existe cierto valor de celda en dicha columna o fila.

Para editar una condición ya existente, haga clic con el botón derecho en el componente de diseño que incluye la condición y seleccione **Editar instrucción de procesamiento condicional**. Se abre el cuadro de diálogo Editar expresión XPath 729, donde puede editar la expresión XPath que prueba si la condición se cumple o no.

Quitar el procesamiento condicional

Para eliminar la instrucción de procesamiento condicional de columnas, filas o encabezados, haga clic con el botón derecho donde corresponda y seleccione **Borrar instrucción de procesamiento condicional**.

Temas relacionados

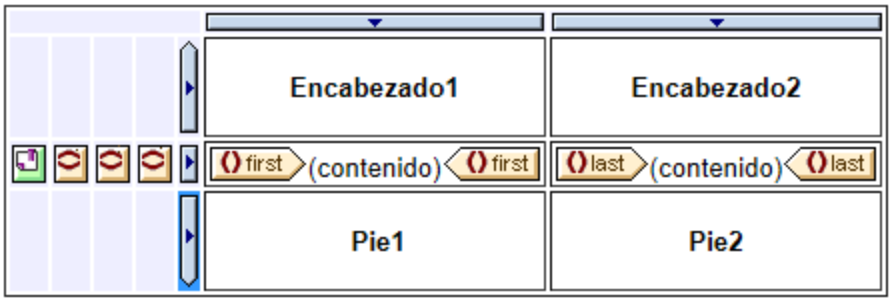
- Crear tablas estáticas 157
- Crear tablas dinámicas 158
- Tablas SPS en la vista Diseño 164
- Formato de tablas estáticas y dinámicas 165
- Tablas CALS/HTML 172

Temas relacionados:

Crear tablas estáticas 157: describe el uso de tablas estáticas.
Crear tablas dinámicas 158: describe el uso de tablas dinámicas.
Tablas SPS en la vista Diseño 164: información sobre cómo se representan las tablas SPS en la vista **Diseño**.
Formato de tablas estáticas y dinámicas 165: información sobre el formato de las tablas.
Tablas CALS/HTML 172: explica cómo trabajar con estructuras de tabla definidas en el documento XML.

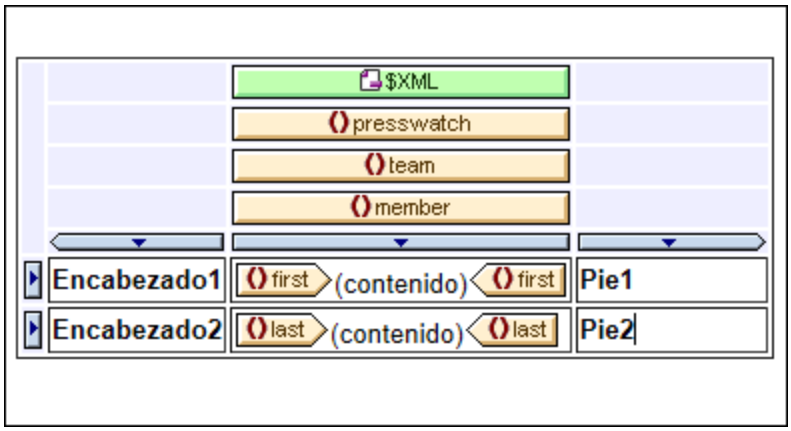
5.6.4 Tablas en la vista Diseño

Las dos imágenes que aparecen a continuación muestran los componentes principales de las tablas SPS estáticas y dinámicas y el marcado de tabla, que está activado (**Tabla | Ver marcado de la tabla**).



La imagen anterior muestra una sencilla tabla cuyo tamaño aumenta de arriba a abajo y que incluye un encabezado y un pie de tabla.

- El indicador de columna es un rectángulo con una punta de flecha apuntando hacia abajo. Los indicadores de columna aparecen sobre las columnas. Para seleccionar una columna entera (si quiere asignar una propiedad de formato a toda la columna, por ejemplo), haga clic en el indicador de la columna.
- El indicador de fila es un rectángulo con una punta de flecha apuntando hacia la derecha. Para seleccionar toda la fila, haga clic en el indicador.
- En las tablas cuyo tamaño aumenta de arriba a abajo (*imagen anterior*) los indicadores de encabezados y pies de tabla son iconos que apuntan hacia arriba y hacia abajo respectivamente. En las tablas cuyo tamaño aumenta de izquierda a derecha, los indicadores de encabezados y pies de tabla apuntan hacia la izquierda y hacia la derecha respectivamente (*imagen siguiente*)
- Para seleccionar toda la tabla, haga clic en la esquina superior izquierda de la tabla.
- Cuando está seleccionada una fila o columna, el fondo de la fila o columna se resalta en color azul oscuro. Por ejemplo, en la imagen anterior está seleccionado el pie de la tabla.
- En las tablas cuyo tamaño aumenta de arriba a abajo, el elemento que se usó para crear la tabla aparece en el extremo izquierdo, fuera de la cuadrícula formada por columnas y filas (*imagen anterior*). En las tablas que aumentan de izquierda a derecha, el elemento que se usó para crear la tabla aparece en el extremo superior, fuera de la cuadrícula formada por columnas y filas (*imagen siguiente*).



 © 2016-2022 Altova GmbH

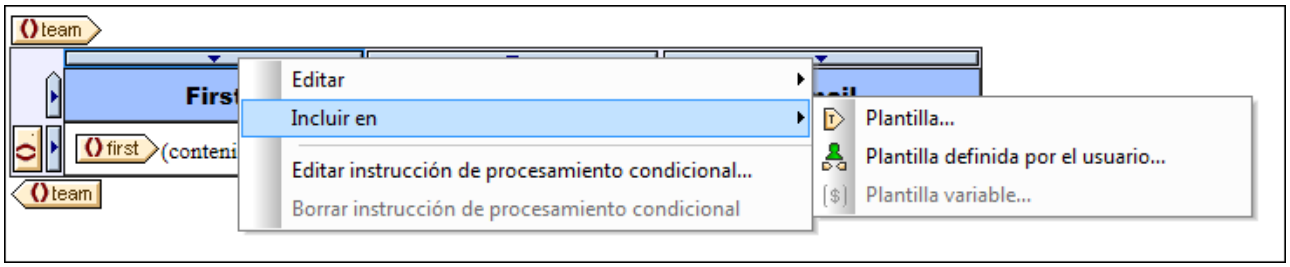
Tras seleccionar una tabla, fila o columna, puede definir sus propiedades y estilos en las ventanas Propiedades y Estilos respectivamente.

Operaciones arrastrar y colocar

Puede arrastrar las columnas y filas de una tabla SPS (estática o dinámica) a una posición nueva dentro de la misma tabla.

Incluir filas y columnas en plantillas o quitar las plantillas

Una fila o columna se puede incluir dentro de una plantilla. Haga clic con el botón derecho en el indicador de fila o columna y en el menú contextual (*imagen siguiente*) seleccione **Incluir en | Plantilla** o **Incluir en | Plantilla definida por el usuario**. En el siguiente paso puede seleccionar un nodo de la estructura del esquema o introducir una expresión XPath para una plantilla definida por el usuario 270. La plantilla se creará alrededor de la fila o columna.



Para quitar la plantilla que rodea la fila o la columna (sin provocar cambios en la fila o columna propiamente dichas) seleccione la etiqueta de la plantilla y pulse la tecla **Suprimir**.

La función para incluir o quitar plantillas es muy útil si quiere eliminar una plantilla sin quitar el contenido de la fila o de la columna y después incluir la fila o columna dentro de otra plantilla. También puede incluir la fila o columna dentro de una plantilla definida por el usuario 270 para usar resultados interesantes obtenidos mediante el uso de filtros (como cálculos automáticos, por ejemplo).

Temas relacionados

- Menú Tabla 844
- Crear tablas estáticas 157
- Crear tablas dinámicas 158
- Formato de tablas estáticas y dinámicas 165
- Tablas CALS/HTML 172
- Operaciones para plantillas nodo 274
- Plantillas definidas por el usuario 270

5.6.5 Formato de tablas

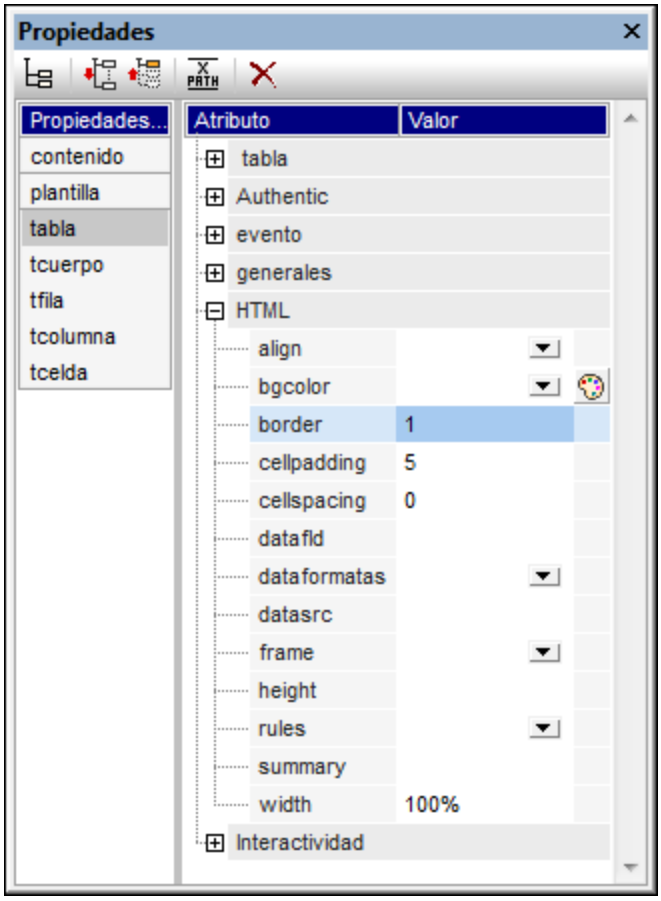
Puede dar formato a las tablas estáticas y dinámicas usando:

- propiedades de formato de tabla HTML (en la ventana Propiedades)
- propiedades de estilo CSS (en la ventana Estilos).

Nota: para los documentos de salida en formato texto, las tablas se generan con carácter definido por el usuario como separador de celdas 787. El carácter predeterminado es el punto y coma. Puede cambiar este carácter en la pestaña XSD/XSLT del cuadro de diálogo Propiedades 787.

Ventana Propiedades

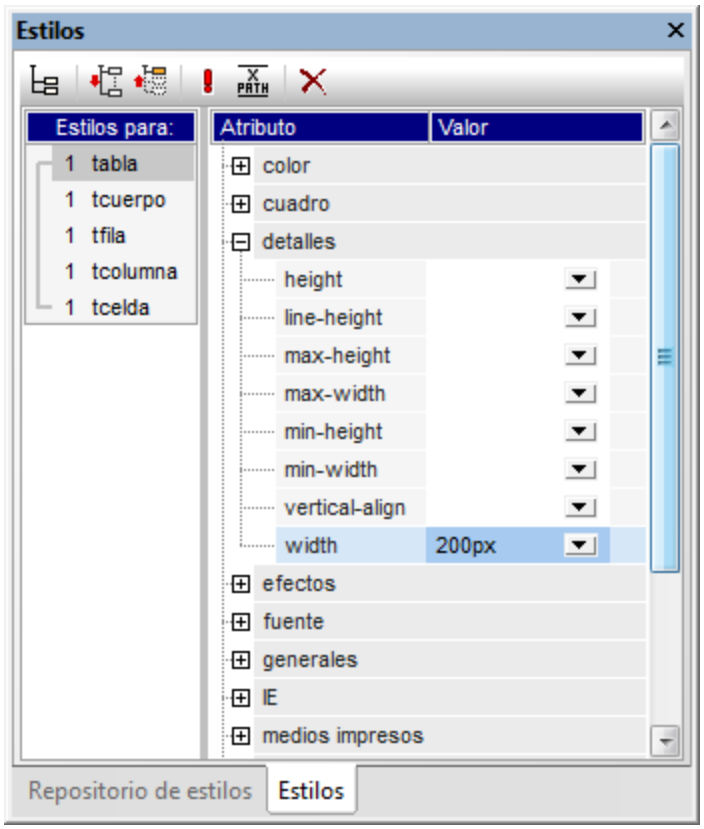
Las propiedades de formato de tabla HTML están disponibles en la ventana Propiedades (*imagen siguiente*). Estas propiedades están disponibles en el grupo `HTML` de propiedades de los componentes y subcomponentes de tabla (**`cuerpo`**, **`fila`**, **`columna`** y **`celda`**).



Ventana Estilos

Las propiedades de formato de tabla CSS están disponibles en la ventana Estilos (*imagen siguiente*). Aquí puede acceder a las propiedades CSS para todos los componentes y subcomponentes de tabla (**`cuerpo`**, **`fila`**, **`columna`** y **`celda`**).

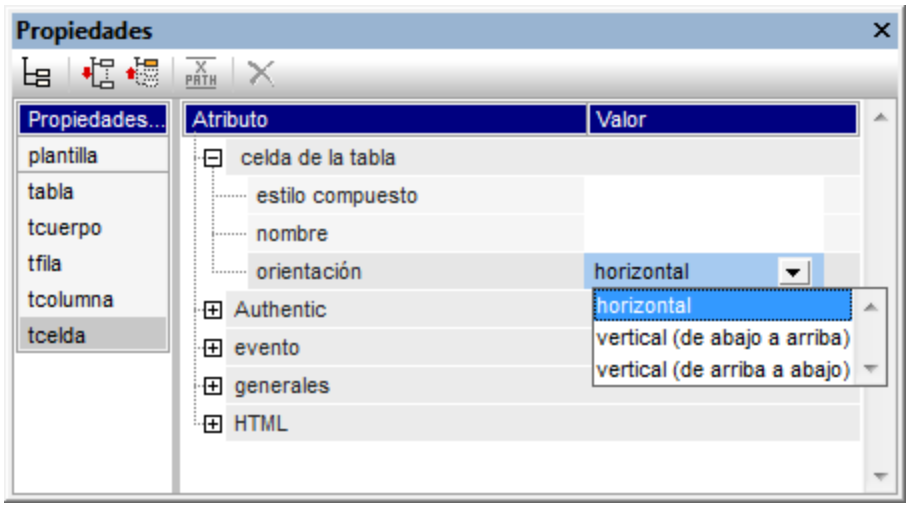
© 2016-2022 Altova GmbH



Nota: si todas las celdas de una fila están vacías, Internet Explorer contrae la fila. En casos así puede usar un truco que consiste en poner un espacio de no separación en las celdas.

Texto vertical

El texto de las celdas de la tabla se puede rotar 90 grados hacia la derecha o hacia la izquierda para que el texto esté en vertical y se pueda leer de arriba a abajo o de abajo a arriba respectivamente. Para rotar el texto seleccione el contenido de la celda de la tabla y en la ventana Propiedades (*imagen siguiente*) seleccione el componente **`tcelda`**. En el grupo de propiedades `celda de la tabla` seleccione el valor que desea usar para la propiedad `orientación`.



No olvide que:

- La rotación se aplica en el formato de salida pero no se observa en el diseño SPS.
- Esta propiedad está pensada para contenidos de tipo texto solamente.
- Además de en las celdas de tabla también puede aplicar esta propiedad al texto de los cuadros de texto 210.

Formato de tablas en las ventanas Propiedades y Estilos

Algunas propiedades de tabla están disponibles tanto en la ventana Propiedades como en la ventana Estilos. La tabla siguiente muestra las propiedades de tabla más importantes.

Componente de tabla	Ventana Propiedades	Ventana Estilos
`Tabla`	border, frame, rules; cellpadding, cellspacing; bgcolor; height, width *(reemplazado por height, width en la ventana Estilos); align*	borders y padding en el grupo `cuadro`; height, width en el grupo `detalles` (*reemplazan a height, width en la ventana Propiedades*); propiedades de `color`, `fuente` y `texto`
`Cuerpo`	align, valign	height, vertical-align; propiedades de `color`, `fuente` y `texto`
`Columna`	align, valign	width, vertical-align; propiedades de `color`, `fuente` y `texto`; propiedades de `cuadro`
`Fila`	align, valign	height, vertical-align; propiedades de `color`, `fuente` y `texto`; propiedades de `cuadro`
`Celda`	align, valign	height, width, vertical-align; propiedades de `color`, `fuente` y `texto`; propiedades de `cuadro`

 © 2016-2022 Altova GmbH

Alto y ancho

El alto y ancho de las tablas, filas, columnas y celdas se define en la ventana Estilos (en el grupo de estilos `detalles`). Cuando use el ratón para cambiar el tamaño de una tabla, columna o fila en la vista **Diseño**, los nuevos valores se introducen automáticamente en el estilo correspondiente de la ventana Estilos. Sin embargo, los estilos `height` y `width` no están disponibles para las celdas que tienen un valor `span` (`rowspan` o `colspan`).

Centrar la tabla

Para centrar una tabla, seleccione el componente **`tabla`** en la ventana Propiedades y defina el valor `center` para la propiedad `align` del grupo de propiedades `HTML`. También puede seleccionar la tabla entera y después hacer clic en el comando de menú **Tabla | Propiedades de la tabla**. La propiedad `align` está en el grupo de propiedades `HTML`.

Si el formato de salida es PDF, para centrar la tabla deberá definir más propiedades, dependiendo del procesador FOP que utilice. Según la especificación FO, la forma correcta de centrar una tabla es incluir el elemento `fo:table` en un elemento `fo:table-and-caption` y definir el valor `center` para el atributo `text-align` del elemento `fo:table-and-caption`. StyleVision no crea el elemento `fo:table-and-caption` automáticamente al insertar una tabla en el diseño, pero puede añadir este elemento como elemento definido por el usuario 151. No obstante, si usa un procesador Apache FOP, tenga en cuenta que el elemento `fo:table-and-caption` no siempre es compatible con el procesador, sino que depende de la versión FOP. Si no es compatible con su versión de procesador, puede usar un truco: convierta la tabla en una tabla de ancho fijo. Para ello, defina el valor `4in` o `120mm` para la propiedad `width` del grupo de propiedades `HTML` de la tabla (comando de menú **Tabla | Propiedades de tabla**).

Dar colores diferentes a filas alternas

Si quiere alternar colores de fondo en las filas de su tabla dinámica:

1. Seleccione el indicador de la fila a la que desea aplicar colores de fondo alternos. Tenga en cuenta que, al ser una tabla dinámica, un elemento corresponde a una fila y el diseño solamente contiene una fila.
2. Con el indicador de fila seleccionado, haga clic en **`tfila`** de la ventana Propiedades (columna **Propiedades para:**).
3. Seleccione la propiedad `bgcolor`.
4. Haga clic en el icono XPath de la barra de herramientas de la ventana Propiedades para abrir el cuadro de diálogo Editar expresión XPath 729. Escriba una expresión XPath como esta:

   ```
   if ( position() mod 2 = 0 ) then "white" else "gray"
   ```

   Esta expresión XPath especifica un color de fondo (`bgcolor`) blanco para las filas pares y un color de fondo (`bgcolor`) gris para las filas impares.

Puede usar el mismo mecanismo para crear un formato aún más complejo si quiere.

Numeración de las filas de una tabla dinámica

Las filas de una tabla dinámica se pueden numerar usando la función `position()` de XPath. Para ello, primero debe insertar una columna en la tabla que alojará los números de fila. Después debe insertar un cálculo automático en la celda de esta columna con la expresión XPath: `position()`. Puesto que el nodo de contexto

es el elemento que corresponde a la fila de la tabla dinámica, la función `position()` devuelve la posición que tiene cada elemento de fila en el conjunto de elementos de fila.

Encabezados y pies de tabla en el formato PDF de salida

Si una tabla abarca más de una página, el encabezado y el pie de tabla aparecen en cada página que ocupa la tabla. Debe tener en cuenta que:

- Si el pie de tabla tiene cálculos automáticos, el pie que aparece al final del segmento de tabla de cada página contiene los cálculos automáticos para toda la tabla (no solo para el segmento de tabla de esa página en particular).
- El encabezado y el pie de tabla no se pueden desactivar en unas páginas y en otras no (por ejemplo, si quiere que un pie de página aparezca al final de la tabla y no al final de cada página).

Para evitar que el encabezado o el pie de tabla aparezcan en la tabla con cada salto de página, use las propiedades (atributos) `table-omit-header-at-break` y/o `table-omit-footer-at-break` del elemento `table`. Estas propiedades están disponibles en la ventana Estilos, en el grupo de propiedades `XSL-FO`. Para omitir el encabezado o el pie de tabla cuando se produce un salto de página, especifique el valor `true` para el atributo correspondiente. Recuerde que el valor predeterminado es `false`. Por tanto, el formato predeterminado es que los encabezados y pies de tabla aparezcan cada vez que se produzca un salto de página.

Dividir el contenido de las celdas de la tabla con guiones

Si quiere dividir con guiones el texto de las celdas de tabla de su PDF, debe indicarlo explícitamente configurando el atributo **`hyphenate`** como **`true`** en los bloques (**`fo:block`**) del documento FO, conforme a la especificación XSL:FO.

5.6.6 Presentación de filas y columnas

Las opciones para ocultar filas y columnas son compatibles con el formato de salida HTML solamente. Estas características **no son compatibles con la vista Authentic** ni con XSLT 1.0 (es decir, debe seleccionar la versión XSLT 2.0 o XSLT 3.0 para el SPS).

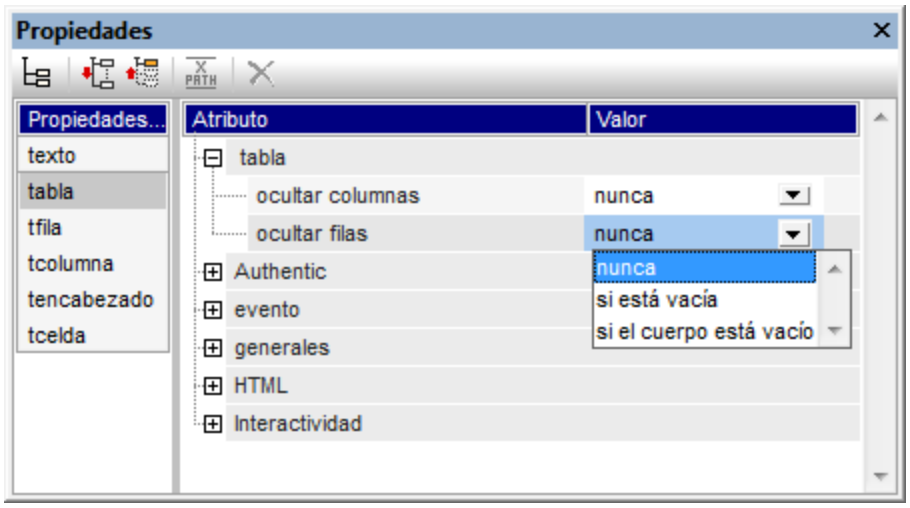
- Las filas y columnas vacías se pueden ocultar automáticamente.
- Cada columna puede tener un botón **Cerrar**, para que el usuario pueda cerrar las columnas que quiera.
- Los elementos de fila con relaciones descendientes se pueden mostrar con botones de expansión y contracción.

Ocultar filas y columnas vacías de forma predeterminada

Para ocultar las filas/columnas vacías en el formato HTML de salida:

1. En la vista **Diseño** seleccione la tabla o parte de ella (una columna, celda o fila).
2. En la ventana Propiedades seleccione el componente **`tabla`** y expanda el grupo de propiedades `tabla` (*imagen siguiente*).

 © 2016-2022 Altova GmbH



3. Seleccione el valor de las propiedades `ocultar columnas` y `ocultar filas`. Las opciones para estas dos propiedades son: (i) `nunca`, (ii) `si está vacía`, (iii) `si el cuerpo está vacío`. La opción `si está vacía` oculta la columna/fila si toda la columna/fila (incluido su encabezado y pie) está vacía. La opción `si el cuerpo está vacío` oculta la columna/fila si está vacía (incluso si el encabezado y el pie no están vacíos).

Nota: si se trata de una tabla no XBRL donde se combinaron varias celdas de una fila o de una columna, puede que la función para ocultar filas y tablas no funcione.

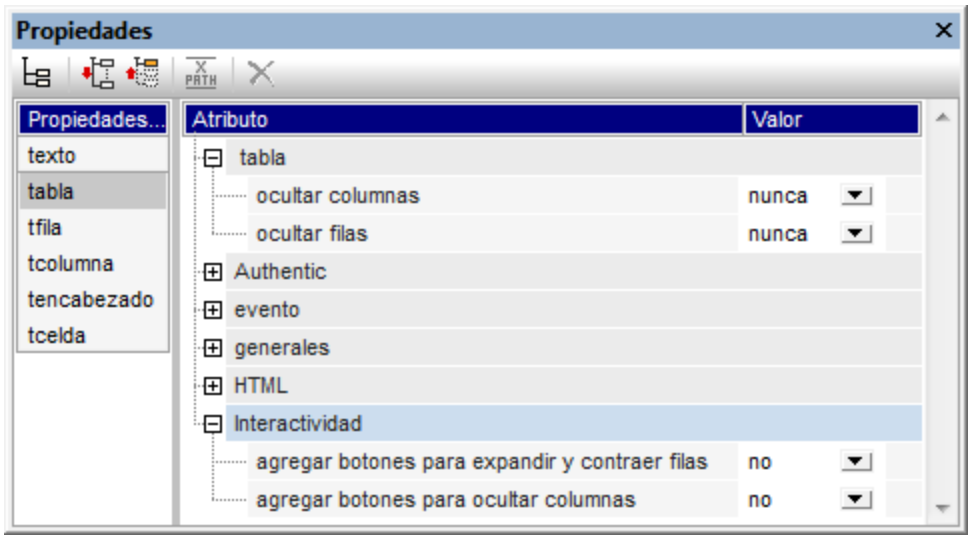
Interacción del usuario para ocultar columnas y expandir/contraer filas

En el diseño puede configurar la tabla para que en el formato de salida HTML cada columna de la tabla tenga un botón **Cerrar** (*imagen siguiente*). El usuario puede usar este botón para ocultar las columnas de la tabla una por una. Si se oculta alguna columna de la tabla, la primera columna incluye un símbolo en forma de **+** (*imagen siguiente*). Al hacer clic en el símbolo **+** se abren otra vez las ventanas que se ocultaron.

Balance Sheet (in Millions)	2004-09-30	2004-07-01 - 2004-09-30	2003-12-31	2004-01-01
Assets, Total	€21.49		€24.02	
Current Assets, Total	€10.65		€12.32	
Non Current Assets, Total	€10.85		€11.7	
Liabilities and Equity, Total	€21.49		€24.02	
Liabilities, Total	€8.9		€10.79	
Minority Interests				
Equity, Total	€12.59		€13.23	
Issued Capital and Reserves	€12.59		€13.23	

Además en el formato de salida HTML los elementos de fila que tienen elementos descendientes pueden incluir un símbolo de expansión/contracción (+/-) (*imagen anterior*). Al hacer clic en estos símbolos en el formato de salida HTML, el elemento de fila se expande o contrae. En el diseño puede especificar la sangría que se aplica a cada fila usando propiedades CSS.

Estas características se configuran en el grupo de propiedades `Interactividad` del componente **`tabla`** (en la ventana Propiedades):



Los valores posibles de estas propiedades son `sí` (la característica se añade) y `no` (la característica no se añade).

Temas relacionados

- Crear tablas estáticas 157
- Crear tablas dinámicas 158
- Tablas SPS en la vista Diseño 164
- Formato de tablas estáticas y dinámicas 165
- Tablas CALS/HTML 172

5.6.7 Tablas CALS/HTML

Una tabla CALS/HTML es una estructura XML jerárquica cuyos elementos: (i) definen la estructura de la tabla, (ii) especifican el formato de dicha tabla e (iii) incluyen el contenido de las celdas de dicha tabla. Esta estructura XML debe corresponderse con el modelo de la tabla CALS o HTML.

Para crear una tabla CALS/HTML en el diseño:

1. Defina la estructura XML como estructura de tabla CALS/HTML 172
2. Especifique estilos de formato para la tabla 175
3. Inserte la tabla CALS/HTML en el diseño SPS 176

Habilitar estructuras de tabla CALS/HTML para los documentos de salida

Un documento XML puede tener una estructura de datos que define la estructura y contenido de una tabla. Por ejemplo, la estructura de datos XML que aparece a continuación corresponde al modelo de tabla HTML y tiene los mismos nombres de elemento que el modelo de tabla HTML:

```
<table>
```

 © 2016-2022 Altova GmbH

```
  <tbody>
    <tr>
      <td/>
    </tr>
  </tbody>
</table>
```

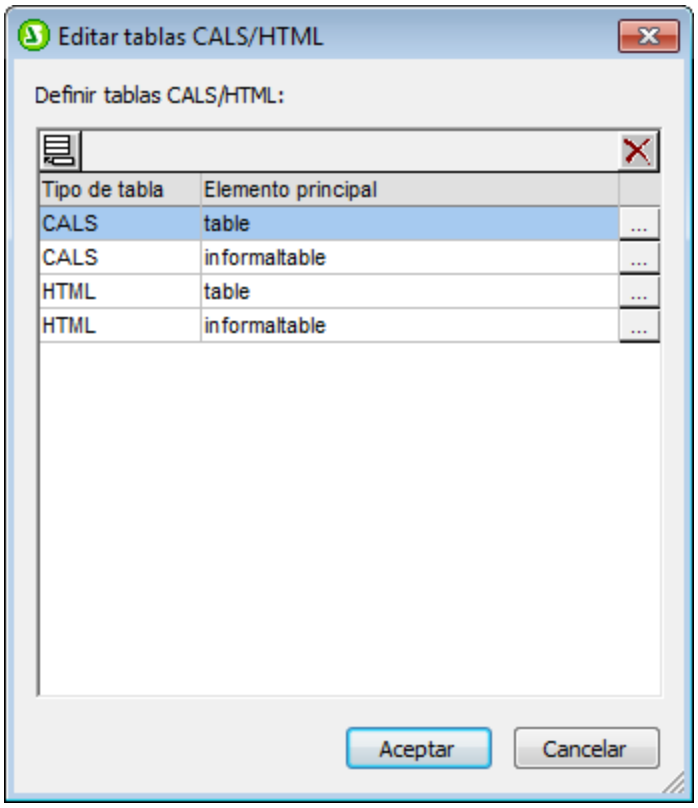
La estructura de datos XML también puede tener una estructura que corresponda al modelo de tabla HTML pero con nombres de elemento diferentes a los del modelo de tabla HTML. Por ejemplo:

```
<semester>
  <subject>
    <class>
      <student/>
    </class>
  </subject>
</semester>
```

Esta estructura de tabla, definida en el documento XML, se puede usar para generar una tabla en varios formatos de salida. Para ello es necesario definir esta estructura de datos XML como tabla CALS o HTML. Si la estructura de datos XML no se define como tabla CALS o HTML (opción predeterminada), los elementos de la estructura de datos se tratarán como elementos ordinarios no de tabla y no se añadirá marcado de tabla al documento de salida.

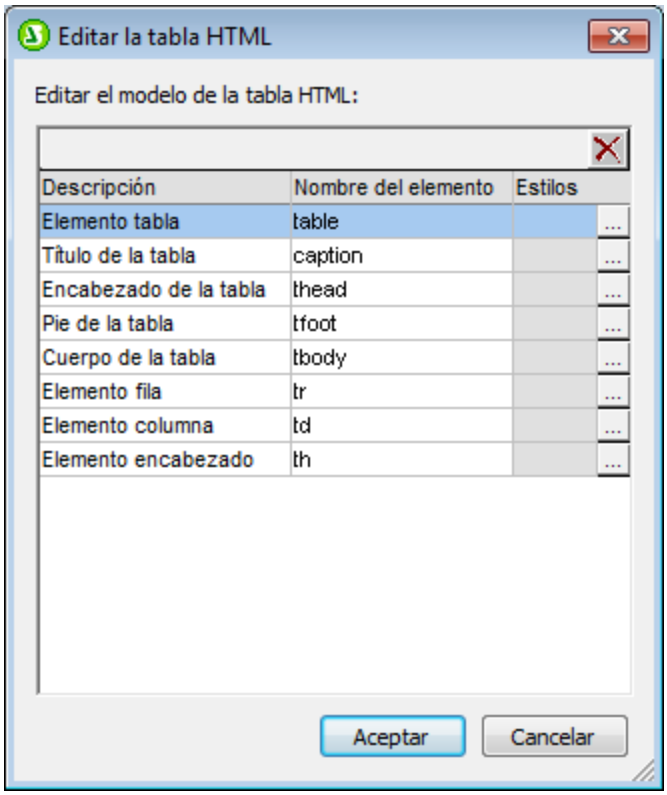
Para habilitar el marcado de tabla CALS/HTML en el documento de salida:

1. Seleccione el comando **Tabla | Editar tablas CALS o HTML**.
2. Aparece un cuadro de diálogo (*imagen siguiente*). Haga clic en el icono situado en la parte superior izquierda del panel para agregar una tabla CALS o una tabla HTML, dependiendo de si la estructura de datos sigue el modelo de tabla CALS o el de tabla HTML. (Para más información sobre el modelo de tabla CALS, consulte el modelo de tabla CALS del sitio web de OASIS.) Para ver un ejemplo abra el archivo SPS `HTMLTable1.sps`, situado en la carpeta `Basics` del proyecto `Examples`. El esquema fuente de este SPS contiene un elemento `table` con una estructura de tabla HTML. Por tanto, si quiere definir un elemento del esquema como elemento de tabla CALS o HTML, haga clic en el icono **Agregar tabla CALS o HTML** situado en la parte superior izquierda del panel del cuadro de diálogo y seleccione el comando correspondiente. En el ejemplo de la imagen siguiente los elementos `table` y `informaltable` se habilitaron como tablas CALS (y como tablas HTML).



3. Se abre otro cuadro de diálogo llamado "Editar la tabla CALS/HTML" que enumera los elementos del tipo de tabla seleccionado (*imagen siguiente*). Los nombres de elemento que se enumeran en este cuadro de diálogo son por defecto los nombres de elemento del modelo de tabla seleccionado (CALS o HTML). Si el esquema del SPS contiene elementos que se llamen igual que los elementos del modelo de tabla elegido, los nombres aparecen en negro (*como en la imagen siguiente*). Si los nombres de elemento no aparecen en el esquema del SPS, entonces los nombres de elemento aparecen en rojo. Puede cambiar el nombre de elemento del modelo de tabla por el nombre del elemento del esquema. Para ello haga doble clic en el nombre de elemento correspondiente y escriba el nombre del elemento del esquema.

 © 2016-2022 Altova GmbH



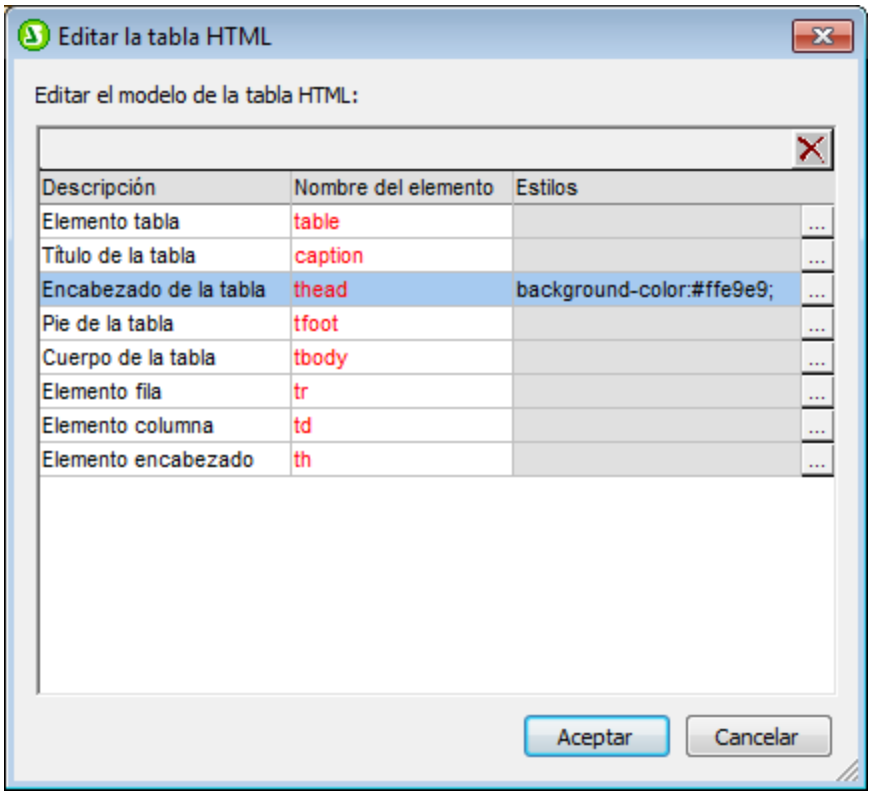
4. Haga clic en **Aceptar** para definir esta estructura de datos XML como tabla CALS o HTML.
5. Puede añadir todas las entradas de estructuras de datos que quiera (*primera imagen*). El mismo elemento principal se puede usar una vez para el tipo de tabla CALS y una vez para el tipo de tabla HTML.
6. Cuando termine de definir las estructuras de datos XML, haga clic en **Aceptar**.

Si definió una tabla CALS/HTML y la estructura de datos XML se insertó correctamente 176 como tabla CALS/HTML, entonces la estructura de datos se envía al formato de salida en forma de tabla. Para quitar la **definición de tabla CALS/HTML**, abra el cuadro de diálogo "Editar tablas CALS/HTML", seleccione la definición que desea eliminar y haga clic en el botón **Eliminar** (situado en la parte superior derecha del panel).

Formato de tabla

El formato de las tablas CALS/HTML se puede definir de dos formas diferentes:

1. Por medio de atributos de formato presentes en el documento XML de origen. Los modelos de tabla CALS y HTML admiten el uso de atributos de formato. Si existen en el documento XML de origen, estos atributos de formato se pasan a los atributos de presentación del marcado de tabla del documento de salida.
2. Por medio de los estilos definidos en la columna *Estilos* del cuadro de diálogo "Editar la tabla CALS/HTML" (*imagen siguiente*).



Para asignar un estilo a un elemento, haga clic en el botón **Agregar estilos** del elemento y asígnele los estilos que quiera en la ventana Estilos 58 que aparece. Cada estilo se añade como atributo CSS del elemento elegido. Observe que los estilos que se añaden por medio del atributo `style` tienen prioridad sobre los estilos añadidos como atributos CSS (como `bgcolor`). Por ejemplo, en el elemento `<thead style="background-color: red" bgcolor="blue"/>` el atributo `style="background-color: red"` tiene prioridad sobre el atributo `bgcolor="blue"`.

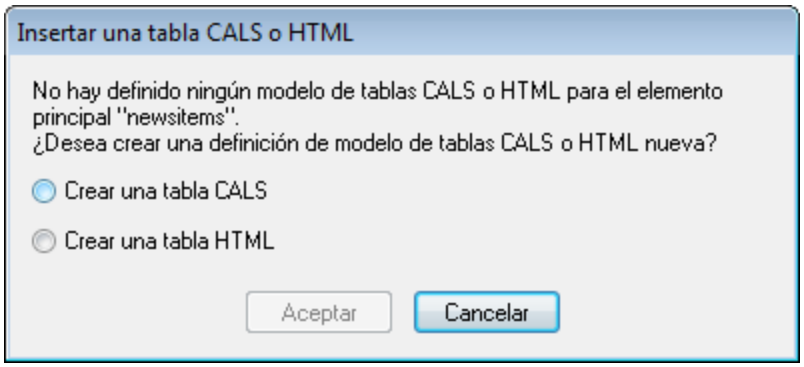
Para quitar un estilo que se asignó un elemento en la definición de tabla CALS/HTML, seleccione el elemento (por ejemplo, en la imagen anterior, se seleccionó el elemento `thead`) y haga clic en el botón Eliminar. Se quitan los estilos del elemento.

Insertar una tabla CALS/HTML en el diseño

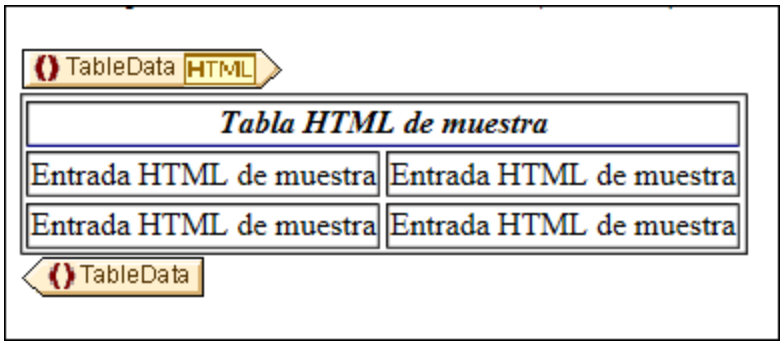
Las estructuras de tabla CALS/HTML se pueden insertar en el diseño de dos formas distintas:

1. El primario del elemento de tabla se inserta en el diseño como `(contenido)`. Cuando el contenido del primario se procesa, se procesa también el elemento de tabla. Si se habilita la presentación de tablas CALS/HTML, el elemento se representa en forma de tabla. De lo contrario, se representa en forma de texto.
2. El elemento de tabla se arrastra desde la estructura del esquema hasta el diseño. Al soltarlo en el diseño, puede eligir si con él se crea una tabla CALS/HTML (con el comando **Crear tabla CALS o HTML**). Si el elemento no se definió como tabla CALS o HTML 172, se abre el cuadro de diálogo "Insertar una tabla CALS o HTML" (*imagen siguiente*) donde puede definir el elemento como tabla CALS o HTML

 © 2016-2022 Altova GmbH



Si el elemento se creó como tabla CALS/HTML, en la posición elegida se inserta un marcador de posición para un elemento de diseño de tabla CALS/HTML (*imagen siguiente*).



Plantillas globales de elementos de tabla

Si se crearon plantillas globales 266 de los elementos de tabla siguientes, las plantillas globales se usarán en el resultado de la tabla CALS/HTML.

- Tablas CALS: `title` y `entry`.
- Tablas HTML: `caption`, `th` y `td`.

Archivos de ejemplo

En el proyecto `Examples` puede encontrar varios diseños de muestra que incluyen tablas CALS y HTML.

Temas relacionados

- Crear tablas estáticas 157
- Crear tablas dinámicas 158

5.7 Listas

En el diseño SPS se pueden crear dos tipos de listas:

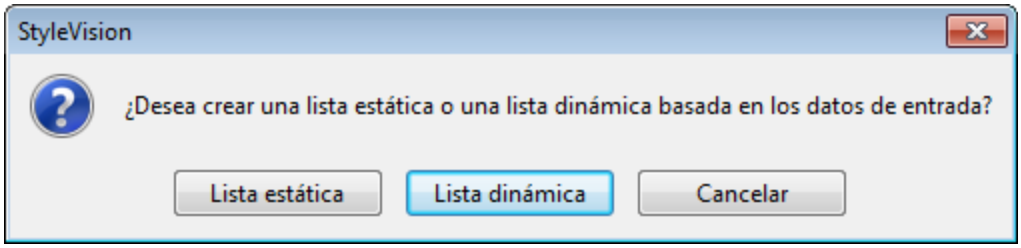
- Listas estáticas 178: sus contenidos se introducen directamente en el SPS. La estructura de las listas estáticas no deriva dinámicamente de la estructura del documento XML.
- Listas dinámicas 180: su estructura y contenido se deriva dinámicamente del documento XML.

Esta sección explica cómo crear estos dos tipos de listas en el diseño SPS.

5.7.1 Listas estáticas

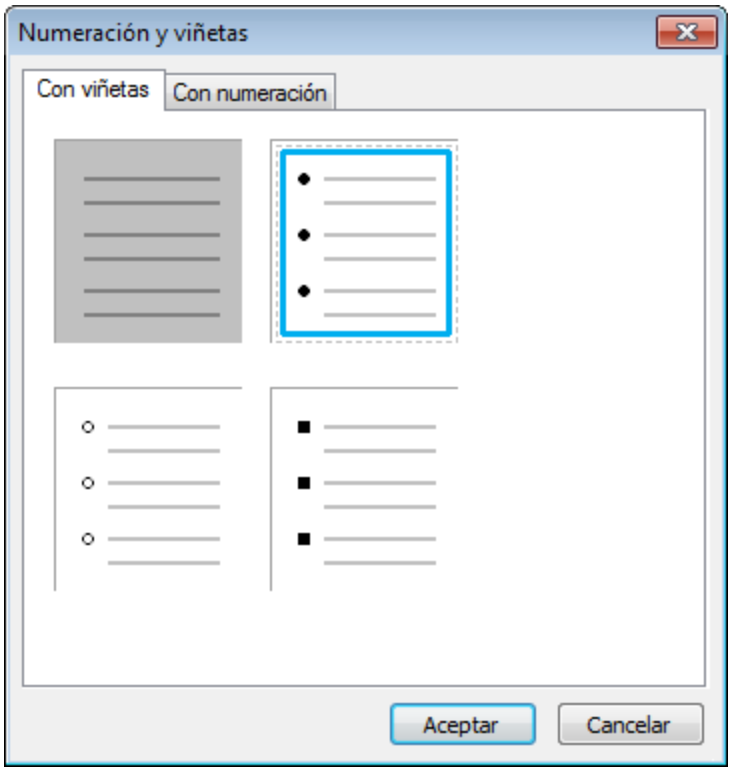
Una lista estática es una lista cuyo contenido lo introduce el usuario directamente en el SPS. Para crear una lista:

1. Ponga el cursor en la posición del diseño donde desea crear la lista estática y seleccione el comando de menú **Insertar | Numeración y viñetas** 823 (o haga clic en el icono **Numeración y viñetas** de la barra de herramientas Insertar elementos de diseño 755). Aparece un aviso preguntando si desea crear una lista estática o una lista dinámica (*imagen siguiente*).



2. Haga clic en el botón **Lista estática**. Aparece el cuadro de diálogo "Numeración y viñetas" (*imagen siguiente*).

© 2016-2022 Altova GmbH



3. Seleccione un marcador de elementos de lista y haga clic en **Aceptar**. Se crea una lista vacía.
4. Escriba el texto en el primer elemento de lista.
5. Pulse **Entrar** para crear un elemento de lista nuevo.

Para crear una lista anidada, ponga el cursor dentro del elemento de lista donde desea insertar la lista anidada y haga clic en el comando de menú **Insertar | Numeración y viñetas** 823. Siga construyendo la lista tal y como acabamos de explicar.

Nota: también puede crear una lista estática si pone el cursor en la posición donde desea crearla y hace clic en el icono de **Viñetas** o en el de **Numeración** de la barra de herramientas Formato 750. El primer elemento de lista se crea en la posición de inserción del cursor.

Convertir texto estático en una lista

Para convertir texto estático en una lista:

1. Resalte el texto que desea convertir en lista y seleccione el comando de menú **Incluir en | Numeración y viñetas** 839.
2. En el cuadro de diálogo que aparece elija un tipo de marcador de elementos de lista y haga clic en **Aceptar**.
3. Si el texto contiene un retorno de carro o avance de línea (que se inserta con la tecla **Entrar**), se crea un elemento de lista por cada fragmento de texto separado por un retorno de carro / avance de línea. Si un fragmento de texto de una línea está resaltado, el texto se convierte en elemento de lista. Después puede añadir todos los elementos de lista que quiera con solo pulsar la tecla **Entrar**.

Recuerde que el comando **Incluir en | Numeración y viñetas** 839 también está disponible en el menú contextual que aparece al hacer clic con el botón derecho en el diseño.

Listas en los documentos de salida en formato texto

Cuando se genera una lista en formato texto como documento de salida, cada elemento de esa lista viene indicado por un guion que le precede.

5.7.2 Listas dinámicas

Las listas dinámicas muestran el contenido de un conjunto de nodos secundarios que tienen el mismo nombre y cada nodo equivale a un elemento de lista. Es decir, la lista se crea a partir de un nudo cuyas instancias se convierten en elementos de lista.

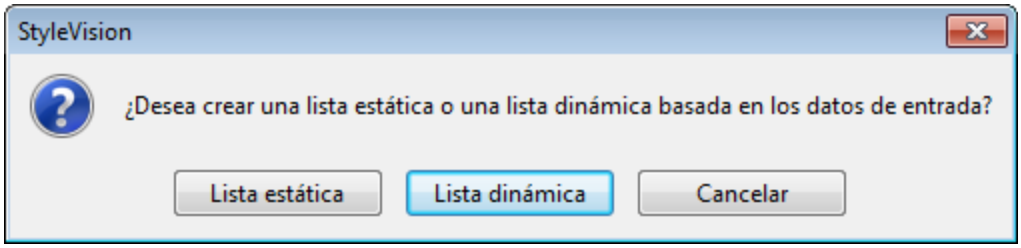
Mecanismo general

- Puede usar cualquier elemento para crear la lista.
- Cuando use un elemento para crear una lista, las instancias del elemento se usan para crear los elementos de la lista. Por ejemplo, si en un elemento `departamento` hay varios elementos `persona` (instancias) y quiere crear una lista con todas las personas del departamento, entonces debe crear la lista a partir del elemento `persona`.
- Después de crear la lista puede modificar el aspecto y contenido de la lista o de los elementos de lista insertando más contenido estático o dinámico (como texto, cálculos automáticos, contenido dinámico, etc.).

Crear una lista dinámica

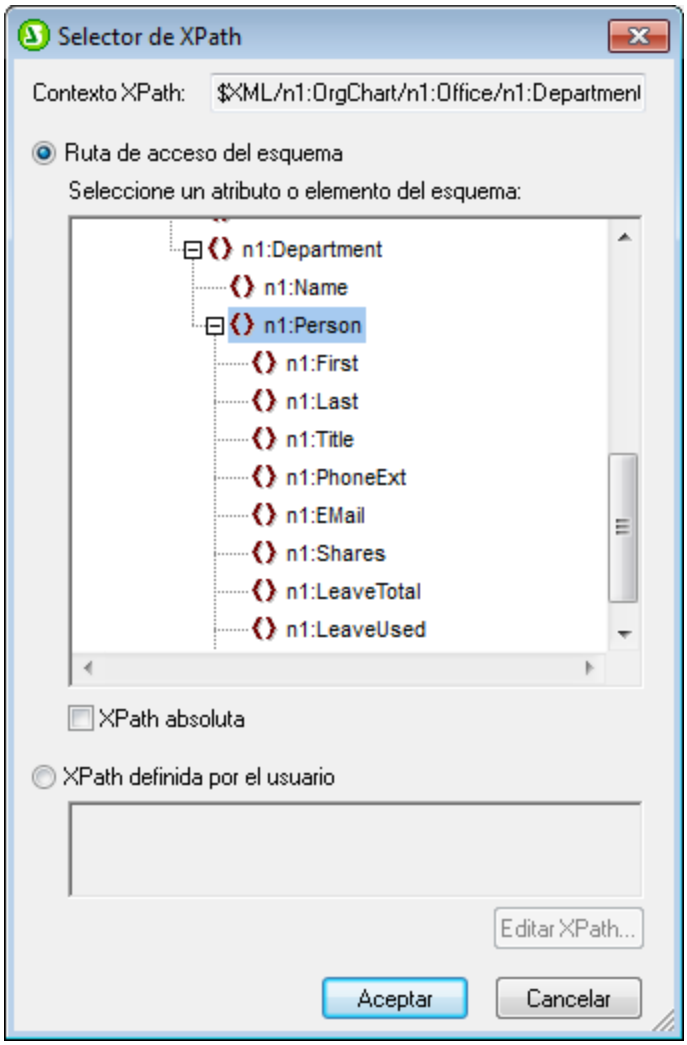
Para crear una lista dinámica:

1. Ponga el cursor en la posición del diseño donde desea crear la lista dinámica y haga clic en el comando de menú **Insertar | Numeración y viñetas** 823. Aparece un aviso preguntando si desea crear una lista estática o una lista dinámica (*imagen siguiente*).



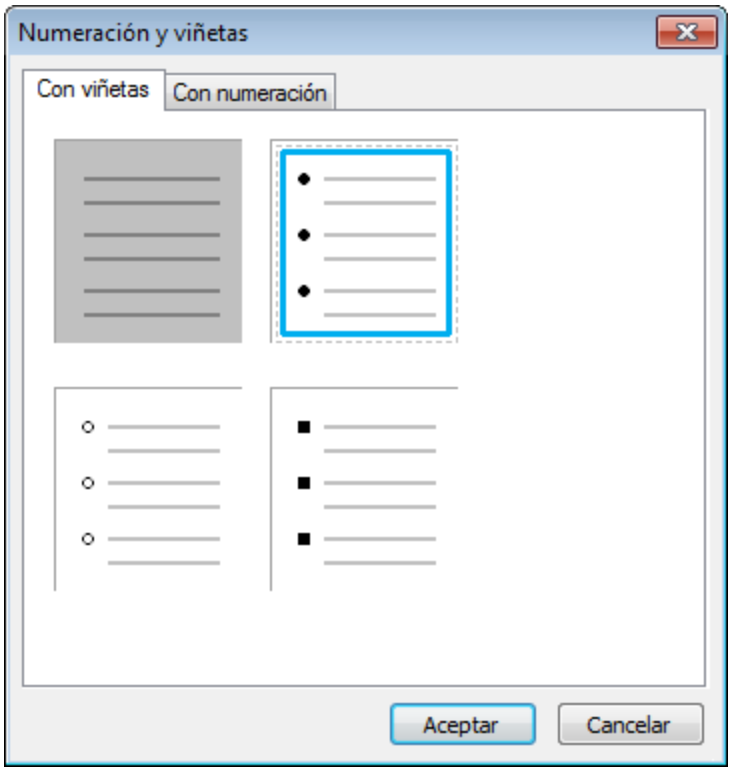
2. Haga clic en el botón **Lista dinámica**. Aparece el cuadro de diálogo "Selector de XPath" (*imagen siguiente*).
3. En este diálogo el campo *Contexto XPath* muestra el contexto de la posición de inserción y no se puede cambiar. Seleccione el nodo que desea usar para crear la lista dinámica. En la imagen siguiente, por ejemplo, el nodo de contexto es `n1:Department` y para crear la lista se seleccionó el nodo `n1:Person`. Esto significa que el contenido de cada nodo `n1:Person` equivaldrá a un elemento de lista.

 © 2016-2022 Altova GmbH



Si selecciona la opción *XPath definida por el usuario,* puede introducir una expresión XPath para seleccionar el nodo a partir del cual se debe crear la tabla dinámica. Cuando termine haga clic en **Aceptar**. Se abre el cuadro de diálogo "Numeración y viñetas" (*imagen siguiente*).

4. En el cuadro de diálogo "Numeración y viñetas" seleccione el tipo de lista que desea usar: una lista con viñetas (con una viñeta, un círculo o un cuadrado como marcador de elementos de lista) o una lista con numeración. Para terminar y crear la lista haga clic en **Aceptar**.



Listas en los documentos de salida en formato texto

Cuando se genera una lista en formato texto como documento de salida, cada elemento de esa lista viene indicado por un guion que le precede.

© 2016-2022 Altova GmbH

5.8 Contenido gráfico

En los diseños SPS puede usar contenido gráfico de dos formas diferentes:

- Como imágenes para el documento de diseño 183
- Como iconos de la barra de herramientas de la vista Authentic para aplicar marcado al documento XML (botones personalizados de barra de herramientas 851).

Nota: los documentos de salida en formato texto no incluyen gráficos.

Cuando se insertan imágenes en el documento de diseño, la ubicación de la imagen se puede especificar en el SPS directamente o se puede derivar de un nodo del documento XML. En el apartado Imágenes: datos insertados y URI 183 de esta sección explicamos cómo se puede especificar la ubicación de la imagen. En el apartado Tipos de imagen y formatos de salida 185 enumeramos los tipos de imagen que son compatibles en cada formato de salida. El apartado Referencia del usuario | Menú Authentic | Personalizar botones de la barra de herramientas 851 describe cómo puede definir iconos de barra de herramientas para la vista Authentic.

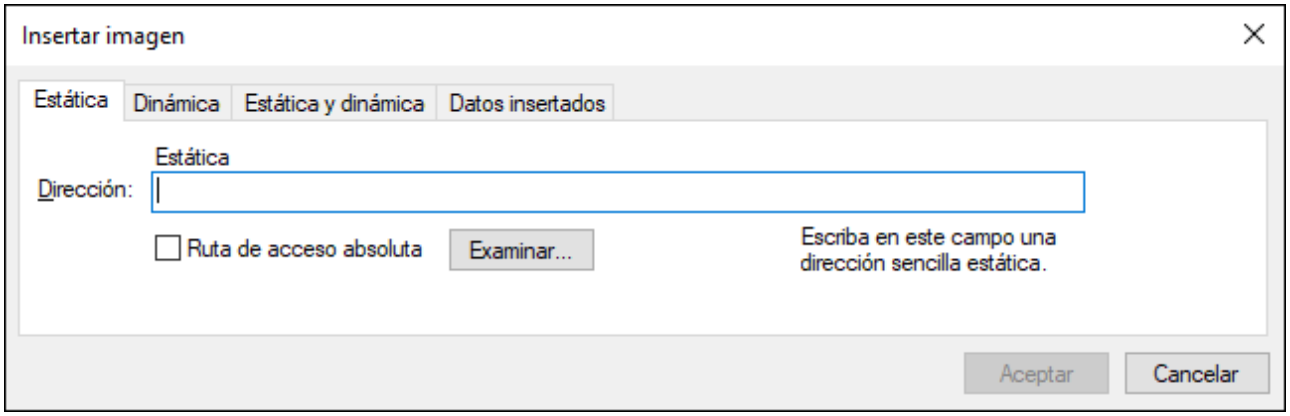
Propiedades de imagen

Las imágenes se pueden configurar en la ventana Propiedades. Para ello seleccione la imagen en el diseño y después, en la ventana Propiedades, (i) seleccione el componente imagen, (ii) seleccione el grupo de propiedades pertinente y (iii) seleccione la propiedad que desea definir. Por ejemplo, para definir el alto y ancho de la imagen, defina el valor de las propiedades `height` y `width` del grupo de propiedades `HTML`.

5.8.1 Imágenes: datos insertados y URI

Puede insertar imágenes en cualquier posición del documento de diseño. Estas imágenes aparecerán representadas en la vista Authentic y en los documentos de salida. En la vista **Diseño**, sin embargo, las imágenes insertadas aparecen señaladas con marcadores de posición o con miniaturas.

Para insertar una imagen haga clic en el comando de menú **Insertar | Imagen** 820. Aparece el cuadro de diálogo "Insertar imagen" (*imagen siguiente*).



Puede acceder a las imágenes de dos maneras distintas:

- Si la imagen es un archivo, introduzca su URI en el cuadro de diálogo "Insertar imagen".
- La imagen es texto codificado como base16 o base64 en un archivo XML.

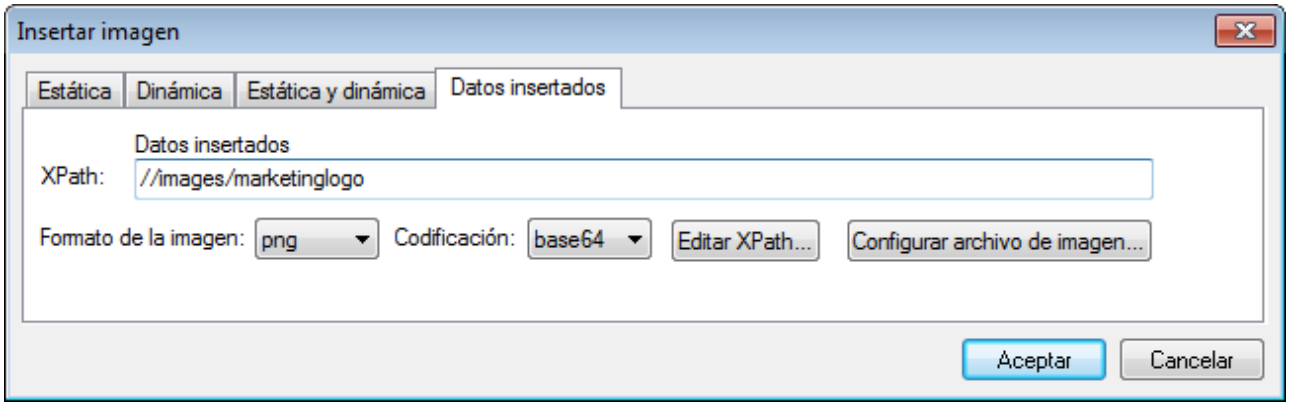
Insertar un archivo de imagen

Los archivos de imagen se insertan en el diseño especificando su URI. Al archivo se accede en tiempo de ejecución y se coloca en el documento. Tiene tres opciones a la hora de introducir el URI de la imagen en el cuadro de diálogo "Insertar imagen" (*imagen siguiente*):

- En la pestaña *Estática* puede introducir directamente el URI como URI absoluto o relativo. Por ejemplo, `nanonull.gif` (*URI relativo*, *ver más abajo* 185) y `C:/images/nanonull.gif` (*URI absoluto*).
- En la pestaña *Dinámica* puede introducir una expresión XPath que selecciona un nodo que contiene (i) un URI (absoluto o relativo) o (ii) un nombre de entidad sin analizar 469. Por ejemplo, la entrada `image/@location` seleccionaría el atributo `location` del elemento `image`, que es secundario del nodo de contexto (es decir, el nodo donde se debe insertar la imagen). El nodo del documento XML contiene el URI de la imagen. Para más información sobre el uso de URI de entidades sin analizar consulte este apartado 469.
- En las pestañas *Estática* y *Dinámica* puede introducir una expresión XPath en la parte dinámica y prefijos y sufijos en las partes estáticas. Por ejemplo, el prefijo estático podría ser `C:/XYZCompany/Personnel/Photos/`; la parte dinámica podría ser `concat(First, Last)`; y el sufijo estático podría ser `.png`. Como resultado obtendríamos este URI absoluto: `C:/XYZCompany/Personnel/Photos/JohnDoe.png`.

Insertar una imagen que es texto codificado

Los archivos XML pueden almacenar imágenes como texto codificado en base16 o base64. Esto tiene una ventaja: no es necesario acceder a la imagen desde otro archivo porque la imagen está presente en forma de texto en el archivo XML de origen. Para insertar una imagen disponible como texto codificado en el XML de origen, utilice la pestaña *Datos insertados* del cuadro de diálogo "Insertar imagen" (*imagen siguiente*).



Utilice una expresión XPath para apuntar al nodo del documento XML que contiene el texto codificado de la imagen. En el cuadro combinado *Formato de la imagen* indique en qué formato se debe generar la imagen (a partir del texto codificado se genera un archivo de imagen y el archivo se usa en el documento de salida). En el cuadro combinado *Codificación* puede seleccionar la codificación utilizada en el XML de origen. Esto permite a StyleVision leer el texto codificado correctamente (usando el formato de codificación especificado).

El cuadro de diálogo "Configuración del archivo de imagen" (que se abre con el botón **Configurar archivo de imagen...**) sirve para dar un nombre al archivo de imagen que se creará. Si prefiere no darle ningún nombre, StyleVision generará un nombre predeterminado.

 © 2016-2022 Altova GmbH

Acceder a la imagen durante la generación de resultados

Durante los procesos que generan los diferentes documentos de salida StyleVision accede a la imagen de varias maneras distintas y varias veces. A este respecto debe tener en cuenta que:

- Cada edición de StyleVision ofrece formatos de salida diferentes: (i) HTML en la edición Basic Edition, (ii) HTML y RTF en la edición Professional Edition y (iii) HTML, RTF, PDF y Word 2007+ en Enterprise Edition.
- En el cuadro de diálogo Propiedades 787 puede definir si las rutas de acceso relativas de la imagen son relativas al SPS o al archivo XML para la vista **Diseño** y la vista Authentic de StyleVision y de los productos de Altova.
- En el caso del formato HTML, el URI de la imagen se pasa al archivo HTML y a la imagen se accede desde el explorador. Por tanto, si la ruta a la imagen es relativa, debe ser relativa a la ubicación del archivo HTML. Para la vista previa **HTML** de StyleVision se crea un archivo HTML temporal en la misma carpeta que el archivo SPS y por ello las rutas relativas deben ser relativas a esta carpeta.
- En el caso del formato RTF, el URI de la imagen se pasa como vínculo de objeto al archivo RTF y se accede a él desde la aplicación RTF (que suele ser MS Word) cuando se abre el archivo. Si el URI es relativo, debe ser relativo a la ubicación del archivo RTF. Para la vista previa **RTF** de StyleVision se crea un archivo RTF temporal en la misma carpeta que el archivo SPS y por ello las rutas relativas deben ser relativas a esta carpeta.
- Independientemente de si el URI es relativo o absoluto, el proceso que representa la imagen en el formato de salida debe tener acceso a la imagen físicamente.

Editar las propiedades de la imagen

Para editar una imagen haga clic con el botón derecho en el marcador de posición de la imagen en la vista **Diseño** y seleccione **Editar URL** en el menú contextual. Aparece el cuadro de diálogo "Editar imagen". Este diálogo es igual que el diálogo "Insertar imagen" (*imagen anterior*) y en él puede cambiar las propiedades de la imagen. El cuadro de diálogo "Editar imagen" también se puede abrir desde la propiedad `URL` del grupo de propiedades `imagen` de la ventana Propiedades. Al grupo de propiedades `imagen` pertenece también la propedad `alt`, que ofrece texto alternativo para la imagen.

Eliminar imágenes

Para eliminar una imagen, selecciónela y pulse la tecla **Suprimir**.

5.8.2 Tipos de imagen y formatos de salida

La tabla que aparece a continuación enumera los tipos de imagen compatibles con los formatos de salida de StyleVision. Recuerde que no todas las ediciones de StyleVision ofrecen todos los formatos de salida:

- Enterprise Edition: HTML, Authentic, RTF, PDF y Word 2007+
- Professional Edition: HTML, Authentic y RTF
- Basic Edition: HTML.

Tipo de imagen	Authentic	HTML	RTF	PDF	Word 2007+

JPEG	✓	✓	✓	✓	✓
GIF	✓	✓	✓	✓	✓
PNG	✓	✓	✓	✓	✓
BMP	✓	✓	✓	✓	✓
TIFF	✓*	✓*	✓	✓	✓
SVG	✓*	✓*	✗	✓	✗
JPEG XR	✓	✓	✗	✗	✗

* *Ver notas:*

- En la vista **Diseño** las imágenes solamente aparecen si su ubicación es una URL estática (es decir, introducida directamente en el SPS).
- Para poder ver imágenes TIFF y SVG en la vista **Authentic** y en la vista **HTML**, es necesario Internet Explorer 9 (o superior).
- Para el formato de salida PDF, las imágenes TIFF solamente pueden estar enlazadas, no incrustadas. Por tanto, su tamaño no se puede modificar.
- Los documentos SVG deben estar en formato XML y en el espacio de nombres SVG.
- FOP emite un error si no encuentra un archivo de imagen o si no genera un PDF.
- Si se usa FOP para generar PDF, la representación de imágenes PNG exige tener instalada la biblioteca JIMI y que FOP pueda acceder a ella.
- Para más información sobre FOP, consulte su sitio web.

Archivo de ejemplo

Como ejemplo puede ver el archivo `Images.sps`, guardado en esta carpeta:

```
C:\Documents and Settings\<usuario>\Mis
      Documentos\Altova\StyleVision2023\StyleVisionExamples/Tutorial/Images
```

Ajustar el tamaño de las imágenes en formato de salida RTF

En el formato de salida RTF solamente se puede ajustar el tamaño de las imágenes JPG y PNG. A este respecto debe tener en cuenta que:

- El ajuste de tamaño solamente es compatible con diseños que utilicen XSLT 2.0 o XSLT 3.0, pero no con XSLT 1.0.

© 2016-2022 Altova GmbH

- Los atributos `height` y `width` de la imagen deben definirse en el grupo de propiedades `Detalles` de la barra lateral Estilos 58. Los atributos `height` y `width` del grupo de propiedades `HTML` (ventana Propiedades) no se utilizan.
- Solamente se admite el uso de unidades absolutas (px, cm, in, etc.). No se admite el uso de porcentajes ni el valor `auto`.
- Las imágenes JPG y PNG se incrustan en el archivo RTF. Esta incrustación se implementa usando una función de extensión de Altova para XSLT 2.0.

Incrustación de imágenes en el formato RTF

En el formato de salida RTF las imágenes se pueden incrustar (cuando se usa XSLT 2.0 o XSLT 3.0) o enlazar. La opción se debe indicar en cada archivo SPS por separado. Para incrustar las imágenes en su diseño:

1. Abra el cuadro de diálogo "Propiedades" del archivo (**Archivo | Propiedades**).
2. Haga clic en la pestaña *Imágenes* y marque la casilla *Incrustar imágenes* (marcada por defecto). Tenga en cuenta que las imágenes no se incrustan si la versión XSLT del SPS activo es XSLT 2.0 o XSLT 3.0.
3. Haga clic en **Aceptar** y guarde el archivo SPS.

Este proceso debe repetirse en todos los archivos SPS en los que desea incrustar imágenes.

Si no marca la casilla *Incrustar imágenes*, las imágenes se enlazan a la ruta de archivo de imagen especificada en las propiedades de la imagen (seleccione la imagen y seleccione la URL en la ventana Propiedades 59). Para más información sobre cómo se resuelven las rutas de acceso, consulte el apartado URI de imagen 183.

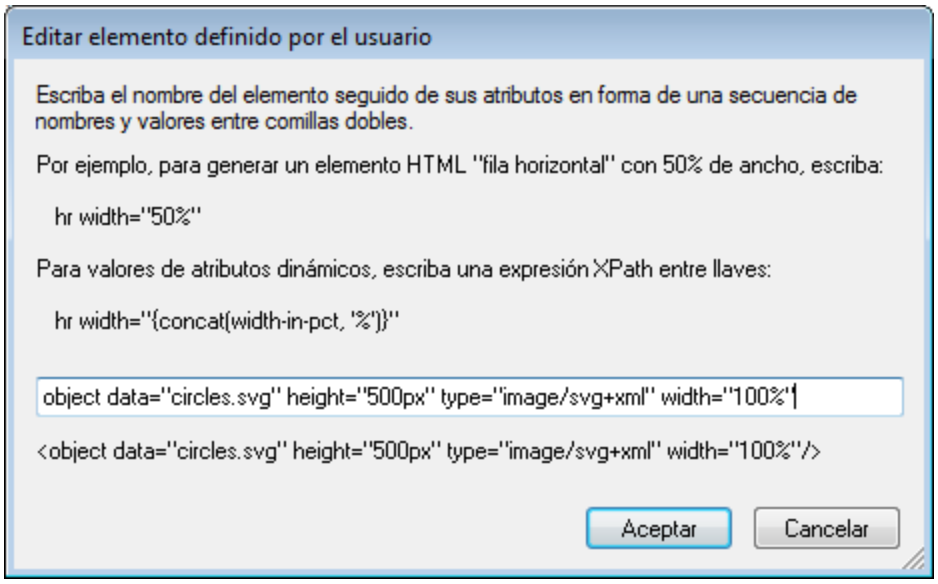
Nota: el formato RTF admite imágenes incrustadas de tipo EMF, JPG y PNG.

Imágenes SVG en HTML

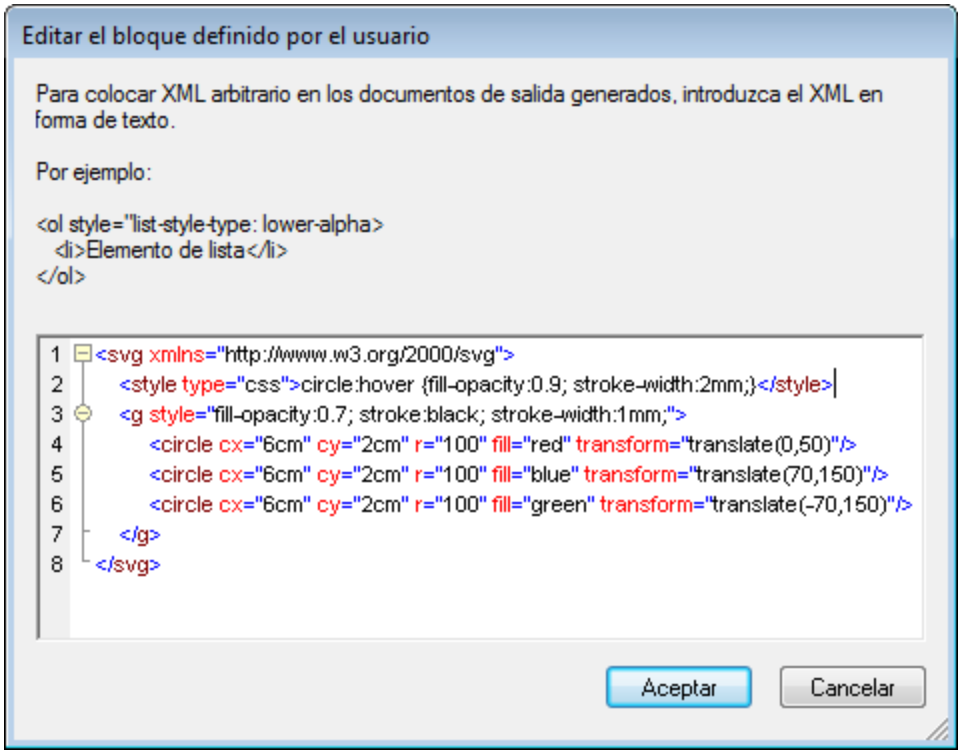
Cuando se usa como imagen un archivo SVG externo con código para eventos de ratón, el archivo SVG se representa dentro de la imagen y deja de ser interactivo. Esta limitación se puede resolver usando un archivo de imagen SVG externo como objeto o incluyendo el fragmento de código SVG en el diseño como bloque XML definido por el usuario.

Las imágenes SVG se pueden incluir en una página web de tres formas diferentes.

1. El archivo SVG externo se inserta como imagen 183: esto genera un elemento `<img>` en el archivo HTML generado y se pierde la interactividad.
2. El archivo SVG se inserta como objeto con ayuda de un elemento definido por el usuario 151 (*imagen siguiente*). Compruebe que inserta correctamente el atributo `type`: `type="image/svg+xml"`. Cuando se inserta de este modo, el objeto SVG sigue siendo interactivo y la función de desplazamiento del ratón funciona.



3. El archivo SVG se inserta con ayuda de un bloque XML definido por el usuario 152. La imagen siguiente muestra un ejemplo de fragmento de código SVG. En este caso el SVG sigue siendo interactivo. No olvide que el elemento `svg` no necesita estar en el espacio de nombres SVG si el método de salida es HTML 4.0 o 5.0, pero sí debe estarlo si el método de salida es XHTML.



 © 2016-2022 Altova GmbH

5.8.3 Ejemplo: una plantilla para imágenes

StyleVision viene con un archivo SPS de ejemplo que muestra el uso de imágenes en StyleVision. Este archivo está en la carpeta (Mis) Documentos [34]: `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\Images\Images.sps`. Este archivo SPS se divide en una introducción y tres partes:

- La primera parte muestra cómo usar los iconos de estado del texto en la vista Authentic (se pueden crear en las ediciones Enterprise y Professional solamente). Cuando abra el archivo `Images.xml` en la vista Authentic de XMLSpy, Authentic Desktop o Authentic Browser, podrá probar el uso de los iconos de estado del texto. Recuerde que los iconos de estado del texto no están disponibles en la vista previa **Authentic** de StyleVision y, por tanto, no podrá probar esta característica en StyleVision. Para aprender a crear iconos de estado del texto, consulte el apartado Referencia del usuario | Menú Authentic | Personalizar botones de la barra de herramientas [851].
- La segunda parte del archivo incluye una tabla que enumera qué formatos de imagen son compatibles con cada formato de salida de StyleVision. En la vista **Diseño** solamente se muestran correctamente las imágenes con URI estático. Todos los formatos de imagen de la tabla aparecen en la tercera parte del diseño SPS `Images.sps`.
- La tercera parte muestra todos los formatos de imagen compatibles con StyleVision uno detrás del otro. Tras abrir el archivo `Images.sps` en StyleVision, puede probar las diferentes vistas previas para ver qué aspecto tiene cada imagen en cada formato de salida. Puesto que la ubicación de la imagen es un nodo XML, también puede introducir la ubicación de otras imágenes en la vista Authentic y probar su aspecto en la pestañas de vista previa.

5.9 Controles de formulario

En StyleVision puede crear dispositivos de entrada de datos a partir de los nodos del documento XML. Los dispositivos de entrada de datos sirven para editar datos con facilidad en la vista Authentic. Por ejemplo, los campos de entrada indican al usuario de la vista Authentic que debe introducir datos en esa posición del documento, mientras que los cuadros combinados enumeran y limitan los valores que el usuario puede introducir. Cuando se introducen datos con dispositivo de entrada, los datos se insertan en el documento XML como contenido de un elemento o como valor de un atributo. En los documentos HTML, texto y RTF de salida, el dispositivo de entrada de datos se representa en forma de objeto, como en la vista Authentic. Recuerde que los dispositivos de entrada de datos aceptan la introducción de datos en el documento XML y, por tanto, no funcionan en los documentos HTML de salida.

Funcionamiento general
A continuación enumeramos los dispositivos de entrada disponibles en StyleVision, (i) sus métodos de entrada de datos y (ii) cómo aparece la salida en formato texto plano.

Dispositivo de entrada de datos	Datos en el documento XML	Salida en texto plano
Campo de entrada (cuadro de texto)	Texto introducido por el usuario	Texto introducido por el usuario
Campo de entrada multilínea	Texto introducido por el usuario	Texto introducido por el usuario
Cuadro combinado	La selección del usuario está asignada a un valor.	Valor seleccionado
Casilla de verificación	La selección del usuario está asignada a un valor.	`[]` (no seleccionado); `[x]` (seleccionado)
Botón de opción	La selección del usuario está asignada a un valor.	`()` (no seleccionado); `(o)` (seleccionado)
Botón	La selección del usuario está asignada a un valor.	Texto de botón

Los valores de texto introducidos en los campos de entrada se introducen directamente en el documento XML como contenido XML. En el caso del resto de dispositivos, la selección del usuario de la vista Authentic se asigna a un valor. StyleVision permite definir la lista de opciones que verá el usuario y asignar un valor XML a cada opción. Estas opciones y sus valores se suelen definir en un cuadro de diálogo.

Uso general

Siga estas instrucciones para crear un dispositivo de entrada de datos:

1. Arrastre un nodo desde la ventana Estructura del esquema hasta la vista **Diseño** y suéltelo en la posición donde desea crear el dispositivo de entrada.
2. Aparece un menú contextual. Seleccione el dispositivo de entrada que desea crear a partir del nodo XML.

 © 2016-2022 Altova GmbH

3. Dependiendo del dispositivo elegido, aparece un diálogo o no. Si aparece un diálogo, introduzca la información necesaria y haga clic en **Aceptar**.

Para **volver a abrir y editar** las propiedades de un dispositivo de entrada, selecciónelo (el dispositivo, no el nodo) y edite sus propiedades en la barra lateral Propiedades.

Nota:

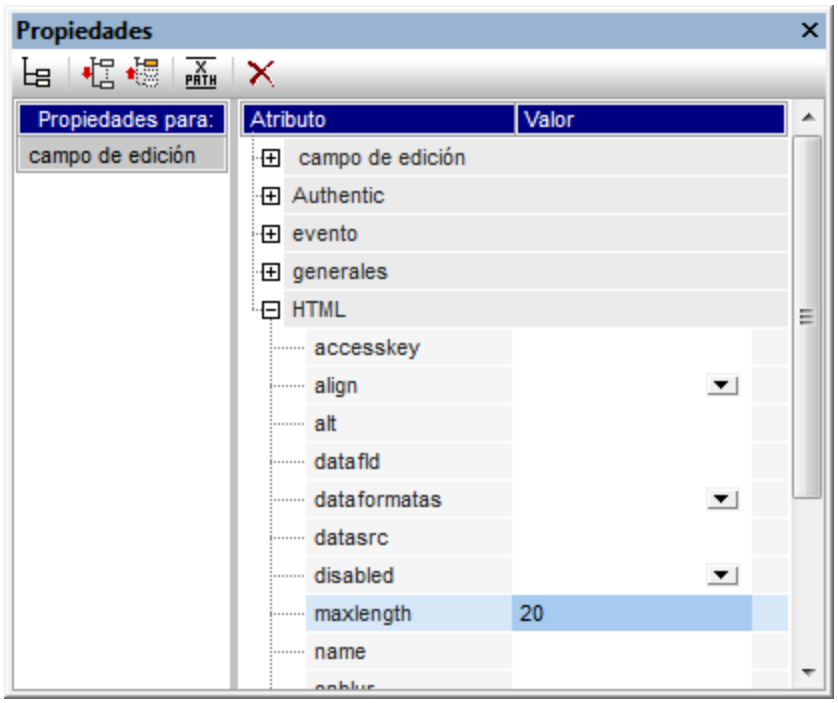
- En los dispositivos de entrada solamente se pueden insertar datos desde la vista Authentic.
- También puede crear dispositivos de entrada de datos cambiando el tipo de un componente de diseño de un nodo. Haga clic con el botón derecho en el componente en la vista **Diseño** y seleccione **Convertir en**.
- En los documentos HTML y RTF de salida, los datos seleccionados por el usuario se presentan en el documento de salida. Si cambia el valor de un dispositivo de entrada de datos en un documento HTML, el valor de texto no cambia ni en el documento XML ni en el documento HTML.
- En algunos casos (p. ej. las casillas de verificación) el dispositivo de entrada de datos no se puede representar correctamente en formatos impresos y, por tanto, el dispositivo se reproduce de otra forma.

5.9.1 Campos de entrada y campos de entrada multilínea

Puede crear un campo de entrada o un campo de entrada multilínea en su SPS si arrastra un nodo desde la Estructura del esquema hasta la vista **Diseño**. El texto que introduzca el usuario de la vista Authentic en estos campos se insertará en el nodo XML utilizado para crear el campo de entrada. El contenido del nodo aparece en el campo de entrada o campo de entrada multilínea.

Editar las propiedades de los campos de entrada

Para modificar las propiedades HTML de los campos de entrada, seleccione el campo de entrada y cambie sus propiedades *HTML* en la ventana Propiedades (*imagen siguiente*).



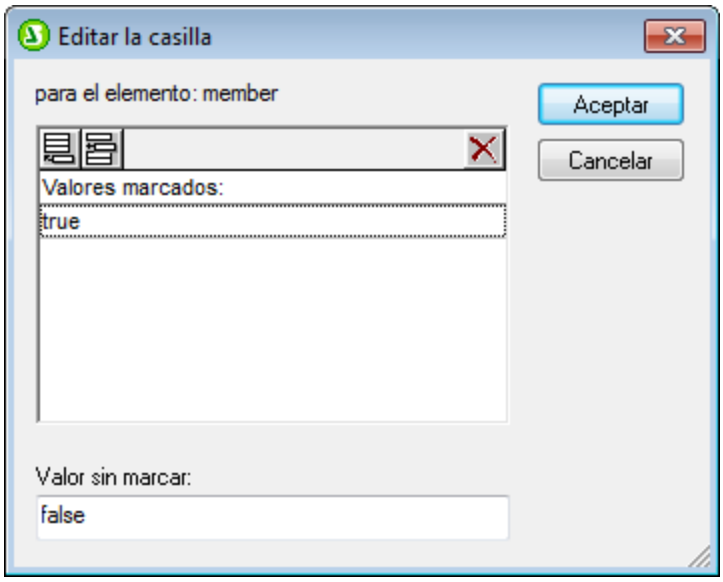
Por ejemplo, seleccione el campo de entrada y en la ventana Propiedades seleccione `campo de edición`, después expanda el grupo de propiedades *HTML* y seleccione la propiedad `maxlength`. Por último haga doble clic en el campo de valor de `maxlength` e introduzca un valor.

Nota: no está permitido insertar secciones CDATA dentro de campos de entrada (es decir, en cuadros de texto). Las secciones CDATA solamente se pueden introducir en los elementos que aparecen como componentes de contenido de texto en la vista Authentic.

5.9.2 Casillas de verificación

Puede crear casillas de verificación en su diseño y utilizarlas como dispositivos de entrada de datos. Con ellas se obliga al usuario a elegir una de dos opciones. En el cuadro de diálogo "Editar la casilla" (*imagen siguiente*) puede especificar qué valores XML se asignan a la casilla cuando está marcada y cuando no lo está.

 © 2016-2022 Altova GmbH



En el ejemplo de la imagen anterior, se creó una casilla de verificación a partir del elemento `member`. Si el usuario de la vista Authentic marca la casilla, se introduce el valor `true` como valor del elemento `member`. Si el usuario no marca la casilla, se introduce el valor `false` como valor XML del elemento `member` (tal y como se definió en el cuadro de diálogo).

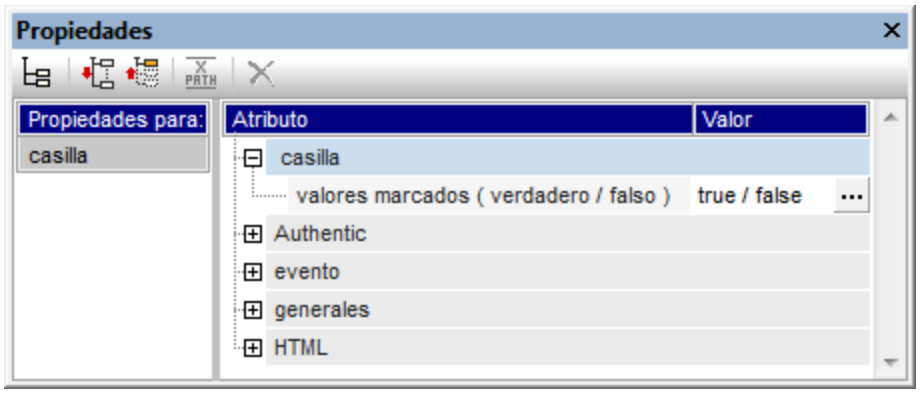
Nota: cuando se cree un elemento `member` nuevo (o una casilla nueva) en la vista Authentic, su valor XML estará vacío (es decir, no toma el valor sin marcar). El valor sin marcar solamente se introduce si el usuario activa la casilla y después la desactiva. Si quiere que el nodo tenga un valor predeterminado, cree un archivo de plantilla XML que contenga el valor predeterminado.

Nota: las casillas de verificación en los documentos de salida en formato texto aparecen como corchetes `[]` si no están seleccionadas y `[x]` si sí lo están.

Abrir el cuadro de diálogo "Editar la casilla"

Al crear una casilla de verificación a partir de un nodo XML aparece el cuadro de diálogo "Editar la casilla". Si quiere volver a abrir este cuadro de diálogo:

1. Seleccione la casilla de verificación en la vista **Diseño**.
2. En la ventana Propiedades seleccione **`casilla`** y después el grupo de propiedades *casilla* (*imagen siguiente*).



3. Después haga clic en el botón **Editar** ... de la propiedad `valores marcados` y aparece el cuadro de diálogo "Editar la casilla".

Nota: para modificar las propiedades HTML de una casilla de verificación, selecciónela y cambie sus propiedades *HTML* en la ventana Propiedades.

5.9.3 Cuadros combinados

Un cuadro combinado ofrece al usuario de la vista Authentic una lista desplegable con varias opciones. La opción seleccionada por el usuario está asignada a un valor, que se introduce en el documento XML. Los valores XML asignados a cada opción de la lista desplegable se especifican en el diseño SPS.

Estas asignaciones de valores se pueden hacer de varias formas en el cuadro de diálogo "Editar el cuadro combinado":

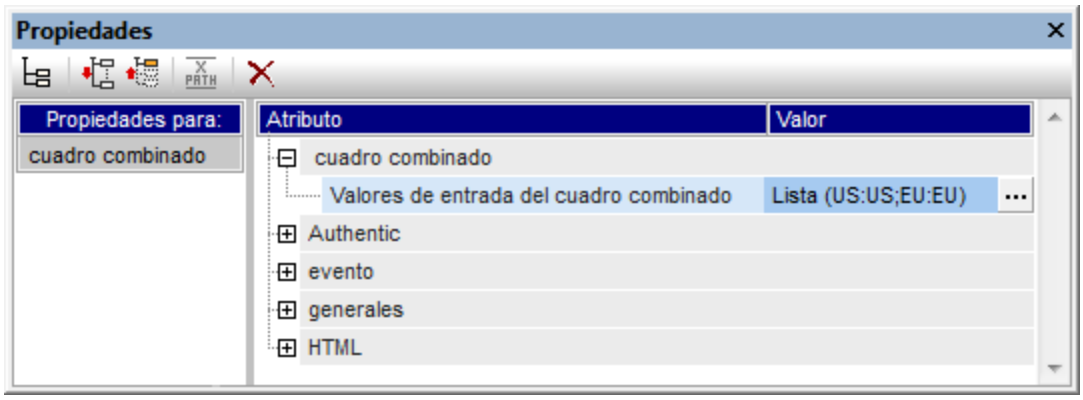
- A partir de las enumeraciones del esquema para el nodo seleccionado. En este caso, la entrada visible (en la lista desplegable) será la misma que el valor XML.
- A partir de una lista definida en el cuadro de diálogo "Editar el cuadro combinado". Introduzca la entrada visible y su valor XML correspondiente, que puede ser diferente.
- A partir de la secuencia de resultados de una expresión XPath relativa al nodo actual. Los elementos de la secuencia de resultados aparecen como entradas en la lista desplegable y la opción de la lista seleccionada por el usuario de la vista Authentic se introduce como valor del nodo. Se trata de una potente manera de usar entradas dinámicas en el cuadro combinado. El nodo que se utiliza para crear el cuadro combinado es muy importante. Por ejemplo, imagine que tiene un elemento llamado `ListaNombres` que puede incluir un número ilimitado de elementos `Nombre`, que a su vez pueden tener elementos secundarios `Nombre` y `Apellido`. Si crea un cuadro combinado a partir del elemento `Nombre` y selecciona el elemento secundario `Apellido` para la lista de valores, en la vista Authentic, obtendrá tantos cuadros combinados como elementos `Nombre` haya en el documento XML y cada cuadro combinado incluirá en la lista desplegable el contenido de los secundarios `Apellido`. Para crear un solo cuadro combinado cuya lista desplegable incluya todos los elementos `Apellido` es necesario usar el nodo primario `ListaNombres` y después seleccionar el elemento `Apellido` en la expresión XPath.

Abrir el cuadro de diálogo "Editar el cuadro combinado"

Cuando se crea un cuadro combinado en el diseño SPS, aparece el cuadro de diálogo "Editar el cuadro combinado". También puede insertar un cuadro combinado en el diseño con el comando de menú **Insertar | Insertar controles de formulario | Cuadro combinado**). Si más tarde quiere volver a abrir el cuadro de diálogo "Editar el cuadro combinado":

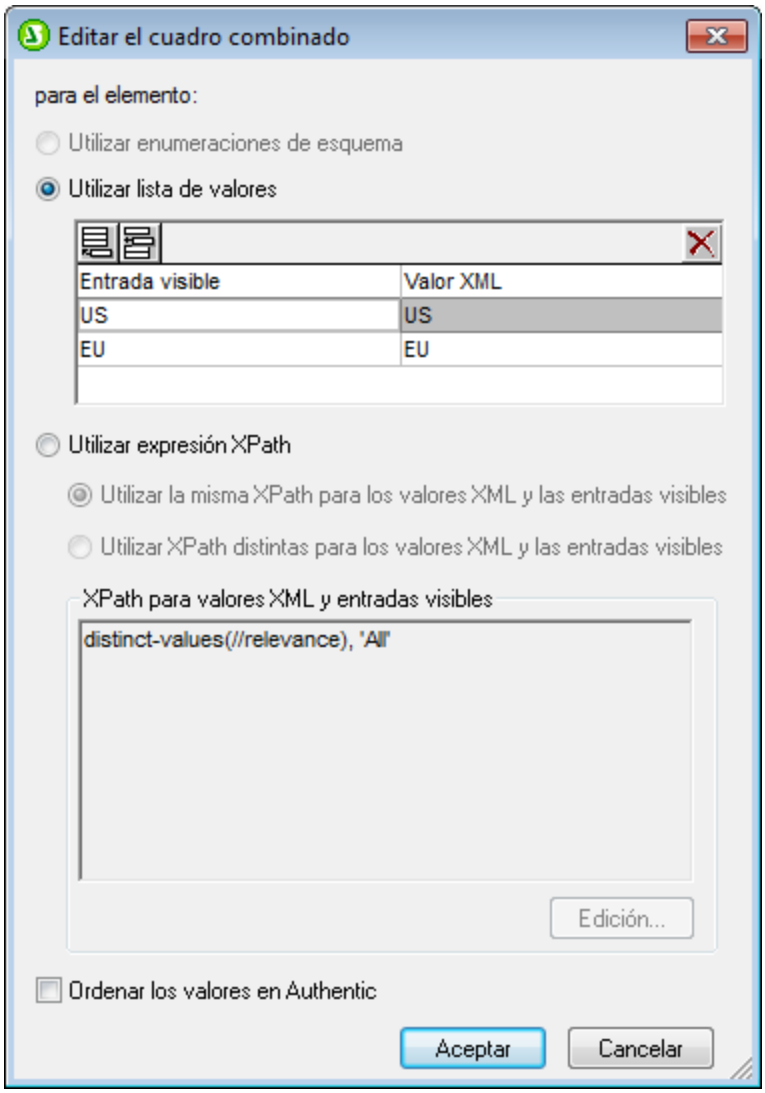
1. Seleccione el cuadro combinado en el diseño.
2. En la ventana Propiedades seleccione el componente **`cuadro combinado`** y después expanda el grupo de propiedades *cuadro combinado* (*imagen siguiente*).

 © 2016-2022 Altova GmbH



3. Haga clic en el botón **Editar** ... de la propiedad `Valores de entrada del cuadro combinado` y aparece el cuadro de diálogo "Editar el cuadro combinado".

Usar el **cuadro de diálogo "Editar el cuadro combinado"**



Para definir las opciones y los valores del cuadro combinado:

1. Seleccione un método para definir las opciones y los valores: (i) *Utilizar enumeraciones de esquema*, (ii) *Utilizar lista de valores* o (iii) *Utilizar expresión XPath*.
2. Si selecciona el botón de opción *Utilizar enumeraciones de esquema*, las enumeraciones del esquema asignadas al nodo se introducen automáticamente (i) como entradas visibles de la lista desplegable del cuadro combinado y (ii) sus correspondientes valores XML (*imagen siguiente*). En el panel situado bajo la opción *Utilizar lista de valores* la columna *Entrada visible* muestra las opciones que aparecerán en la lista desplegable del cuadro combinado. Cada entrada visible de la lista desplegable tiene asignado un valor XML. El valor XML correspondiente a la entrada visible que seleccione el usuario de Authentic será el valor XML que se introduce en el archivo XML. En la imagen siguiente, por ejemplo, las entradas visibles y sus valores aparecen atenuados porque todos ellos se obtienen de las enumeraciones del esquema y no se pueden editar.

 © 2016-2022 Altova GmbH

Si selecciona el botón de opción *Utilizar lista de valores*, podrá insertar, anexar, editar y eliminar opciones para la lista desplegable del cuadro combinado, así como sus valores XML. Estos cambios se realizan en el panel situado justo debajo. También puede usar una expresión XPath para crear las entradas visibles y sus valores XML. Los elementos de la secuencia que devuelve la expresión XPath se usarán para las entradas visible y los valores XML. Además puede especificar: (i) si se utiliza la misma expresión XPath para las entradas visibles y para los valores XML o (ii) si se utilizan expresiones XPath distintas. En caso de utilizar expresiones XPath distintas, la correspondencia entre una entrada visible y un valor XML se determina mediante una asignación de índice entre los elementos de las dos secuencias. Si las secuencias no tienen el mismo número de elementos, se emite un error.

3. Si quiere que los elementos de la lista desplegable del cuadro combinado aparezcan ordenados en la vista Authentic, marque la casilla *Ordenar los valores en Authentic*.
4. Para terminar haga clic en **Aceptar**.

Nota

- Utilice una expresión XPath para seleccionar las entradas de la lista desplegable del cuadro combinado y así podrá crear cuadros combinados con entradas dinámicas a partir del archivo XML mismo.
- Si los elementos de la lista desplegable del cuadro combinado se obtienen de las enumeraciones del esquema, los elementos aparecen por orden alfabético. Si los elementos se obtienen de un archivo XML, entonces aparecen en orden de documento. Si los elementos se obtienen de una BD, debe definir como esquema principal el esquema de la BD. Si los elementos se obtienen de una BD que no es el esquema principal, debe incluir en el diseño una plantilla para la fila de la BD a la que apunta la expresión XPath, incluso si deja la plantilla vacía. Además, en estos casos debe asegurarse de que se recuperan todas las instancias de la fila a la que se apunta con la expresión XPath.
- En los documentos de salida en formato texto los cuadros combinados muestran el valor seleccionado.

5.9.4 Botones y botones de opción

Puede crear dos tipos de botones en su diseño SPS: botones corrientes y botones de opción. Los botones de opción permiten al usuario de la vista Authentic introducir datos en el archivo XML. Los botones corrientes **no permiten** introducir datos en la vista Authentic, pero se pueden utilizar para desencadenar eventos en los documentos HTML de salida.

Botones de opción
Con los botones de opción puede dar al usuario la oportunidad de elegir una opción entre varias alternativas. Cada botón de opción que inserte en el diseño SPS se asigna a un valor XML y el usuario selecciona uno de ellos. Los botones de opción de un nodo se excluyen mutuamente, solamente se puede seleccionar un botón de opción y su valor XML asociado se introduce como valor del nodo. Por tanto, debe crear varios botones de opción a partir del nodo en el que desea introducir datos con ayuda de este dispositivo. Por cada botón de opción debe introducir también (i) un texto estático que indique su valor al usuario y (ii) un valor XML único. Para editar el valor XML del botón de opción en la ventana Propiedades, seleccione el botón de opción y haga clic en el botón **Editar** de la propiedad `botón de opción`. A continuación aparece el cuadro de diálogo "Editar el botón de opción".

Botones
También puede introducir un botón corriente con un texto. Estos botones sirven para asociar scripts a eventos de botón en los documentos HTML de salida. Sin embargo, tenga en cuenta que los botones corrientes no tienen asignados valores XML y que no permiten introducir datos en la vista Authentic. No obstante, el usuario de la vista Authentic puede seleccionar el valor del nodo primario de un botón 464 y puede diseñar el SPS para que la presentación cambie dependiendo de lo seleccionado por el usuario de la vista Authentic.

Nota: seleccione el botón para modificar sus propiedades HTML en la ventana Propiedades.

Nota: los botones de opción en los documentos de salida en formato texto aparecen como paréntesis `()` si no están seleccionadas y `(o)` si sí lo están. Los botones solamente generan el texto del botón en la salida como texto.

 © 2016-2022 Altova GmbH

5.10 Vínculos

En el diseño SPS puede crear vínculos (o hipervínculos) a marcadores del documento, así como a recursos externos (p. ej. páginas web). También puede crear vínculos para generar delimitadores dinámicamente. StyleVision ofrece una gran flexibilidad a la hora de generar URI de destino para los hipervínculos.

Para más información sobre cómo crear marcadores dinámicos y estáticos en el documento y cómo crear vínculos a los marcadores y a documentos externos, consulte la sección Marcadores e hipervínculos 364.

Nota: los vínculos no aparecen como tal en la salida en formato texto.

5.11 Códigos de barras

Los códigos de barras son un elemento de diseño compatible solamente con el **modo XSLT 2.0 o XSLT 3.0** (pero no con XSLT 1.0). Este elemento de diseño sirve para generar códigos de barras (*imagen siguiente*) en los documentos de salida. Para ello elija la posición del diseño donde desea introducir el código de barras, inserte el código de barras 200 y especifique sus propiedades 202.

Importante: para que funcionen los códigos de barras es necesario tener instalado Java Runtime Environment versión 1.4 (o superior) de 32 o 64 bits (dependiendo de la versión de StyleVision que tenga instalada).

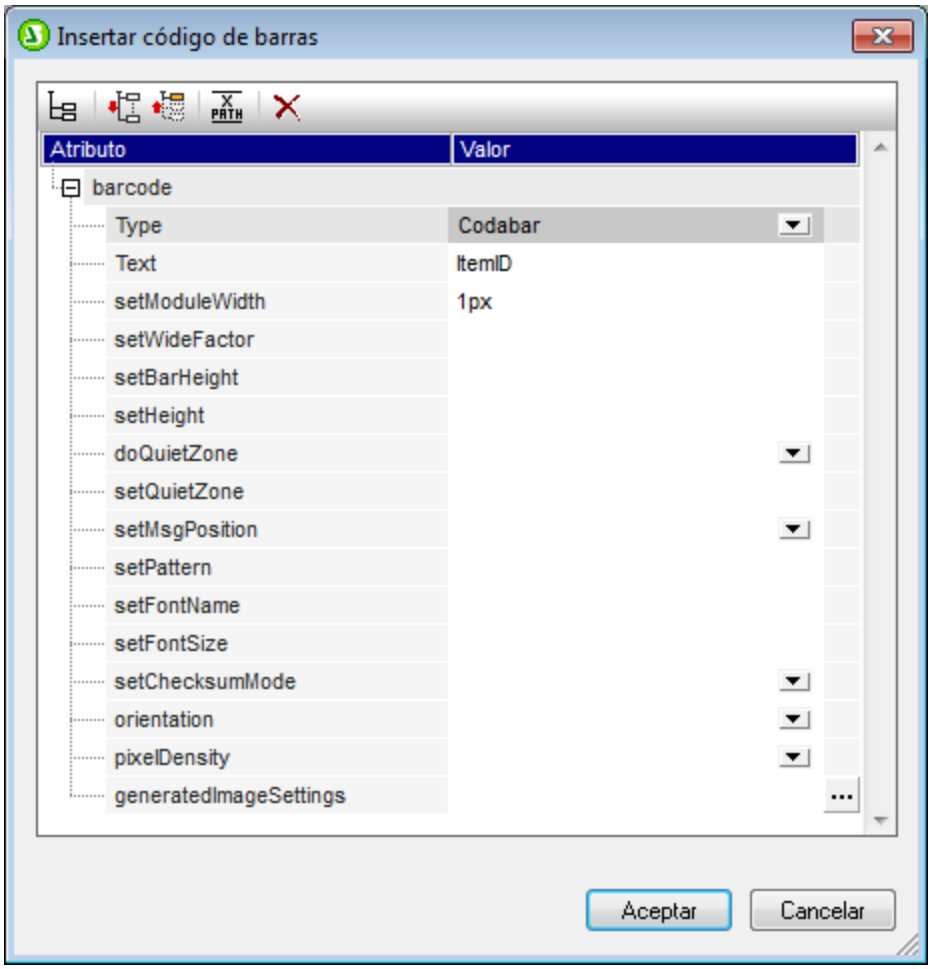
Importante: para generar códigos de barras en el resultado debe usar el procesador XSLT de Altova para generar ese resultado. La razón es que los códigos de barras archivo SPS se generan llamando a unas funciones de extensión Java especiales que no forman parte del estándar XSLT. Los procesadores XSLT de Altova son compatibles con estas funciones de extensión específicas, mientras que lo más probable es que otros procesadores XSLT no lo sean. El resultado es que los códigos de barras no se generan si se usa un procesador XSLT distinto al de Altova. El procesador XSLT de Altova está incluido en StyleVision; al generar resultados con el comando Generar del menú Archivo se llama automáticamente a este procesador integrado. También puede usar RaptorXML Server, que es el procesador XSLT independiente de Altova.

Insertar un código de barras

Siga estos pasos para insertar un código de barras en el diseño:

1. Elija la posición del diseño donde desea insertar el código de barras, haga clic con el botón derecho y seleccione el comando **Insertar código de barras**. Otra opción es seleccionar el comando de menú **Insertar | Insertar código de barras** o hacer clic en el icono **Código de barras** de la barra de herramientas y después hacer clic en la posición del diseño donde desea insertarlo. También puede arrastrar un nodo desde la Estructura del esquema hasta la vista **Diseño** y después seleccionar **Crear código de barras** en el menú contextual que aparece. Se cual sea el método de inserción elegido, aparece el cuadro de diálogo "Insertar código de barras" (*imagen siguiente*).

© 2016-2022 Altova GmbH



2. Hay dos propiedades que son obligatorias: `Type` y `Text`. Las demás son opcionales y pueden tener valores predeterminados. La propiedad `Type`, cuyo valor se puede seleccionar en una lista desplegable, especifica el tipo de código de barras (p. ej. EAN-13, que incluye códigos de barras ISBN, y UPC-A). La propiedad `Text` especifica el valor que generará el código de barras (p. ej. el número ISBN). Las demás propiedades de los códigos de barras se describen más abajo 202. Recuerde que si quiere usar un valor del archivo XML como valor de una propiedad, introduzca una expresión XPath 62 para buscar el nodo XML al que desea acceder. Esto se hace así: seleccione la propiedad, active el botón **XPath** de la barra de herramientas del cuadro de diálogo y después escriba la expresión XPath en el cuadro de diálogo que aparece 729. La expresión XPath se evaluará dentro del nodo de contexto actual.
3. Cuando termine de definir las propiedades haga clic en el botón **Aceptar**. La imagen del código de barras se inserta y el código de barras generado (*imagen siguiente*) se puede ver inmediatamente en las diferentes vistas previas de StyleVision.

Nota: las imágenes de códigos de barras se generan como archivos PNG.

Propiedades de los códigos de barras

A continuación enumeramos todas las propiedades de los códigos de barras. `Type` y `Text` son propiedades obligatorias y las demás son opcionales. Cada tipo de código de barras tiene propiedades diferentes.

- `Type`*:* el sistema de código de barras bajo el que se interpretará el mensaje, como EAN y UPC.
- `Text`*:* el valor que se usará para generar el patrón del código de barras.
- `SetModuleWidth`*:* el ancho de las barras del código.
- `SetBarHeight`*:* el alto de las barras.
- `SetHeight`*:* el alto del gráfico del código de barras.
- `DoQuietZone`*:* los valores `Yes` o `No` determinan si se implementan o no los márgenes del código de barras (que se especifican en las propiedades *SetQuietZone* y *SetVerticalQuietZone*).
- `SetQuietZone`*:* define los márgenes que rodean al código de barras. En los códigos de barras de una dimensión, el valor se aplica a la dimensión horizontal. En los códigos de barras 2D, el valor se aplica a las dimensiones horizontal y vertical. El valor de la dimensión vertical puede reemplazarse con el valor especificado en la propiedad *SetVerticalQuietZone*. La unidad de longitud debe ser milímetros (`mm`). Ejemplo: `2mm`.
- `SetVerticalQuietZone`*:* define los márgenes para la dimensión vertical de los códigos de barras 2D. La unidad de longitud debe ser milímetros (`mm`). Ejemplo: `2mm`.
- `SetMsgPosition`*:* especifica dónde aparece el texto del mensaje en relación al código de barras. Los valores son `top` (arriba), `bottom` (abajo) y `none` (ningún mensaje).
- `SetPattern`*:* especifica el patrón del mensaje de texto, para que sea legible. Más abajo puede ver una tabla con la sintaxis de los diferentes patrones.
- `SetFontName`*:* la fuente del texto.
- `SetFontSize`*:* el tamaño de la fuente del texto.
- `SetChecksumMode`*:* tiene estos valores: (i) `Add`*:* la suma de comprobación se añade automáticamente al mensaje; (ii) `Check`*:* la suma de comprobación se revisa mientras se genera el código de barras (se entiende que la suma de comprobación esté presente); (iii) `Ignore`*:* no se procesa la suma de comprobación; (iv) `Auto`*:* se habilita el comportamiento predeterminado del tipo de código de barras elegido.
- `Orientation`*:* indica la orientación del código de barras. Se puede girar de 90 en 90 grados en el sentido contrario a las agujas del reloj.
- `PixelDensity`*:* especifica la densidad de los píxeles de la imagen del código de barras. Cuanto mayor sea la densidad, más nítida será la imagen.
- `GeneratedImageSettings`*:* permite dar un nombre al archivo de imagen de código de barras generado. Si deja esta propiedad vacía, StyleVision genera un nombre de archivo automáticamente.

Sintaxis de los patrones

Los patrones sirven para que los mensajes de los códigos de barras sean más legibles. En el patrón, cada carácter del texto del mensaje se representa con un carácter de subrayado (_). Los demás caracteres del patrón se insertan en la posición correspondiente del texto del mensaje. La barra diagonal inversa (`\`) se usa como carácter de escape. Por tanto, `\?` insertaría el carácter '`?`' en el texto del mensaje (porque de lo contrario '`?`' es un carácter comodín). El carácter '`#`' se puede utilizar para eliminar un carácter del mensaje original. Aquí puede ver algunos ejemplos de patrones:

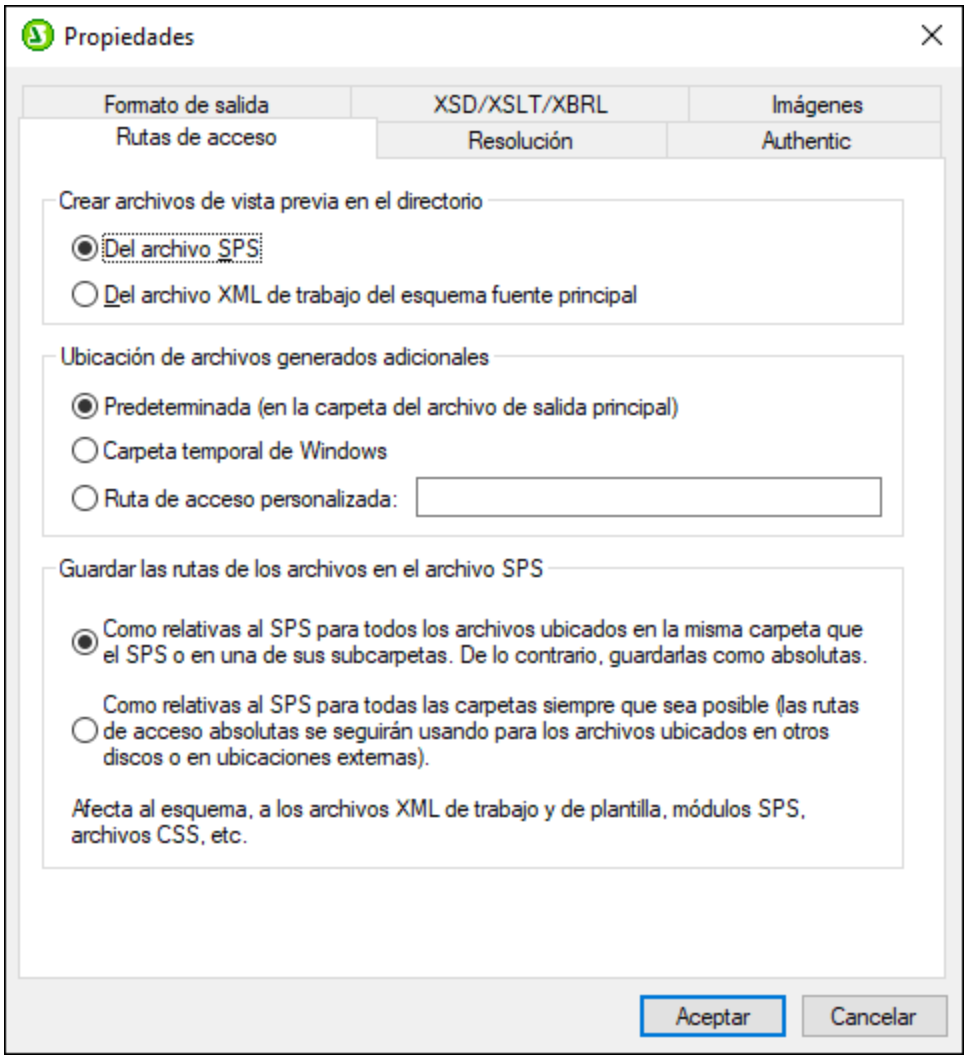
Texto de entrada del mensaje	Patrón	Texto de salida del mensaje
`123456`	`__ __ __`	`12 34 56`

© 2016-2022 Altova GmbH

15032011094655	________ __:__:__ UTC	15\03\2011 09:46:55 UTC
15-03-2011	__#/__#/____	15/03/2011

Generar archivos de salida

Los archivos de imagen de código de barras que se generan para los documentos de salida se guardan en el directorio especificado en la pestaña *Rutas de acceso* del cuadro de diálogo "Propiedades" (*imagen siguiente*), al que se accede con el comando de menú **Archivo | Propiedades**.



Los archivos de imagen de código de barras se pueden crear en el mismo directorio que el SPS o como archivo XML de trabajo. Estos archivos son temporales y se eliminan cuando se cierra el archivo SPS. Los archivos de imagen de código de barras que se crean cuando los documentos de salida se generan con el comando **Archivo | Guardar archivos generados** puede crearse en cualquier ubicación. La ubicación de destino se especifica en el grupo de opciones *Ubicación de archivos generados adicionales* (*imagen anterior*).

5.12 Módulos de diseño

Los módulos de diseño son objetos que contienen elementos de diseño. Estos módulos se insertan en el diseño SPS y aparecen en forma de bloque dentro del flujo del documento. Dentro de los módulos puede crear otros módulos como cuadros de diseño (cada uno de ellos con sus propios elementos de diseño). Por tanto, los módulos de diseño permite un proceso de diseño similar al de las aplicaciones de diseño gráfico tradicionales.

Estos son los pasos necesarios para crear un módulo de diseño:

1. Inserte un contenedor de diseño 204. Este contenedor puede ocupar todo el ancho de la página o tener otras dimensiones. Puede incluir una imagen modelo del diseño que sirva de guía o puede darle formato (en la ventana Estilos).
2. Inserte cuadros de diseño 208 en el contenedor de diseño. Los cuadros de diseño pueden incluir varios elementos de diseño (como texto estático, nodos de esquema, cálculos automáticos, imágenes, listas, etc.) y puede aplicarles formato (en la ventana Estilos). Los cuadros de diseño también pueden apilarse unos encima de otros.
3. También puede dibujar líneas 212, aplicarles formato, cambiar su posición y subirlas o bajarlas en la pila de objetos de diseño (cuadros de diseño y otras líneas).

Nota: los módulos de diseño no aparecen como tal en la salida en formato texto.

Diseños basados en formularios

Cuando crea un SPS nuevo 761, el usuario de StyleVision tiene la opción de crear un diseño de movimiento libre o un diseño basado en un formulario. Los diseños basados en formularios son diseños compuestos por un contenedor de diseño.

Nota: los módulos de diseño son compatibles con la vista Authentic de las ediciones Enterprise Edition de los productos de Altova.

5.12.1 Contenedores de diseño

Los contenedores de diseño tienen estas propiedades:

- Se pueden insertar 205 dentro del flujo de un documento, es decir, dentro de una plantilla. O se pueden usar como contenedor del diseño de documento.
- Pueden tener las dimensiones de la página de la sección (propiedad `ajustar automáticamente al tamaño de la página`) o puede tener las dimensiones que usted quiera. Para más información consulte el apartado Tamaño del contenedor de diseño 205.
- La posición de los objetos del contenedor de diseño se puede controlar con facilidad gracias a la cuadrícula de diseño 205 y a la función de zoom 206.
- Pueden tener propiedades de estilos 206 como contorno, colores de fondo, fuentes, etc.
- Pueden contener cuadros de diseño y líneas 207, pero ningún otro elemento de diseño. (Si quiere insertar otros elementos de diseño, debe hacerlo dentro de los cuadros de diseño.)
- Pueden contener imágenes modelo 207 que sirvan de referencia al diseñador de la hoja de estilos. Así podrá crear un diseño que coincida totalmente con la imagen modelo.

© 2016-2022 Altova GmbH

Nota: los contenedores de diseño son compatibles con la vista Authentic de las ediciones Enterprise Edition de los productos de Altova

Insertar un contenedor de diseño

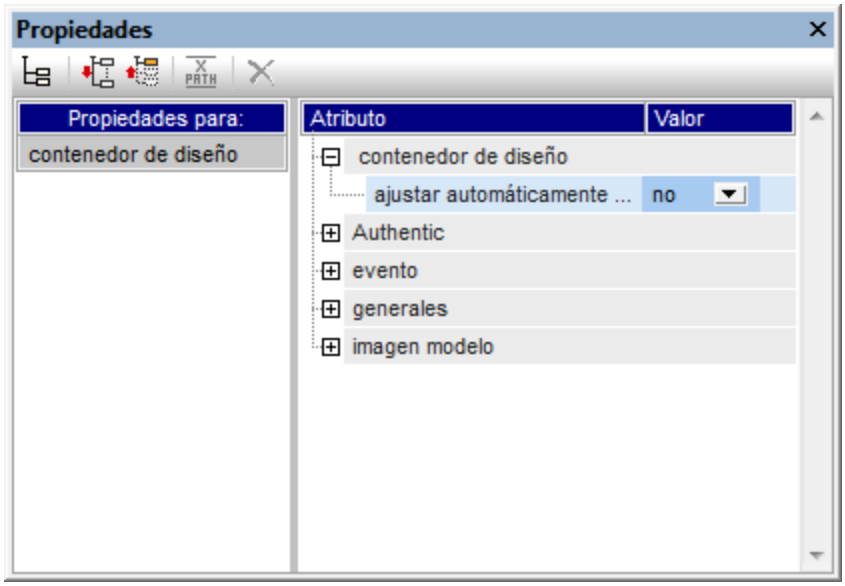
Para insertar un contenedor de diseño, haga clic en el icono **Insertar contenedor de diseño** de la barra de herramientas Insertar elementos de diseño 755 y después haga clic en la posición del diseño donde desea insertar el contenedor de diseño. Aparece un cuadro de diálogo preguntando si desea que el contenedor de diseño se ajuste automáticamente al tamaño de la página. Si hace clic en **Sí**, el contenedor de diseño tendrá el mismo tamaño que la página en dicha sección del documento (según las dimensiones definidas en las propiedades de la página). Si hace clic en **No**, se crea un contenedor de diseño con el tamaño predeterminado de 3.5 x 5.0 pulgadas.

Recuerde que también puede crear un contenedor de diseño nada más empezar a crear el diseño SPS.

Tamaño del contenedor de diseño

Hay dos conjuntos de propiedades que afectan al tamaño del contenedor de diseño:

- La propiedad `ajustar automáticamente al tamaño de la página` (ventana Propiedades). Si quiere que el contenedor tenga las mismas dimensiones que la página de la sección de documento, seleccione el valor `yes`. Por el contrario, si quiere que el contenedor tenga otras dimensiones, seleccione el valor `no`.



- Las propiedades `height` y `width` del grupo *detalles* de la ventana Estilos. Estas propiedades especifican las dimensiones del contenedor de diseño (cuando está seleccionado el valor `no` en la propiedad `ajustar automáticamente al tamaño de la página` de la ventana Propiedades).

Cuadrícula del contenedor de diseño

El contenedor de diseño cuenta con una cuadrícula que facilita la inserción de componentes dentro del contenedor. La cuadrícula incluye estas funciones de uso:

- *Mostrar/ocultar cuadrícula:* este comando de la barra de herramientas "Insertar elementos de diseño" activa/desactiva la cuadrícula del contenedor de diseño.
- *Tamaño de la cuadrícula:* en la pestaña *Diseño* del cuadro de diálogo "Opciones" (**Herramientas | Opciones** 894) puede especificar la longitud horizontal y vertical de la cuadrícula. No obstante, tenga en cuenta que, si selecciona unidades de longitud grandes, la cuadrícula podría dejar de verse.
- *Ajustar a la cuadrícula:* este comando de la barra de herramientas "Insertar elementos de diseño" habilita/deshabilita la función *Ajustar a la cuadrícula*. Cuando está habilitada, el borde superior e izquierdo de los cuadros de diseño y los extremos de las líneas se alinean con las líneas y los puntos de la cuadrícula respectivamente.

Función de zoom

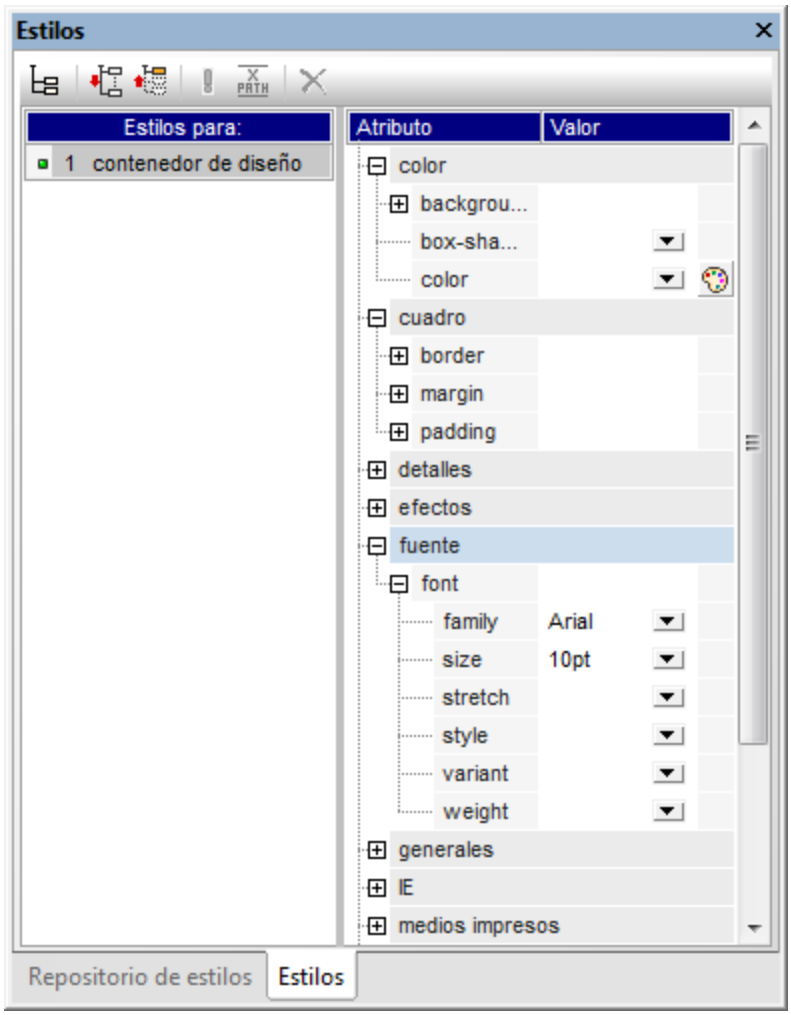
Si quiere, puede ampliar la vista del contenedor de diseño. Para ello cambie el factor de zoom en el cuadro combinado **Zoom** (de la barra de herramientas "Estándar") o mantenga pulsada la tecla **Ctrl** mientras mueve la rueda del ratón hacia adelante.

Propiedades de estilo de los contenedores de diseño

Hay dos tipos de propiedades de estilo que se pueden aplicar a los contenedores de diseño:

- Las que solamente afectan al contenedor de diseño y que no se heredan (p. ej. las propiedades `border` y `background-color`).
- Las que son heredadas por los cuadros de diseño que forman parte del contenedor de diseño (p. ej. las propiedades relacionadas con la fuente del texto).

© 2016-2022 Altova GmbH



Las propiedades de estilo de un contenedor de diseño se definen en la ventana Estilos (`imagen anterior`).

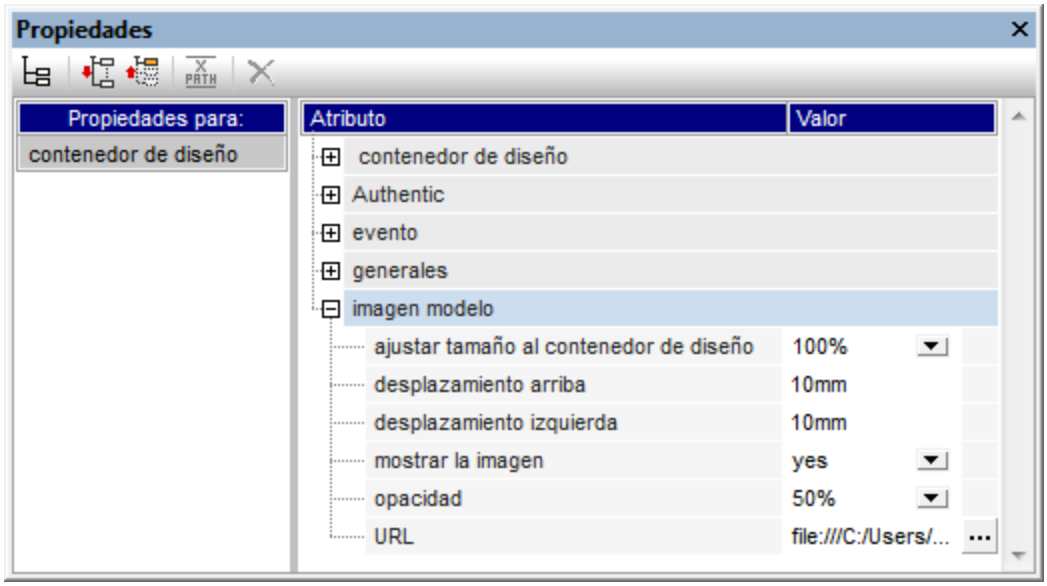
Contenido de los contenedores de diseño

Los únicos componentes de diseño que se pueden insertar dentro de un contenedor de diseño son líneas y cuadros de diseño. También puede insertar una imagen modelo (que no cuenta como elemento de diseño) para utilizarla como guía. Si quiere introducir elementos de diseño en el contenedor, inserte primer cuadros de diseño y en ellos podrá crear los demás elementos.

Imágenes modelo

En cada contenedor de diseño puede introducir una imagen modelo y utilizarla como guía mientras crea el SPS. La imagen modelo es un archivo de imagen que se puede ajustar totalmente al tamaño del contenedor de diseño. Si lo prefiere y si la imagen modelo es más pequeña que el contenedor de diseño, también puede desplazar la imagen a la posición deseada (vea más abajo el grupo de propiedades *imagen modelo*). Después puede crear el diseño encima de la imagen modelo, siguiendo su composición. De este modo podrá colocar elementos de diseño en la misma posición que en la imagen modelo. Recuerde que la imagen modelo solamente es visible en la vista **Diseño** y **no** aparece en la vista previa de los resultados: su único propósito es ayudarle a diseñar el SPS.

Puede controlar las propiedades de la imagen modelo desde la ventana Propiedades: seleccione el componente **`contenedor de diseño`**, expanda el grupo *imagen modelo* y modifique los valores de las propiedades (*imagen siguiente*).



También puede especificar la opacidad del contenedor de diseño para que la imagen modelo no oculte totalmente el diseño. O puede ocultar totalmente la imagen modelo con la propiedad `mostrar la imagen` (`yes/no`).

5.12.2 Cuadros de diseño

Todos los elementos de diseño (como texto estático, nodos del esquema, cálculos automáticos, imágenes, listas, etc.) deben introducirse dentro de **cuadros de diseño**. Los cuadros de diseño que contienen estos elementos de diseño se distribuyen por el contenedor de diseño. Recuerde que los elementos de diseño no se pueden insertar directamente en el contenedor de diseño. Deben estar en un cuadro de diseño.

En este apartado explicamos cómo:

- Insertar cuadros de diseño 208
- Seleccionar y mover cuadros de diseño 209
- Modificar el tamaño del cuadro de diseño 209
- Definir las propiedades de estilo del cuadro de diseño 209
- Insertar contenido en el cuadro de diseño 210
- Ordenar los cuadros de diseño en una pila de cuadros 211

Insertar un cuadro de diseño

Los cuadros de diseño solamente se pueden insertar dentro de contenedores de diseño 204. Para añadir un cuadro de diseño, haga clic en el icono **Insertar cuadro de diseño** de la barra de herramientas Insertar elementos de diseño 755 y después haga clic en la posición del contenedor donde desea insertarlo. Acto seguido aparece un cuadro de diseño sin contorno y con un texto predeterminado.

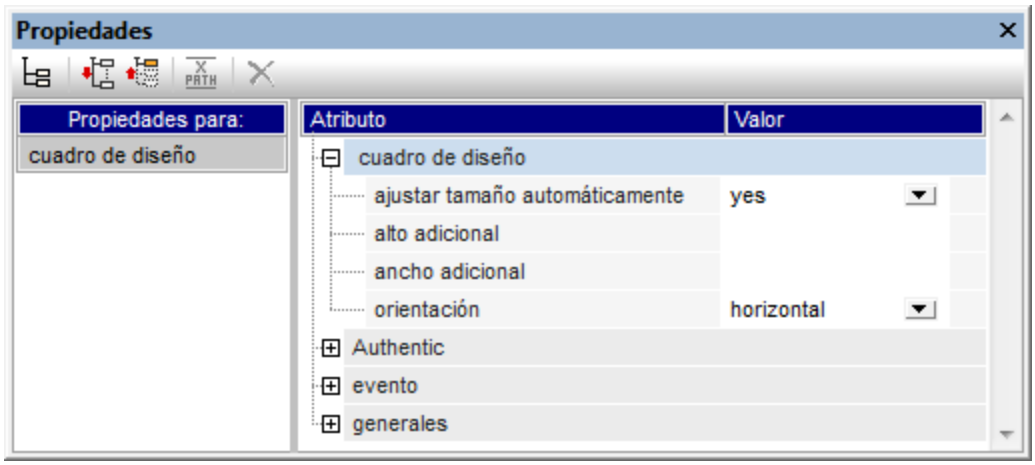
 © 2016-2022 Altova GmbH

Seleccionar y mover un cuadro de diseño

Para seleccionar un cuadro de diseño, pase el cursor sobre el borde izquierdo o superior del cuadro. Cuando el cursor se convierta en una doble flecha cruzada, haga clic con el ratón para seleccionar el cuadro de diseño. Mantenga pulsado el botón del ratón para arrastrar el cuadro de diseño a otra posición del contenedor. También puede mover el cuadro de diseño a la izquierda, derecha, arriba o abajo con las teclas de dirección. Cuando el cuadro de diseño está seleccionado, sus propiedades y estilos aparecen en las correspondientes ventanas laterales.

Tamaño del cuadro de diseño

Todos los cuadros de diseño tienen una propiedad llamada `ajustar tamaño automáticamente` (*imagen siguiente*). Si el valor de esta propiedad es `yes`, el tamaño del cuadro se ajusta automáticamente al contenido estático (incluido el marcado) que se inserte en el cuadro en la vista de diseño. Si el valor de la propiedad es `no`, el tamaño del cuadro de diseño no cambia al insertar contenido nuevo. Para cambiar a mano el tamaño del cuadro de diseño, arrastre su borde derecho e inferior. También puede cambiar el tamaño con las teclas de dirección: primero seleccione todo el cuadro de diseño 209 y después, (i) con la tecla **Mayús** pulsada, pulse la tecla de dirección izquierda o derecha para mover el borde derecho del cuadro de diseño o (ii), con la tecla **Mayús** pulsada, pulse la tecla de dirección arriba o abajo para mover el borde inferior del cuadro de diseño.

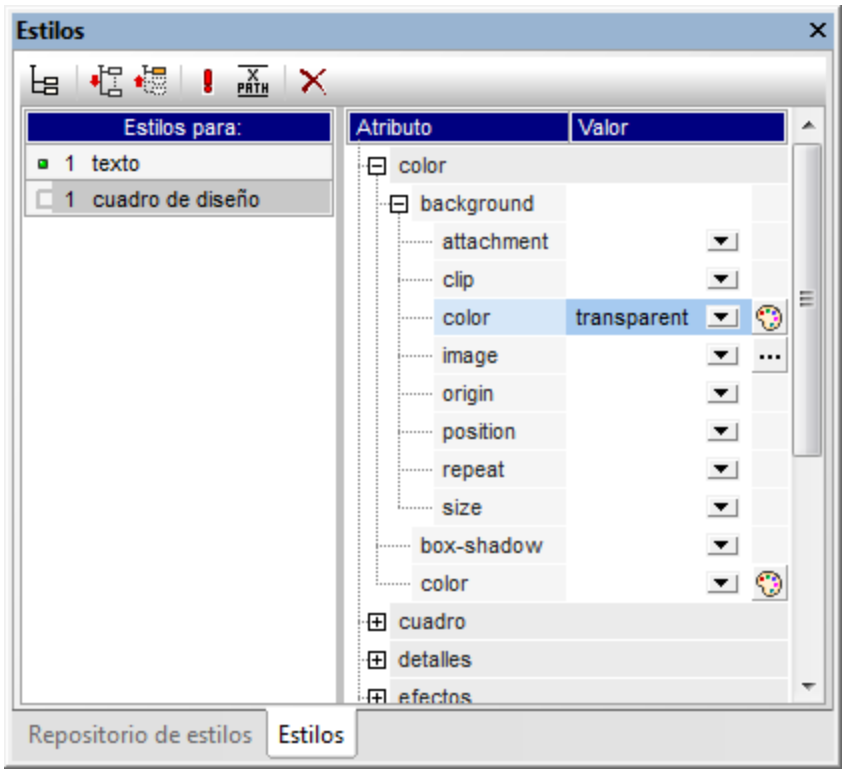


Las propiedades `alto adicional` y `ancho adicional` son las medidas que se obtienen al cambiar a mano el tamaño del cuadro de diseño (son medidas adicionales porque se suman a la dimensión óptima que se determina durante el ajuste automático del tamaño). Por tanto, puede cambiar el valor de estas propiedades para cambiar el tamaño del cuadro de diseño.

Nota: en los cuadros de diseño los saltos de línea se crean pulsando la tecla **Entrar**. Es importante recordarlo porque si añade contenido que no tiene un salto de línea, la longitud de la línea actual aumenta, aumentando también el ancho óptimo del cuadro de diseño (lo cual afecta también al valor de la propiedad `ancho adicional`, que se calcula en relación con el ancho óptimo).

Propiedades de estilo del cuadro de diseño

Las propiedades de estilo de un cuadro de diseño se definen en la ventana Estilos (componente **`cuadro de diseño`**) (*imagen siguiente*). Los estilos aparecen en esta ventana cuando se selecciona el cuadro de diseño.



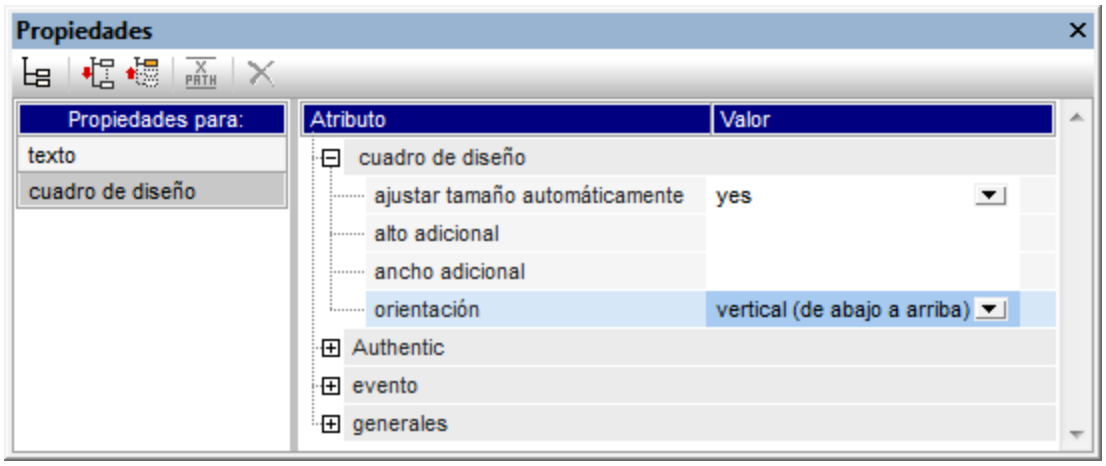
Nota: el valor `transparent` del estilo `background-colour` solamente se puede seleccionar en la lista desplegable del cuadro combinado (*imagen anterior*) y no se puede seleccionar en la paleta de colores como el resto de valores. Es importante recordar esto porque el valor `transparent` puede serle muy útil cuando trabaje con cuadros de diseño que forman parte de una pila de cuadros.

Insertar contenido en un cuadro de diseño

En un cuadro de diseño puede insertar cualquier tipo de elemento de diseño y los elementos de diseño se insertan en el cuadro igual que en el diseño SPS. Recuerde que en los cuadros de diseño no se pueden insertar ni contenedores 204 ni líneas 212. Debe tener en cuenta estos puntos:

- Si inserta elementos de diseño que necesitan un nodo de contexto, el nodo actual se toma como nodo de contexto. El nodo actual es el nodo en el que se creó el módulo de diseño.
- El contenido de texto de un cuadro de diseño se puede girar 90 grados en el sentido de las agujas del reloj y en el sentido contrario de las agujas del reloj. Así el texto se podrá leer en vertical de arriba a abajo o de abajo a arriba. Para ello, seleccione el texto que desea girar y en la ventana Propiedades (*imagen siguiente*), seleccione el componente **`cuadro de diseño`**. Después expanda el grupo de propiedades *cuadro de diseño* y seleccione un valor para la propiedad `orientación`.

 © 2016-2022 Altova GmbH

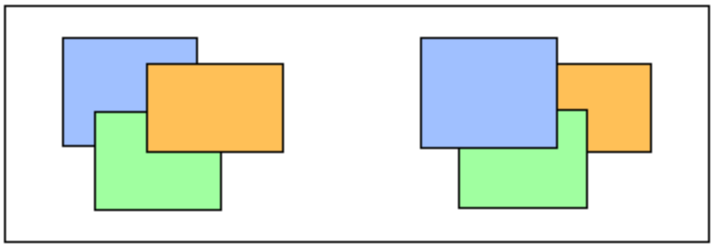


Es importante recordar que:

- La rotación se aplicará al documento de salida, pero no estará visible en la vista **Diseño** del SPS.
- Esta propiedad se puede aplicar a las celdas de tablas 165 que contienen texto.

Orden de apilamiento de los cuadros de diseño

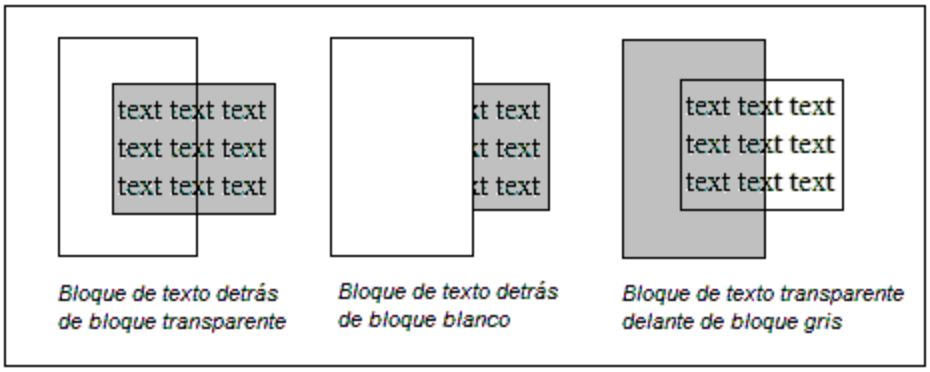
También puede colocar varios cuadros de diseño unos encima de otros. Cuando un cuadro de diseño opaco se coloca encima de otro, el cuadro opaco oculta al otro. Esto se puede extender naturalmente a todos los cuadros de una pila de cuadros. En casos así, el cuadro de diseño situado en la parte superior de la pila debe estar totalmente visible y los demás están parcialmente o totalmente cubiertos.



Los cuadros de diseño se pueden traer hacia adelante o enviar hacia detrás con ayuda de los comandos del menú **Orden** (del menú contextual del cuadro de diseño). Con estos comandos puede cambiar el orden de los cuadros de diseño: (i) en relación a los demás cuadros (con los comandos **Traer hacia adelante** y **Enviar hacia detrás**) o (ii) en relación a toda la pila (con los comandos **Traer al frente** y **Enviar al fondo**). En las imágenes anteriores, por ejemplo:

- *Pila de la izquierda:* el orden es naranja, verde y azul
- *Pila de la derecha:* el orden es azul, verde y naranja

No olvide que si los cuadros de diseño con fondo transparente (el color de fondo predeterminado) no tienen bordes y hay otros cuadros de diseño transparentes en la pila, puede parecer que los cuadros no se mueven debido a su transparencia. Por ejemplo, pueden darse estos casos:



Nota: a la pila de cuadros de diseño también puede añadir líneas [212] y cada línea se puede mover en relación a los demás componentes de la pila.

5.12.3 Líneas

Las líneas se pueden insertar dentro de los contenedores de diseño [212] (pero no en los cuadros de diseño). Después puede seleccionar la línea, cambiar su tamaño y moverla [212] por el contenedor de diseño, asignarle propiedades [213] y moverlas hacia adelante y hacia detrás en una pila de componentes de diseño [213] compuesta por cuadros de diseño y otras líneas.

Insertar una línea

Para añadir una línea a un contenedor de diseño:

1. Haga clic en el icono **Insertar línea** de la barra de herramientas **Insertar elementos de diseño** [755].
2. A continuación haga clic en la posición del contenedor de diseño donde desea que comience la línea.
3. Arrastrando el puntero y sin soltar el botón del ratón, dibuje la línea hasta el punto donde desea que finalice. Después suelte el botón del ratón.

Como resultado se inserta una línea de color negro. En cada extremo de la línea aparece un punto negro que resalta el inicio y el final de la línea.

Seleccionar la línea, moverla y cambiar su tamaño

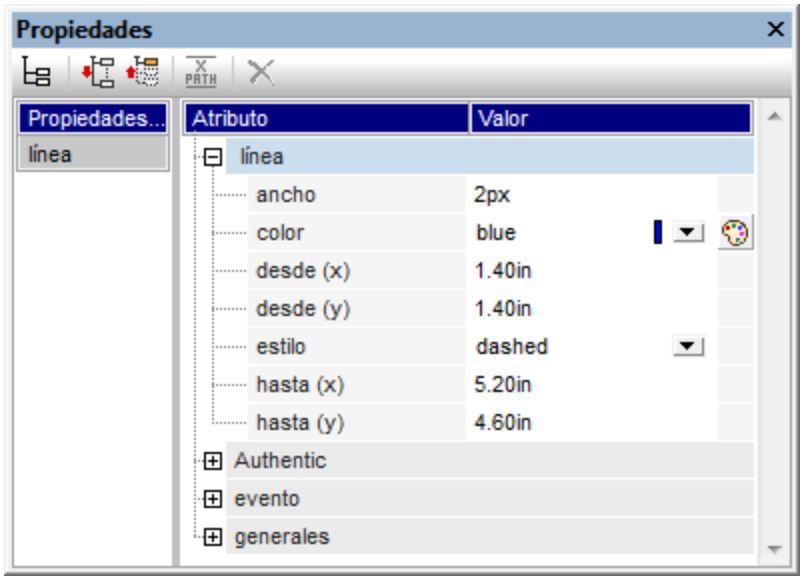
Puede realizar todas estas operaciones de arrastrar y colocar:

- Para seleccionar una línea, haga clic en cualquier parte de ella (el cursor se convierte en una doble flecha cruzada cuando está encima de la línea). Una vez seleccionada, sus propiedades aparecen en la ventana Propiedades, donde se pueden editar (*imagen siguiente*).
- Para mover una línea, selecciónela y arrástrela hasta su nueva posición. También puede mover la línea con las flechas de dirección.
- Para cambiar el tamaño o la orientación de la línea, seleccione uno de sus extremos y cambie su posición hasta obtener el tamaño o la orientación deseados. También puede cambiar el tamaño o la orientación de la línea pulsando la tecla **Mayús** y las teclas de dirección (las teclas izquierda/derecha mueven el extremo derecho de la línea hacia la izquierda y la derecha, mientras que las teclas arriba/abajo mueven el extremo derecho hacia arriba y hacia abajo).

 © 2016-2022 Altova GmbH

Propiedades de las líneas

Cuando está seleccionada una línea sus propiedades aparecen en la ventana Propiedades (imagen siguiente), donde se pueden editar. También puede hacer clic con el botón derecho en una línea y seleccionar el comando Editar propiedades de línea, lo cual abre un cuadro de diálogo con las propiedades de la línea.

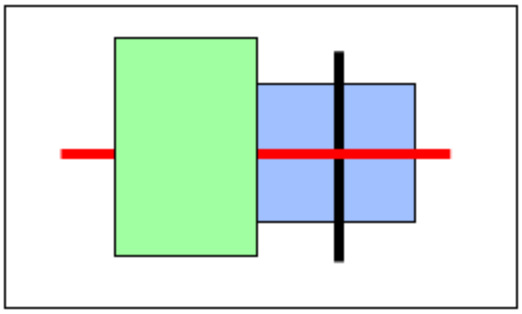


Estas son las propiedades que puede editar:

- *color:* especifica el color de la línea. Negro es el color predeterminado.
- *tamaño y posición:* la ubicación de los extremos de la línea se determina mediante un sistema de coordenadas cartesianas (x-y = horizontal-vertical). El marco de referencia es la esquina superior izquierda del contenedor de diseño, cuyas coordenadas serían `(x=0, y=0)`.
- *ancho:* especifica el grosor de la línea.

Líneas en pilas de componentes de diseño

Cuando una línea está en una pila compuesta por cuadros de diseño y otras líneas, puede enviarla hacia adelante o hacia detrás con los comandos del menú **Orden** (del menú contextual que aparece al hacer clic con el botón derecho en la línea). Con estos comandos puede cambiar el orden de las líneas: (i) en relación a los demás componentes de la pila (con los comandos **Traer hacia adelante** y **Enviar hacia detrás**) o (ii) en relación a toda la pila (con los comandos **Traer al frente** y **Enviar al fondo**).



En la imagen anterior, por ejemplo, el orden de los componentes de la pila es: cuadro verde, línea roja, línea negra, cuadro azul.

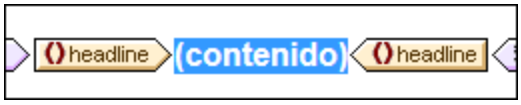
© 2016-2022 Altova GmbH

5.13 Función Convertir en

La función **Convertir en** está disponible cuando está seleccionada una plantilla o el contenido de la plantilla y sirve para cambiar: (i) el nodo al que se aplica la plantilla o (ii) qué tipo de contenido se crea a partir del nodo en el diseño.

¿Qué se puede cambiar con la función Convertir en?

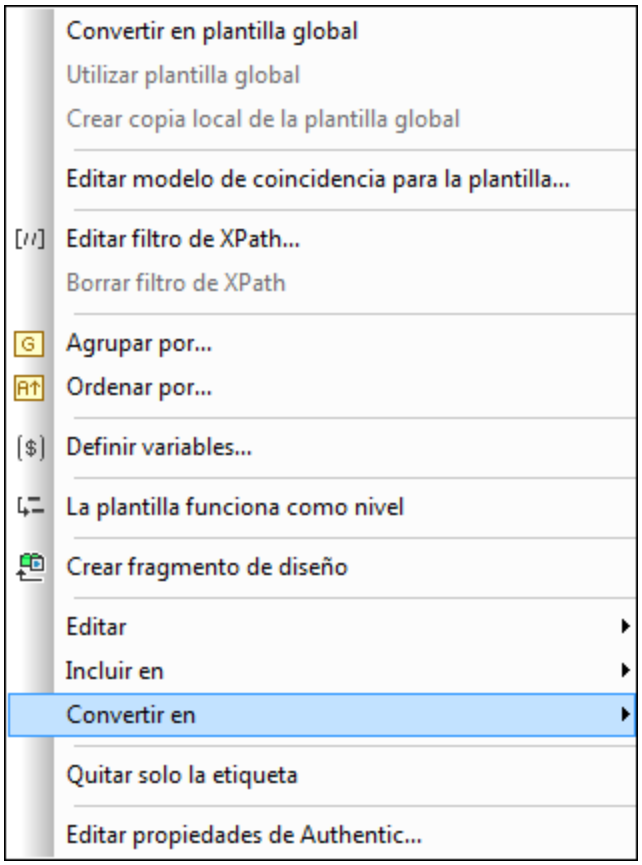
Se puede cambiar tanto el nodo como su contenido. En la imagen siguiente de la izquierda, se seleccionó el nodo. En la imagen siguiente de la derecha, se seleccionó el contenido del nodo.



El elemento `Headline` de la imagen anterior se usó para crear `(contenido)` en el diseño, lo cual se representa con el marcador de posición `(contenido)`. El nodo también se podría haber insertado en forma de otro tipo de contenido, como por ejemplo un campo de entrada o un cuadro combinado. También puede seleccionar otros tipos de contenido.

El comando Convertir en

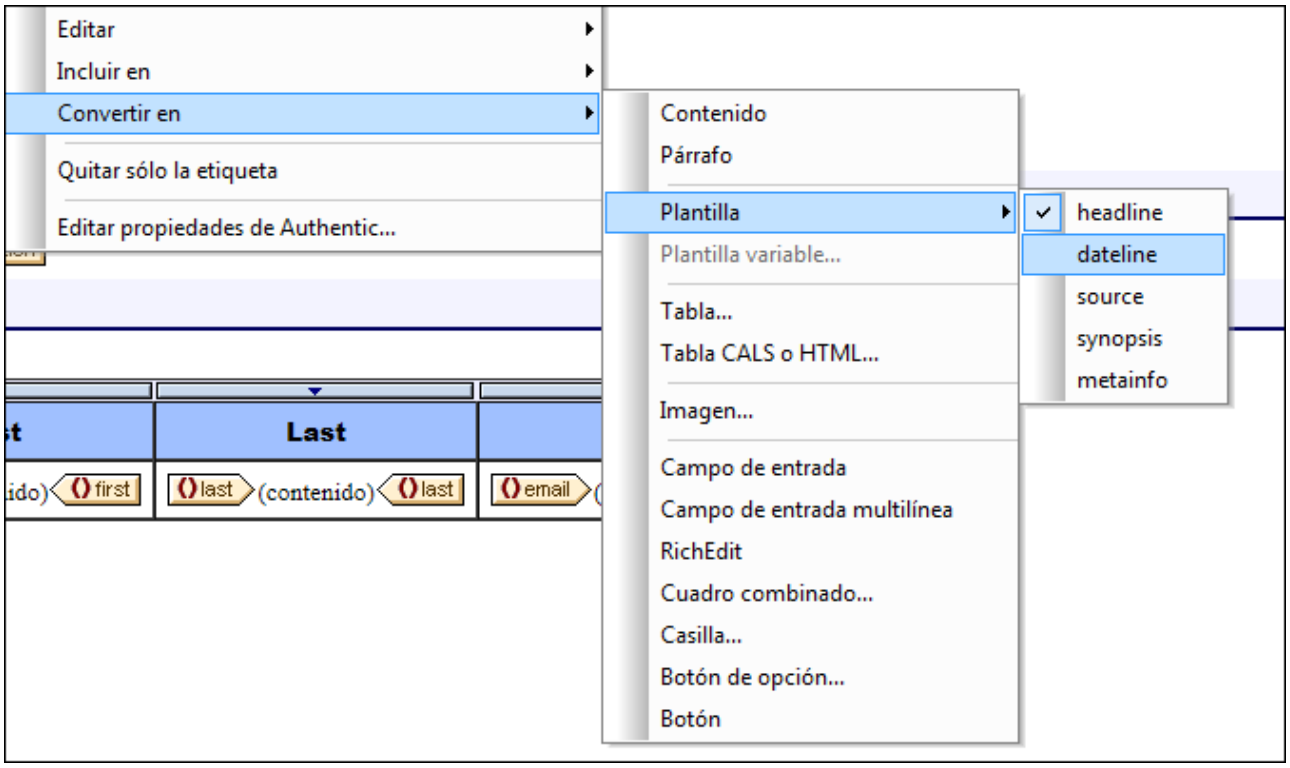
Al hacer clic con el botón derecho en un nodo, aparece un menú contextual que incluye el comando **Convertir en** (*imagen siguiente*).



Cambiar los modelos de coincidencia de la plantilla

Si hace clic con el botón derecho en una plantilla, puede usar el comando **Convertir en** para cambiar el nodo al que se aplica la plantilla. Por ejemplo, imagine que el nombre de elemento cambió en el esquema. Para cambiar el nodo al que se aplica la plantilla, pase el ratón por encima del comando **Convertir en**. En el submenú que aparece seleccione la opción **Plantilla** y después, en el nuevo submenú, seleccione el nodo al que desea aplicar la plantilla.

 © 2016-2022 Altova GmbH

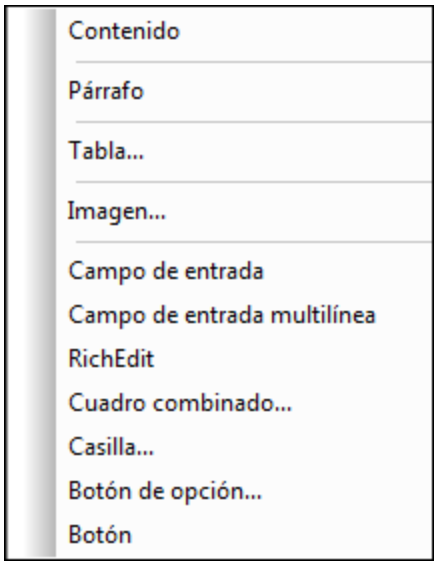


Si el modelo de contenido del nodo seleccionado no coincide con el modelo descrito en la plantilla, habrá incoherencias estructurales. Estas incoherencias se consideran errores y se señalan tachando en rojo las etiquetas de los nodos que no son válidos.

También puede cambiar el modelo de coincidencia de la plantilla por una plantilla variable 273 en lugar de un nodo.

Cambiar el tipo de contenido del nodo

Si hace clic con el botón derecho en una plantilla o en su contenido, también puede usar el comando **Convertir en** para cambiar el tipo de contenido que se creó a partir del nodo. Pase el ratón por el comando **Convertir en** y en el submenú que aparece seleccione el nuevo tipo de contenido (*imagen siguiente*).



© 2016-2022 Altova GmbH

6 Archivos SPS: estructura

La estructura de un documento SPS está determinada tanto por sus datos de entrada como por el diseño de los documentos de salida. La estructura del documento SPS se controla mediante:

- Esquemas fuente 221
- Archivos SPS modulares 251
- Plantillas y fragmentos de diseño 265

Estructura según los datos de entrada: esquemas fuente y archivos SPS modulares

Cuando decimos *estructura según los datos de entrada* nos referimos al hecho de que los esquemas fuente de los archivos SPS definen la estructura de los documentos de entrada y en esta estructura se basa el documento SPS. Por ejemplo, si un esquema fuente especifica una estructura basada en una secuencia de elementos `Office`, el diseño SPS tendrá una plantilla para el elemento `Office`. Durante la fase de procesamiento, esta plantilla se aplicará a cada elemento `Office` del documento de datos de origen.

Otro ejemplo de cómo la estructura del esquema fuente define el diseño del archivo SPS es el uso de tablas. Imagine un elemento `Office` con varios elementos secundarios `Person` y que cada elemento `Person` contiene un conjunto de elementos secundarios (como `Name`, `Address`, `Telephone`, etc.). En este caso podríamos crear una plantilla en forma de tabla para el elemento `Person`. Cada elemento `Person` se puede presentar en una fila distinta de la tabla (*imagen siguiente*) y cada columna de la tabla aloja los datos del elemento `Person` (es decir, los elementos secundarios de `Person`).

First	Last	Title (sorted by)
Loby	Matise	Accounting Manager
Frank	Further	Accounts Receivable
Vernon	**Callaby**	Office Manager

Tal plantilla solo es posible gracias a la estructura del elemento `Office` y a que los elementos `Person` son todos elementos del mismo nivel. En la plantilla de tabla, se diseña una fila para el elemento `Person` y este procesamiento (el diseño de la fila) se aplica a su vez a cada elemento `Person` del documento de origen, creando una fila nueva por cada elemento `Person` y una columna por cada uno de sus elementos secundarios.

En la sección Esquemas fuente 221 encontrará información sobre cómo usar los diferentes tipos de esquemas fuente.

Además, en StyleVision puede reutilizar sus archivos SPS como módulos en otros archivos SPS. Así podrá incluir módulos dentro de una estructura y modificarla. No obstante, las estructuras de esquema que forman parte de un módulo deben ajustarse a la estructura del esquema subyacente del SPS que contiene el módulo. Para más información sobre cómo trabajar con archivos SPS modulares consulte la sección Archivos SPS modulares 251.

Estructura según el diseño de los documentos de salida: plantillas y fragmentos de diseño

Mientras que el esquema fuente define la estructura del documento de datos de entrada, el diseño del documento de salida se especifica en el archivo SPS propiamente dicho. Este diseño se guarda en una

plantilla de documento llamada ***plantilla principal***. La plantilla principal suele contener varias plantillas más y puede hacer referencia a plantillas globales. Para más información consulte la sección Plantillas y fragmentos de diseño 265.

Esta flexibilidad a la hora de crear composiciones de diseño (usando varias plantillas) también se debe a una característica de StyleVision llamada ***fragmentos de diseño***. Estos componentes permiten asignar un procesamiento concreto a un fragmento del documento y reutilizarlo. Los fragmentos de diseño se diferencian de las plantillas globales en que: (i) un fragmento de diseño puede estar formado por varias plantillas y (ii) el mismo contenido puede tener asignado un procesamiento distinto en fragmentos de diseño distintos (y el usuario puede usar uno de los fragmentos en una plantilla, dependiendo de lo que necesite en cada momento). Por ejemplo, en algunos casos necesitará usar el nodo `Email` como hipervínculo que abre un correo electrónico vacío y en otros como texto en negrita y color rojo. En este caso puede crear dos fragmentos de diseño diferentes, uno para cada situación, y utilizarlos cuando los necesite.

Para más información consulte la sección Fragmentos de diseño 278.

 © 2016-2022 Altova GmbH

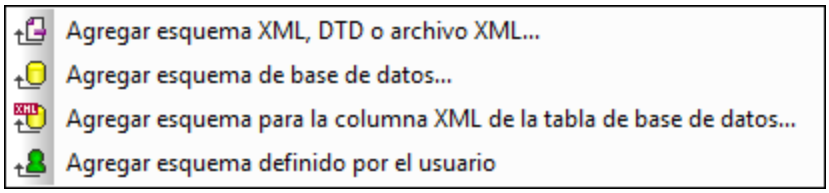
6.1 Esquemas fuente

Los esquemas fuente son la piedra angular y el punto de partida del diseño SPS. La estructura del diseño SPS está determinada por: (i) las decisiones que toma el usuario durante la selección del esquema y (ii) los elementos raíz seleccionados en el esquema.

Selección de esquemas

Hay varias maneras de seleccionar el esquema para el archivo SPS:

1. Haga clic en **Archivo | Nuevo** y seleccione directamente el tipo de esquema fuente que desea añadir (todas las opciones, excepto **Archivo nuevo (vacío)**).
2. Haga clic en **Archivo | Nuevo** y seleccione la opción **Archivo nuevo (vacío)**. En el diálogo que aparece seleccione **Crear un documento con movimiento libre** y pulse **Aceptar**. Se crea un archivo SPS nuevo que aparece en la vista **Diseño**. Después, en la ventana Vista general del diseño 45, haga clic en `Agregar fuente nueva...`. Aparece un menú contextual con los métodos que puede usar para añadir esquemas al diseño (*imagen siguiente*). Cada comando de este menú se describe en los apartados de esta sección.



El esquema fuente seleccionado puede ser un archivo, una BD, o o un esquema fuente definido por el usuario. Llegados a este punto es recomendable pensar si utilizará plantillas globales y si los elementos que desea crear como plantillas globales están definidos como elementos globales en el esquema. Cuando añada un esquema fuente a partir de un archivo DTD, recuerde que todos los elementos definidos en la DTD son elementos globales. Cuando añada un esquema a partir de un esquema XML, compruebe qué elementos están definidos como elementos globales y, si quiere realizar cambios en el esquema, compruebe si puede hacerlo en su entorno XML. Cuando añada un esquema a partir de una BD, podrá seleccionar qué tablas de la BD se importan. Esta selección determina la estructura del esquema XML que se generará a partir de la BD.

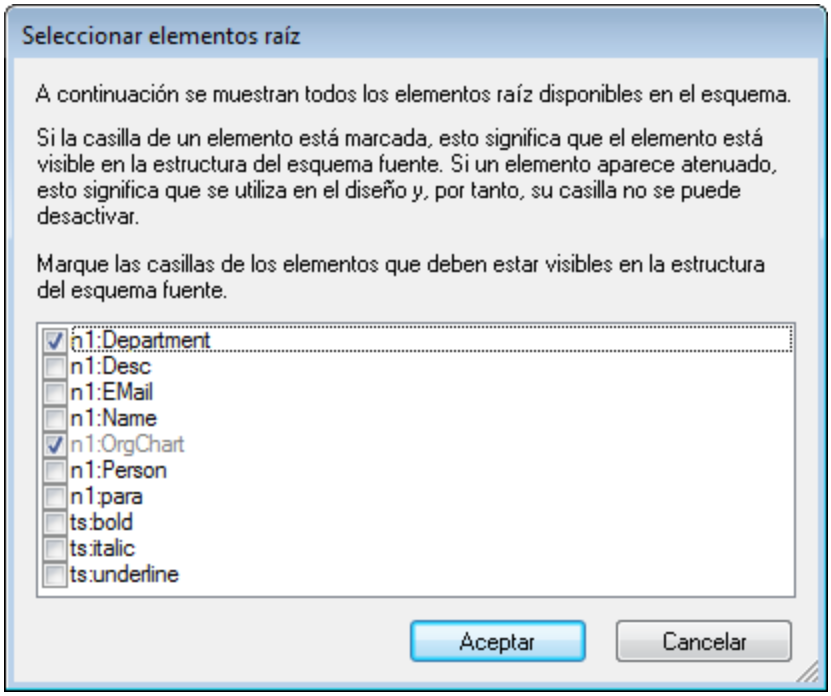
Nota: si quiere añadir un espacio de nombres al SPS o a la hoja de estilos XSLT que se genera a partir del SPS, el espacio de nombres debe añadirse al elemento de nivel superior `schema` del esquema XML en el que se basa el SPS.

Elementos raíz

Si un esquema fuente tiene varios elementos globales 31, entonces puede seleccionar varios elementos raíz (elementos de documento 30) para usarlos en el diseño. Es decir, el diseño SPS puede tener plantillas que coincidan con varios elementos de documento. La ventaja es que si un SPS, como `UniversalSPS.sps`, por ejemplo, basado en el esquema `UniversalSchema.xsd`, tiene una plantilla para cada uno de sus dos elementos raíz (`Element-A` y `Element-B`), entonces el SPS se puede usar con un documento XML de instancia que tenga `Element-A` como elemento de documento pero también con un XML que tenga `Element-B` como elemento de documento. Por cada XML de instancia, se utiliza solo la correspondiente plantilla. Esto se debe a que en el SPS solamente hay una plantilla que coincida con cada elemento de documento del archivo XML de instancia. Por ejemplo, el elemento de documento `/Element-A` coincide con la plantilla que

selecciona `/Element-A` pero no coincide con la plantilla que selecciona `/Element-B`. En este sentido es importante recordar que si se definieron varios elementos globales en el esquema, entonces es válido cualquier documento XML que tenga como elemento de documento uno de esos elementos globales (siempre y cuando su subestructura sea válida según el esquema, por supuesto).

Si quiere poder usar varios elementos raíz (elementos de documento [30]) con su SPS, haga clic en el botón ... situado a la derecha de la entrada `/Elementos raíz` (ventana Estructura del esquema). Al hacer clic en este botón se abre este cuadro de diálogo:



El cuadro de diálogo enumera todos los elementos globales del esquema. Seleccione los elementos globales que desea usar como elementos raíz (como elementos de documento [30]) y haga clic en **Aceptar**. Los elementos seleccionados estarán disponibles como elementos raíz del documento y aparecerán en la lista `/Elementos raíz`. Ahora puede crear una plantilla para cada uno de estos elementos de documento. Cada una de estas plantillas sirve de plantilla de elemento raíz alternativa. Cuando se procese un documento XML con este SPS, solamente se usará una de estas plantillas de elemento raíz: la que coincida con el elemento (documento) raíz del documento XML.

Así que, cuando con el SPS se procese un documento XML con elemento de documento `Element-A`, se dispara la plantilla raíz del SPS que coincide con `Element-A`, mientras que las demás plantillas de elemento raíz del SPS se ignoran. Si se procesa un documento XML con elemento de documento `Element-B`, se dispara la plantilla raíz del SPS que coincide con `Element-B`, mientras que las demás plantillas de elemento raíz del SPS se ignoran. De este modo, solo hace falta un SPS para procesar dos o más documentos XML, aunque tengan elementos raíz (de documento) diferentes.

Temas relacionados

- Estructura del esquema [48]

© 2016-2022 Altova GmbH

6.1.1 Esquemas XML y DTD

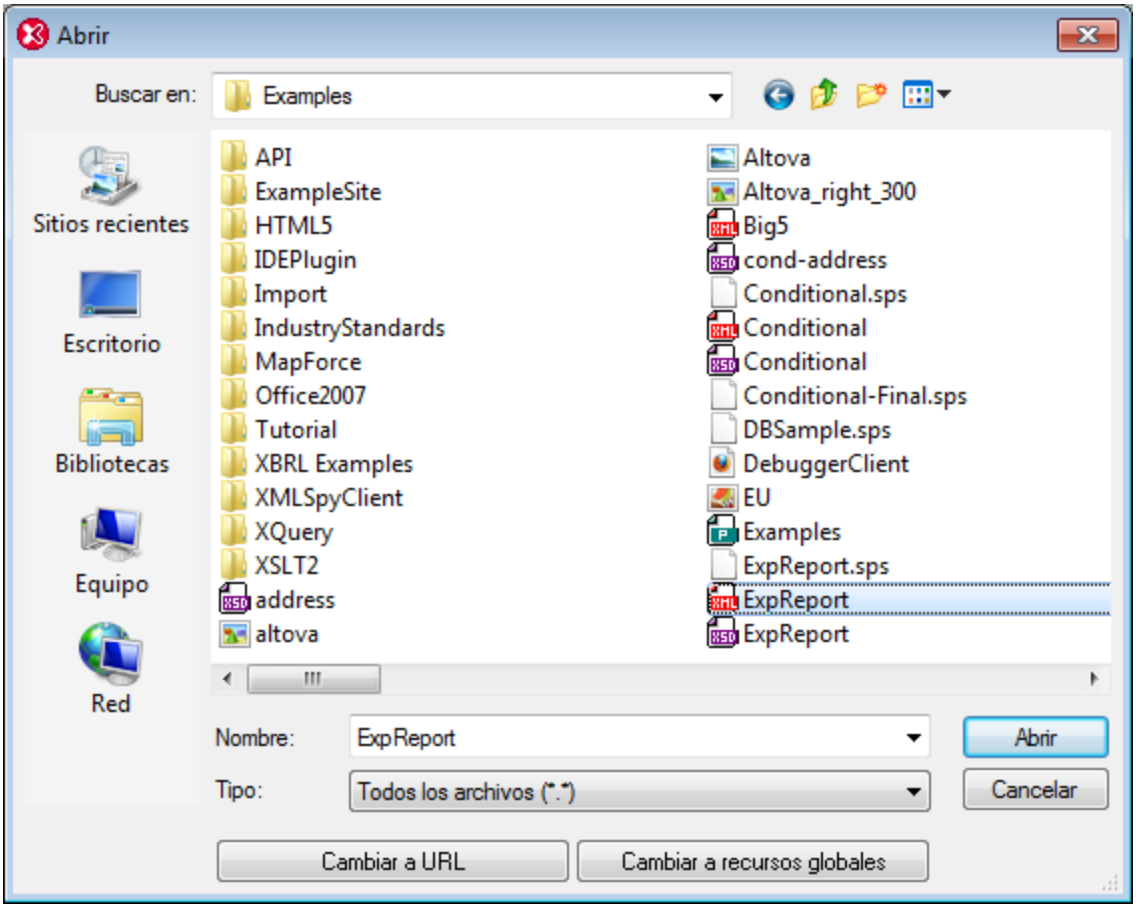
Un diseño SPS puede estar basado en un esquema XML o en una DTD. Hay dos maneras de crear un esquema fuente a partir de un archivo XSD o DTD:

- El esquema fuente se crea a partir del esquema XML o de la DTD nada más crear el archivo SPS (con el comando de menú **Archivo | Nuevo | Nuevo a partir de un esquema XML, DTD o archivo XML**).
- El esquema XML o la DTD se añade a un SPS vacío (desde la ventana Vista general del diseño 45).

Si el esquema XML o el archivo DTD seleccionado es válido, se crea una entrada de esquema fuente bajo el nodo `Fuentes` de la ventana Estructura del esquema. También puede crear un esquema fuente a partir de un archivo XML. Si este archivo XML tiene un esquema XML (.xsd) o un archivo DTD asociados, éste se carga como esquema fuente y el archivo XML se carga como archivo XML de trabajo. Si el archivo XML no tiene asociado ningún esquema, aparece un cuadro de diálogo preguntando si desea generar un esquema XML basado en la estructura y el contenido del archivo XML o si prefiere buscar un esquema ya existente. Si se decanta por la primera opción, el esquema generado se carga como esquema fuente y el archivo XML se carga como archivo XML de trabajo.

▼ Seleccionar y guardar archivos a través de direcciones URL y recursos globales

En varios cuadros de diálogo "Abrir archivo" y "Guardar archivo" de StyleVision 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.

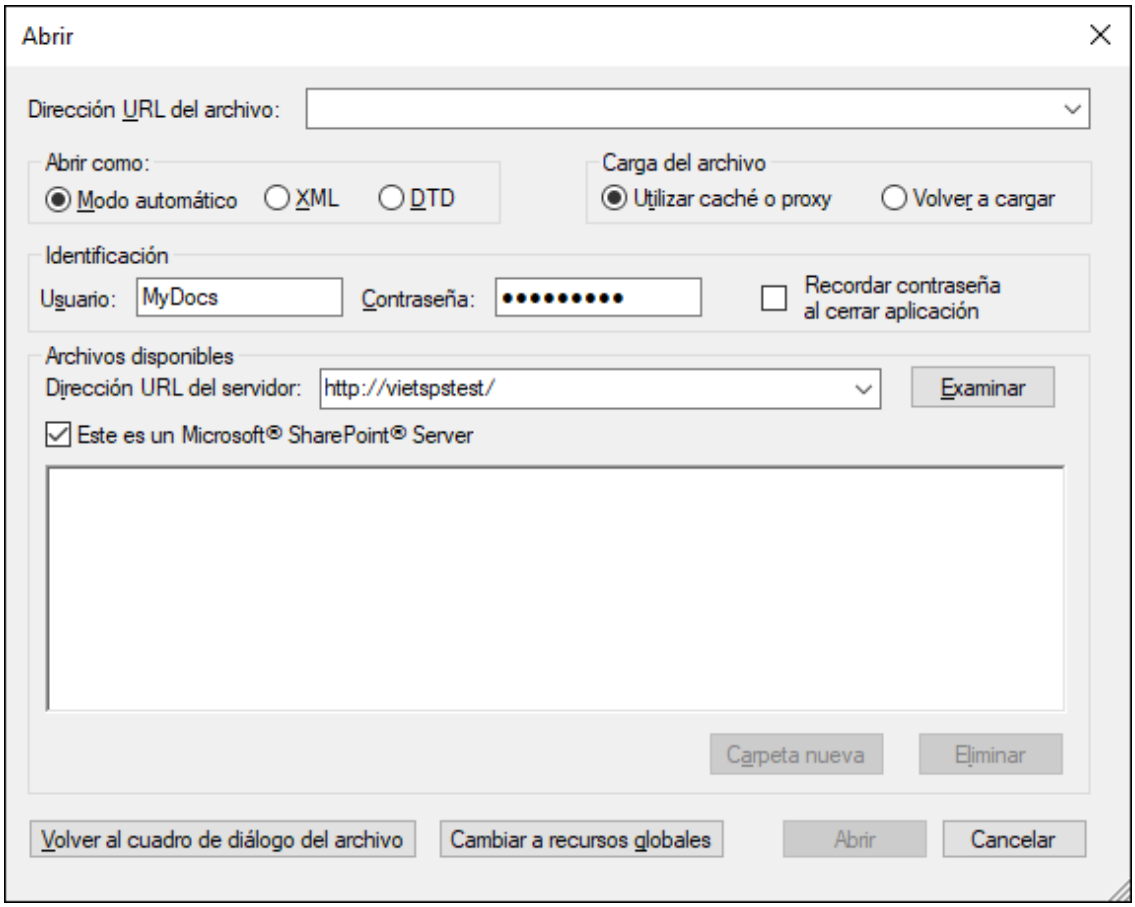


Seleccionar archivos a través de direcciones URL

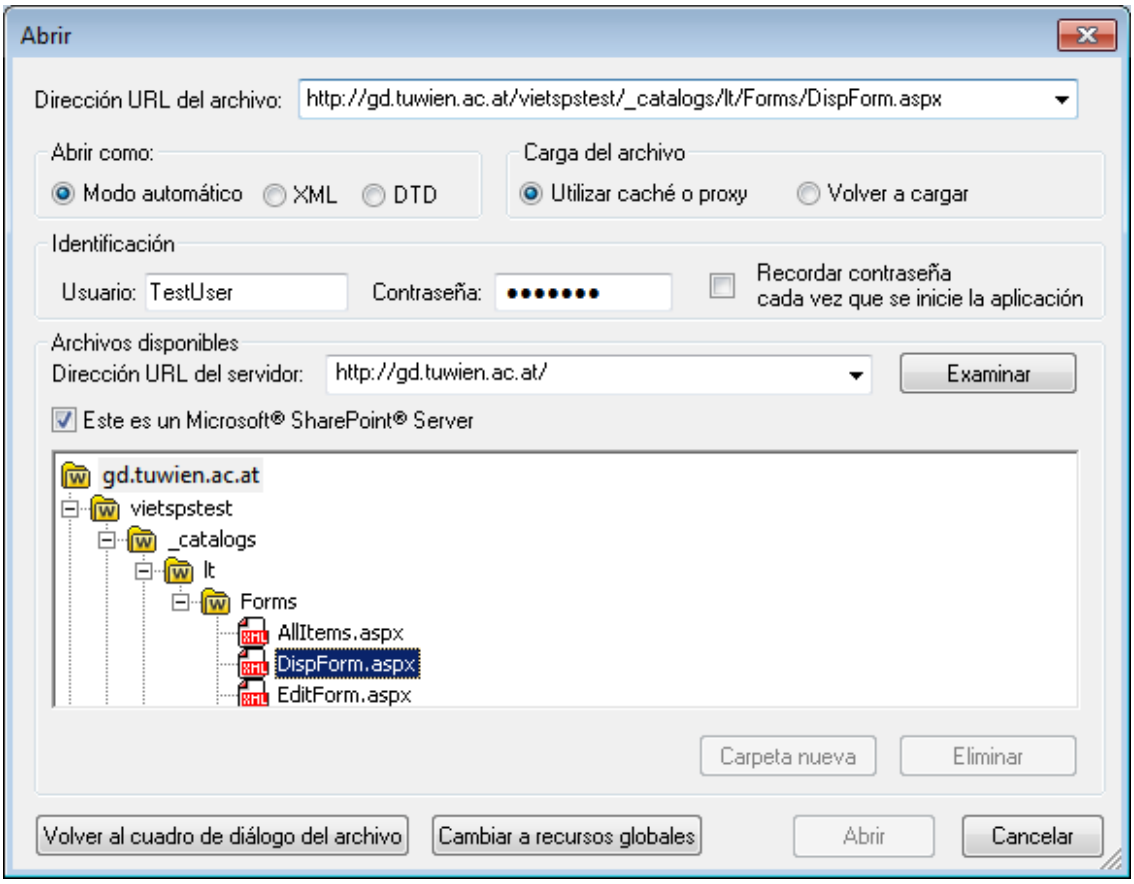
Siga estas instrucciones para seleccionar un archivo a través de una dirección URL (ya sea para abrirlo o para guardarlo):

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:

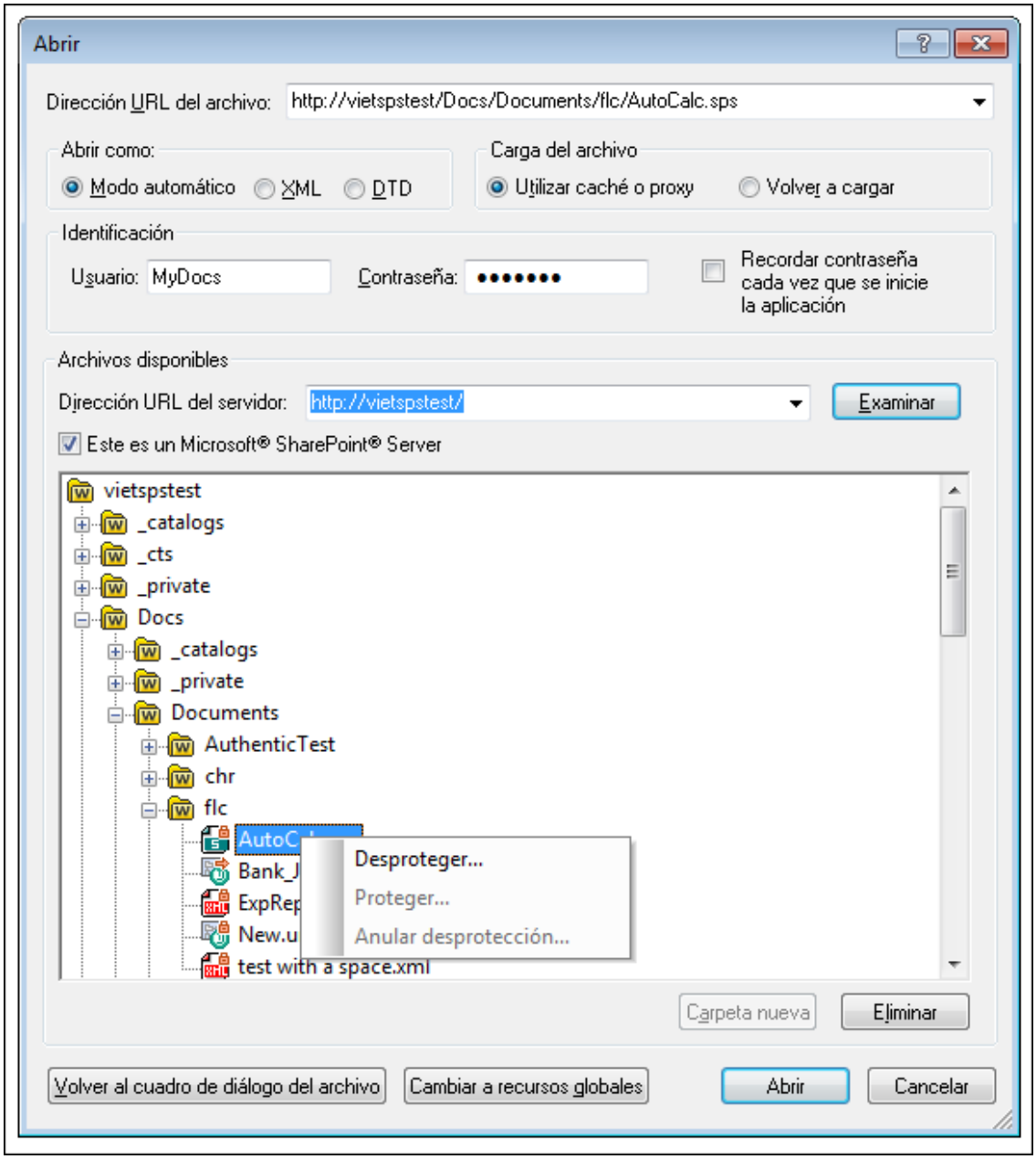
- La función **Examinar** solamente está disponible en servidores compatibles con WebDAV y servidores Microsoft SharePoint. Los protocolos compatibles son FTP, HTTP y HTTPS.
- Si desea un mayor control durante el proceso de carga del archivo, StyleVision ofrece la opción de cargar el archivo desde la 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 456. Para obtener información sobre esta característica consulte la sección Recursos globales 444 de este manual.

El tipo de datos `anyType` de XML Schema

Si un elemento del esquema XML tiene asignado el tipo de datos `anyType` de XML Schema o no tiene asignado ningún tipo de datos, el árbol del esquema de la ventana Estructura del esquema indicará que dicho elemento puede tener como secundarios todos los elementos globales del esquema. Por ejemplo, si un elemento llamado `email` no tiene asignado ningún tipo de datos, aparecerá en la Estructura del esquema con todos los elementos globales como posibles secundarios (p. ej. `person`, `address`, `city`, `tel`, etc.). Para evitar esto, asigne un tipo de datos al elemento `email`, por ejemplo `xs:string`.

6.1.2 Esquemas de BD

El diseño SPS también puede estar basado en un esquema generado a partir de una BD o puede tener como esquema fuente un esquema de BD (basado en una BD). Hay dos formas de crear un esquema fuente a partir de un esquema de BD:

- El esquema de BD se genera a partir de una BD directamente nada más crear el archivo SPS nuevo (con el comando de menú **Archivo | Nuevo | Nuevo a partir de una base de datos** o **Archivo | Nuevo | Nuevo a partir de una columna XML de una tabla de base de datos**).

 © 2016-2022 Altova GmbH

- El esquema de BD se añade a un diseño SPS vacío que no tiene esquemas fuente (desde la ventana Vista general del diseño 45).

Los respectivos comandos de generación de esquemas de BD generan un esquema XML temporal a partir de la BD seleccionada y, a partir de él, crean una fuente en la ventana Estructura del esquema 48. O el comando carga un esquema basado en una BD (esquema de BD) a partir de una BD XML. Como elemento de documento se crea el elemento extraño `DB` y dentro de este elemento de documento se crea la estructura de la BD. Durante el proceso de creación del esquema, deberá seleccionar las tablas o los datos de la BD que desea importar. En el esquema XML estas tablas de la BD se convierten en elementos secundarios del elemento `DB` y también aparecen como entradas de la lista `Plantillas globales` (ventana Estructura del esquema).

Crear el esquema XML a partir de la BD

El proceso de creación del esquema XML a partir de la BD tiene dos pasos:

- Conexión a la BD 523: consiste en buscar el archivo de BD (si es una BD MS Access) o generar una cadena de conexión (para los demás tipos de BD)
- Selección de datos de la BD 609: consiste en seleccionar la tabla o la estructura de datos de la BD. En el caso de una BD no XML, el esquema XML temporal que se genera y el archivo XML que se crea estarán basados en las tablas de datos seleccionadas.

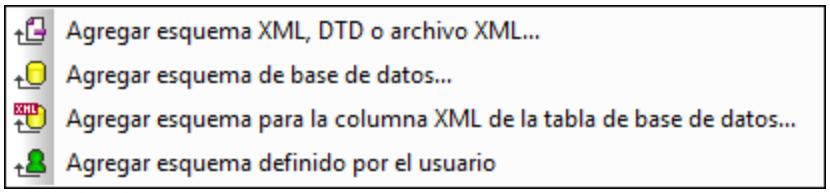
Cuando el esquema fuente aparece en la ventana Estructura del esquema, ya puede empezar a diseñar el SPS. Cada vez que haga clic en una pestaña de vista previa se genera un archivo XML temporal. La estructura de estos archivos XML está basada en el esquema XML generado y contiene los datos de las tablas seleccionadas de la BD.

6.1.3 Esquemas definidos por el usuario

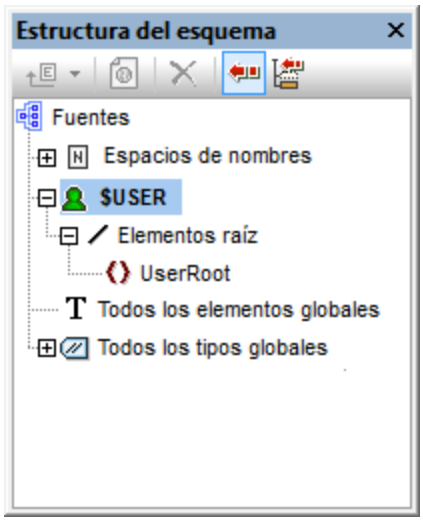
En la ventana Estructura del esquema 48 es muy fácil crear un esquema definido por el usuario, lo cual puede ser útil si tiene un documento XML que no está basado en ningún esquema pero desea crear un diseño SPS para este documento XML.

Para crear y añadir un esquema definido por el usuario:

1. Haga clic en el comando de menú **Archivo | Nuevo | Archivo nuevo (vacío)**. En la ventana Vista general del diseño 45, haga clic en el vínculo `Agregar fuente nueva...` (situado bajo la entrada **Fuentes**) y seleccione el comando **Agregar esquema definido por el usuario** (*imagen siguiente*).

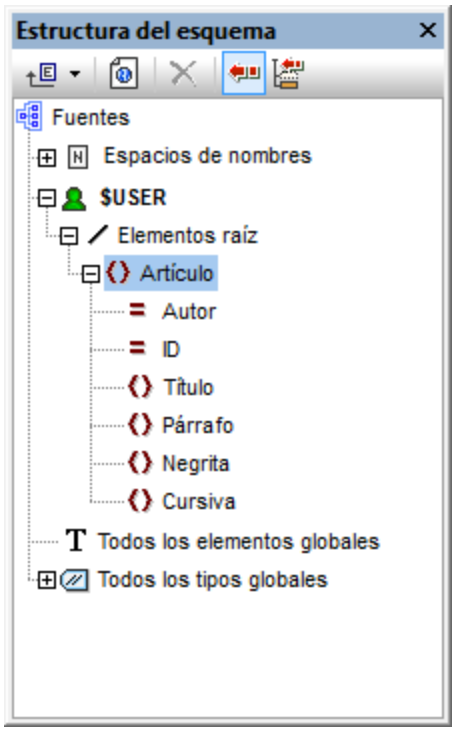


   Se crea el nuevo esquema, que lleva el parámetro `$USER` en la ventana Estructura del esquema (*imagen siguiente*).

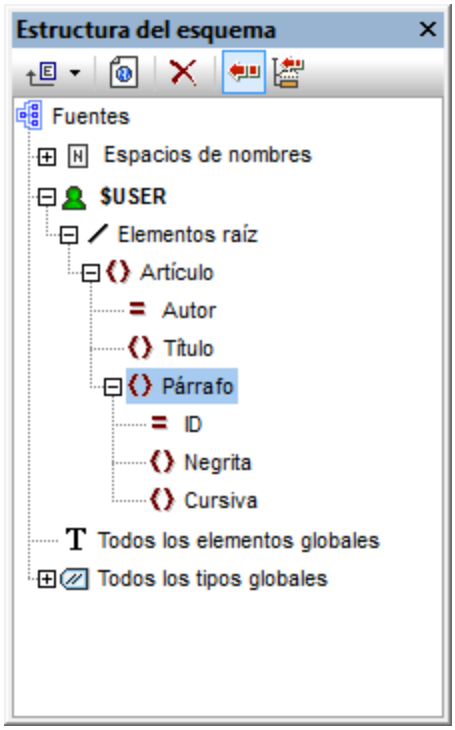


2. Bajo la entrada `/ Elementos raíz` aparece un solo elemento raíz (elemento de documento) 30 llamado `UserRoot`.
3. Haga doble clic en `UserRoot` y póngale el nombre del elemento de documento 30 del documento XML para el que está creando el esquema (p. ej. `Artículo`).
4. Para asignar un elemento secundario o un atributo al elemento de documento, seleccione el elemento de documento (`Artículo`) y haga clic en (i) el icono de la barra de herramientas de la ventana Estructura del esquema 48 (para añadir un elemento secundario) o en (ii) la flecha del icono y después en el comando **Anexar atributo** (para añadir un atributo). Otra opción es hacer clic con el botón derecho en el elemento de documento y seleccionar el comando correspondiente en el menú contextual. Si está seleccionado un elemento, los elementos nuevos se anexan/insertan como elementos secundarios del elemento seleccionado. Si está seleccionado un atributo, puede anexar o insertar otro atributo después o antes del atributo seleccionado, respectivamente. Tras añadir el elemento o atributo al árbol del elemento raíz, escriba el nombre del nombre o atributo. También puede arrastrar nodos de una posición a otra (*ver paso siguiente*). En la imagen siguiente, por ejemplo, el elemento `Artículo` es el elemento de documento. Los elementos `Título`, `Párrafo`, `Negrita` y `Cursiva` y los atributos `ID` y `Autor` se añadieron como secundarios de `Artículo`.

 © 2016-2022 Altova GmbH



5. Para mover los elementos `Negrita` y `Cursiva` y el atributo `ID` al nivel secundario de `Párrafo`, seleccione cada nodo y arrástrelo bajo el elemento `Párrafo`. Cuando aparezca el icono de flecha apuntando hacia abajo ( ), suelte el nodo, que se crea como secundario de `Párrafo` (*imagen siguiente*).



6. Cuando se selecciona un elemento que no es el elemento de documento y se añade un elemento o un atributo nuevo, el nodo nuevo se añade al mismo nivel que el elemento seleccionado. Y si arrastra un nodo (elemento o atributo) a un nodo de elemento, el nodo se convierte en secundario del elemento.

Editar nombres de nodo y eliminar nodos

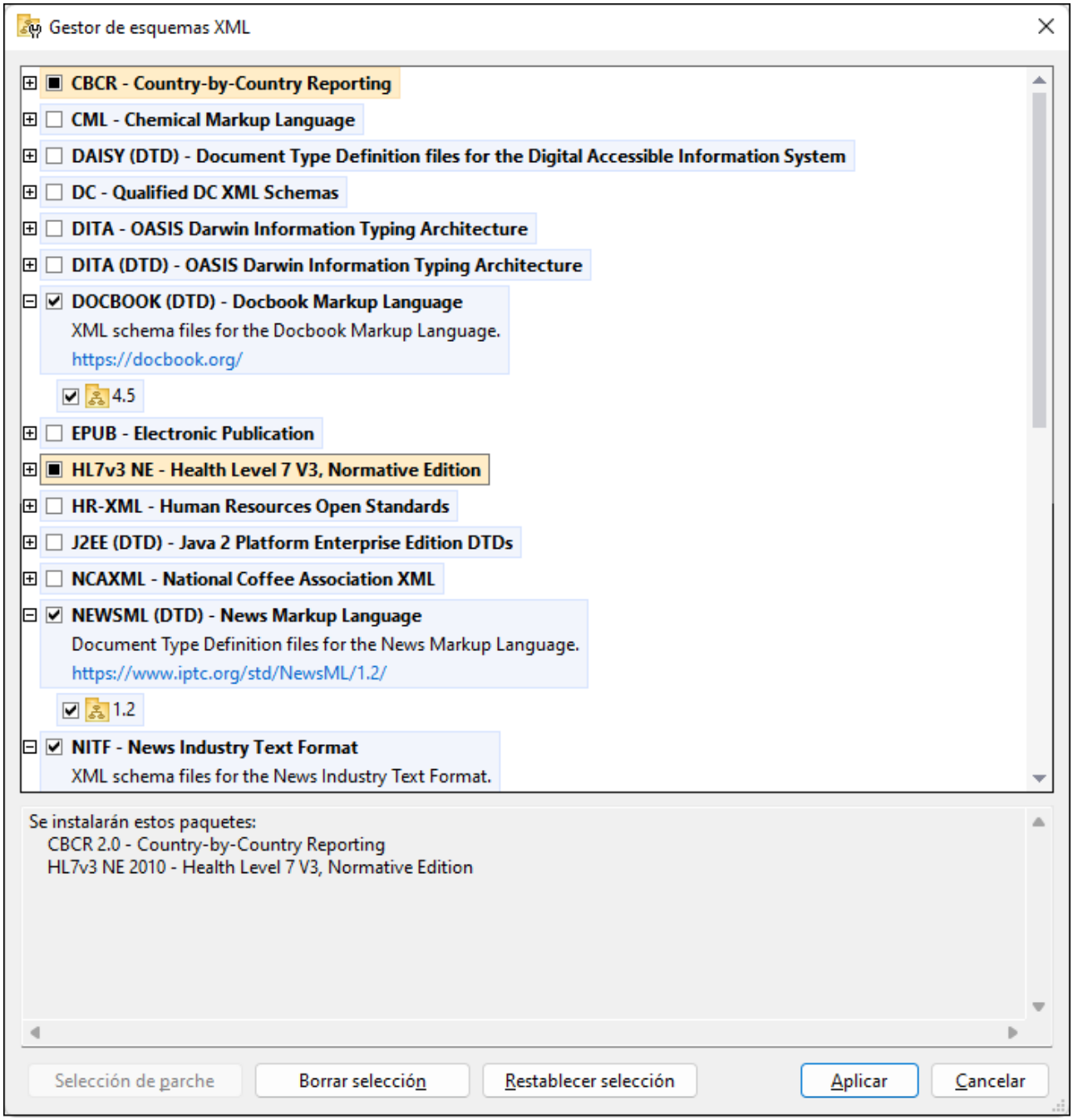
Para editar el nombre de un elemento o atributo, haga doble clic en su nombre y edítelo. Para eliminar un nodo, selecciónelo y haga clic en el icono **Eliminar** de la barra de herramientas. También el comando **Eliminar elemento** del menú contextual.

6.1.4 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 StyleVision.

- 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` 243, 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 StyleVision por primera vez en un mismo día natural.
- si StyleVision 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** 246 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** 244 desde la línea de comandos y después ejecute el comando **initialize** 242.

6.1.4.1 Ejecutar el gestor de esquemas

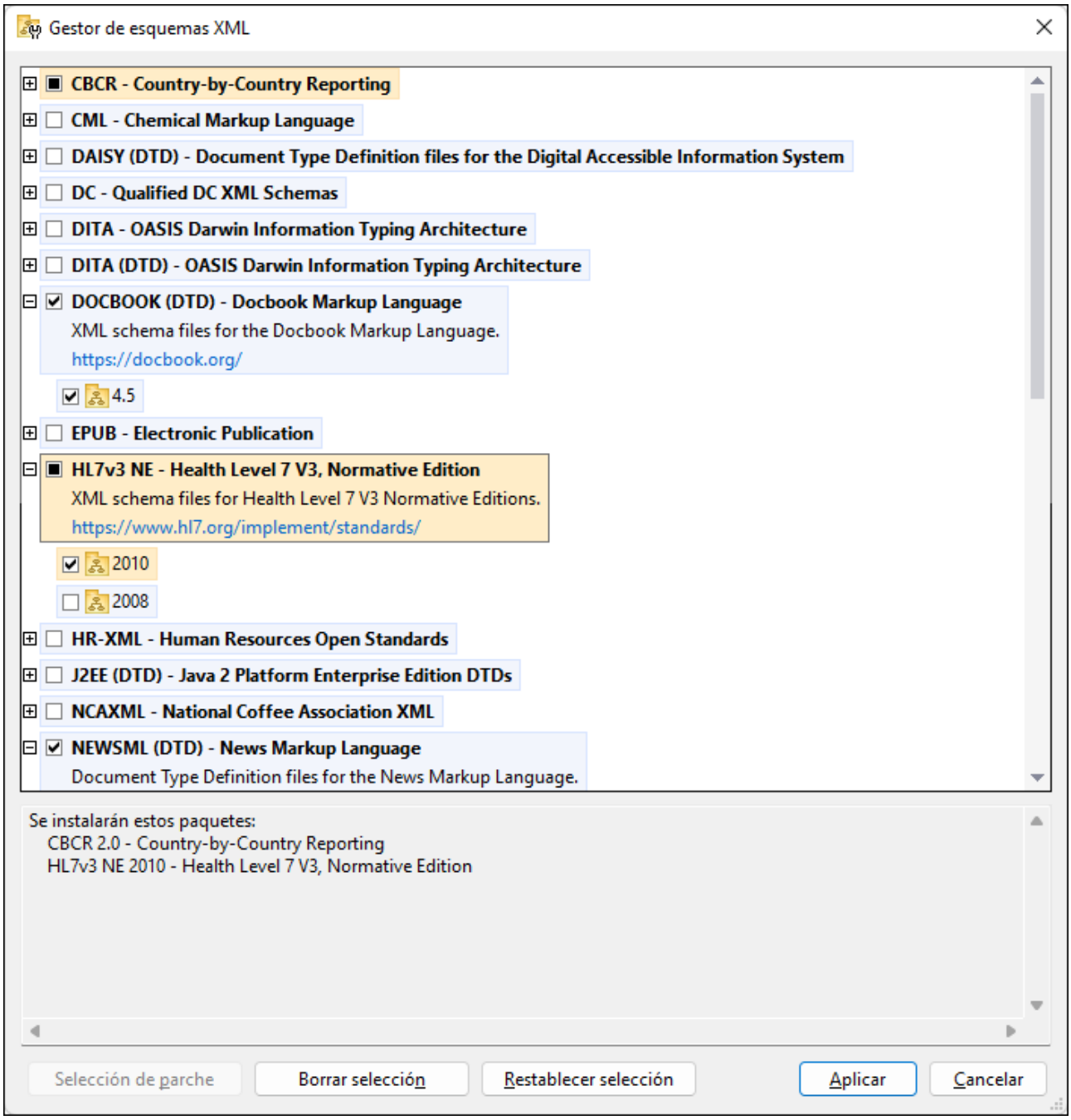
Interfaz gráfica del usuario

Hay varias formas de acceder a la IGU del Gestor de esquemas:

- *Durante la instalación de StyleVision:* 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 StyleVision:* 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 241, a continuación.

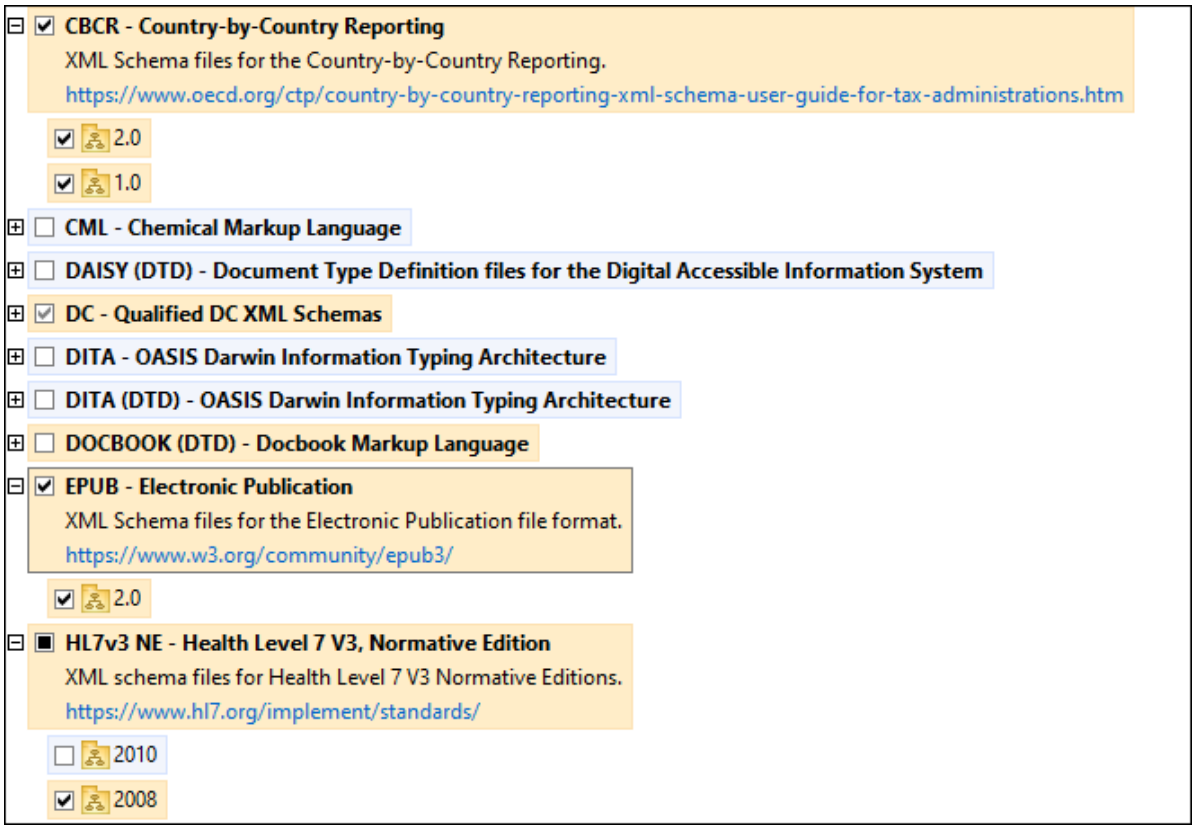
Para mostrar al ayuda de los comandos ejecute:

- *en Windows:* `xmlschemamanager.exe --help`
- *en Linux o macOS (solo para aplicaciones de servidor):* `sudo ./xmlschemamanager --help`

6.1.4.2 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`

6.1.4.3 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* 237.) 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` 244. Aparece una lista con los esquemas para las que hay parches disponibles.
1. Ejecute el comando `upgrade` 247 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` 244. Puede introducir tantos esquemas como quiera. Para más detalles consulte la descripción del comando `install` 243.

Nota: en Linux o macOS, use el comando `sudo ./xmlschemamanager`.

Instalar un esquema requerido
Si ejecuta un comando en StyleVision y StyleVision 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**.

6.1.4.4 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` 245:

```
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` 244. Puede introducir tantos esquemas como quiera. Para más detalles consulte la descripción del comando `uninstall` 245.

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` 244.

 © 2016-2022 Altova GmbH

Una vez haya ejecutado este comando, asegúrese de que ejecuta también el comando initialize 242 para recrear el directorio caché. También puede ejecutar el comando reset 244 con la opción -i.

Recuerde que reset -i 244 restaura la instalación original del producto, por lo que es recomendable ejecutar el comando update 246 después de restaurar el gestor. Puede ejecutar el comando reset 244 con las opciones -i o -u.

6.1.4.5 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.

6.1.4.5.1 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
```

6.1.4.5.2 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`244.)
- 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
```

6.1.4.5.3 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
```

6.1.4.5.4 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` 244. 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* 232.

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
```

6.1.4.5.5 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**

6.1.4.5.6 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 242 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 246 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**

6.1.4.5.7 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` 244. 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* 232.

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
```

6.1.4.5.8 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` 244 e `initialize` 242.

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

6.1.4.5.9 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 Combinar datos XML de distintas fuentes

Si usa la versión XSLT 2.0 o XSLT 3.0 con el diseño SPS podrá combinar los datos XML de varios archivos XML de origen.

Lo normal es basar la combinación de datos en un componente de datos común, como un número de identificación. Por ejemplo, un empleado de una compañía, al que se identifica con un número de Id. personal, puede tener sus datos personales repartidos por varios archivos XML del departamento de recursos humanos: (i) sus datos personales, (ii) datos sobre su nómina, (iii) datos sobre horas de trabajo y vacaciones, (iv) formación y cursos, etc. Los datos de todos estos archivos se pueden unir en un solo documento de salida usando el número de Id. como clave.

Nota: en la edición Enterprise Edition se pueden incluir varios esquemas fuente en el diseño SPS. De ese modo, puede seleccionar nodos de otros esquemas usando el nombre de parámetro del esquema pertinente (como en el ejemplo que aparece más abajo). En las ediciones Professional y Basic Edition puede usar la función XPath 2.0 `doc()` para buscar el archivo XML correspondiente y el nodo XML deseado. La función XPath 2.0 `doc()` ofrece acceso a la raíz de documento de documentos XML externos, lo cual permite insertar en documentos de salida el contenido de nodos de documentos XML externos. Por tanto, puede usar cálculos automáticos 293 que utilicen la función `doc()` para combinar datos XML (*ver ejemplo siguiente*).

Ejemplo

La carpeta (Mis) Documentos 34, `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples`, contiene el archivo de ejemplo `MergeData_2_Files.sps`. Este archivo ilustra cómo combinar datos de varios archivos XML diferentes. El diseño SPS selecciona datos de un pedido (`MergeOrder.xml`) realizado por un cliente ficticio:

```
<?xml version="1.0" encoding="UTF-8"?>
<Order xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="MergeOrder.xsd">
   <Item partNum="238-KK" quantity="3" shipDate="2000-01-07" comment="With no inclusions,
please."/>
   <Item partNum="748-OT" quantity="1" shipDate="2000-02-14" comment="Valentine's day
packaging."/>
   <Item partNum="229-OB" quantity="1" shipDate="1999-12-05"/>
   <Item partNum="833-AA" quantity="2" shipDate="1999-12-05" comment="Need this for the
holidays!"/>
</Order>
```

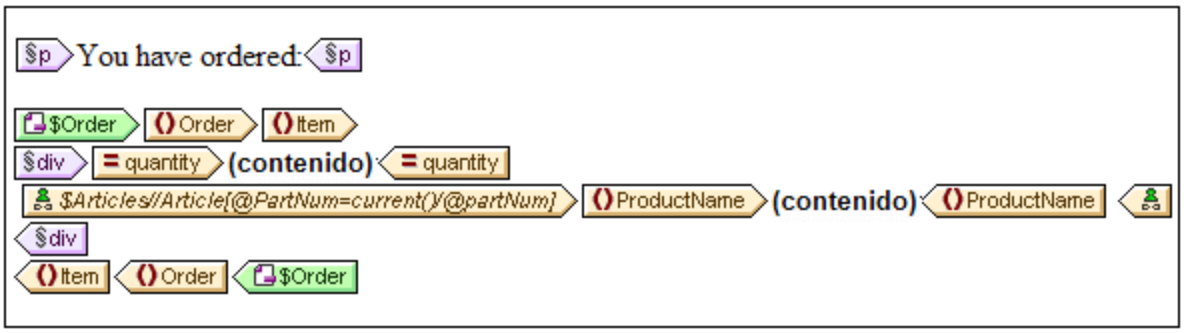
El valor del atributo `/Order/Item/@partNum` de este archivo (*fragmento anterior*) se usa para seleccionar los productos del pedido en el catálogo de artículos, que está en este otro archivo (`MergeArticles.xml`):

```
<?xml version="1.0" encoding="UTF-8"?>
<Articles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="MergeArticles.xsd">
   <Article PartNum="833-AA">
      <ProductName>Lapis necklace</ProductName>
      <Price>99.95</Price>
   </Article>
   <Article PartNum="748-OT">
      <ProductName>Diamond heart</ProductName>
      <Price>248.90</Price>
```

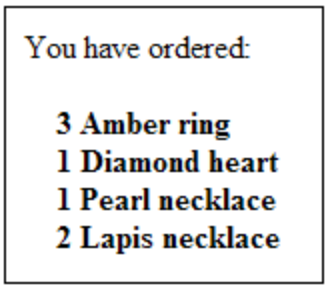
 © 2016-2022 Altova GmbH

```
   </Article>
   <Article PartNum="783-KL">
      <ProductName>Uncut diamond</ProductName>
      <Price>79.90</Price>
   </Article>
   <Article PartNum="238-KK">
      <ProductName>Amber ring</ProductName>
      <Price>89.90</Price>
   </Article>
   <Article PartNum="229-OB">
      <ProductName>Pearl necklace</ProductName>
      <Price>4879.00</Price>
   </Article>
   <Article PartNum="128-UL">
      <ProductName>Jade earring</ProductName>
      <Price>179.90</Price>
</Article>
       ...
</Articles>
```

Para combinar los datos, primero se creó una plantilla definida por el usuario 270 dentro de la plantilla `/Order/Item` (*imagen siguiente*) que selecciona el correspondiente elemento `Article` en el archivo `MergeArticles.xml` con ayuda del número del atributo `partNum`. La expresión XPath (que está en el contexto `/Order/Item`) sería esta: `$Articles//Article[@PartNum=current()/@partNum]`

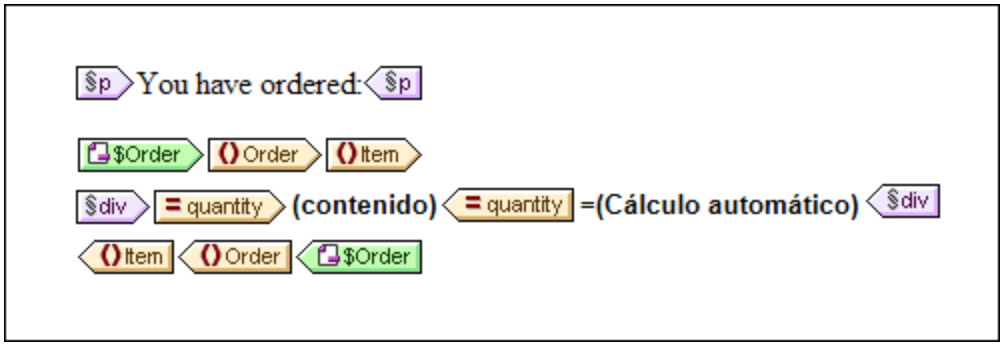


Esta plantilla genera un resultado parecido a este:



Observe que la cantidad solicitada de cada artículo se toma del archivo `MergeOrder.xml`, mientras que el nombre del artículo solicitado se toma del archivo `MergeArticles.xml`. Además, el nodo `ProductName` se selecciona dentro del contexto de la plantilla `/Articles/Article`.

Podemos obtener el mismo resultado con un cálculo automático 293 (*imagen siguiente*). Arrastre el atributo `quantity` desde la ventana Estructura del esquema y cree contenido a partir del atributo. Después añada un cálculo automático como el de la imagen siguiente y asígnele una expresión XPath (*ver más abajo*).



La expresión XPath del cálculo automático puede apuntar al nodo necesario usando el parámetro de otro esquema fuente o usando la función `doc()`:

```
$Articles//Article[@PartNum=current()/@partNum]/ProductName
```

o

```
doc('MergeArticles.xml')//Article[@PartNum=current()/@partNum]/ProductName
```

Observe que, mientras la primera expresión XPath utiliza un parámetro para hacer referencia a otro esquema XML (función disponible solamente en la edición Enterprise Edition), la segunda expresión utiliza la función XPath 2.0 `doc()` (función disponible también en las ediciones Professional y Basic).

© 2016-2022 Altova GmbH

6.3 Archivos SPS modulares

En StyleVision puede usar las plantillas globales de un diseño SPS, así como sus fragmentos de diseño, funciones JavaScript y componentes de diseño de página, en otros diseños SPS. Así podrá:

1. Reutilizar las plantillas globales y otros componentes en varios SPS diferentes. Con ello conseguirá documentos de salida consistentes a partir de un solo diseño.
2. Modular los diseños SPS y darles una estructura más flexible.

Resumiendo, puede añadir diseños SPS como módulos de otros diseños SPS. Algunos tipos de componentes (u objetos) de estos módulos estarán a disposición del SPS modular de destino.

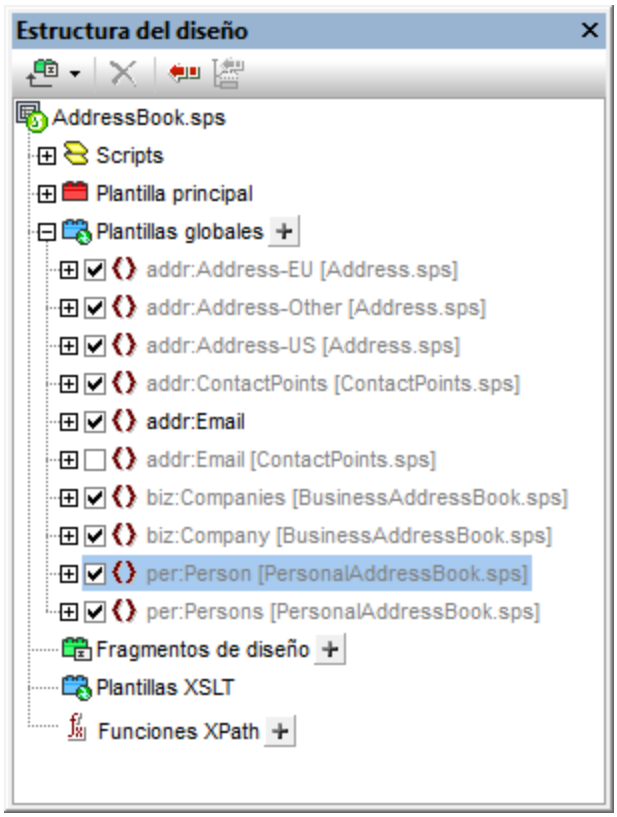
Esta sección se divide en estos apartados:

Objetos modulares disponibles

En este apartado no solo describimos bajo qué condiciones están disponibles los componentes de un SPS modular. También enumeramos los componentes a los que no tendrá acceso el SPS modular. No olvide que si un módulo contiene módulos, estos se añaden recursivamente al SPS modular. De modo que la composición por medio de módulos puede extenderse varios niveles.

Crear un diseño SPS modular

Para crear un diseño SPS modular primero debemos añadir el módulo SPS necesario 255 al SPS principal. Todas las funciones JavaScript, plantillas globales, fragmentos de diseño y funciones XPath del módulo recién añadido estarán a disposición del diseño SPS modular. Todos estos objetos se enumeran en la ventana Estructura del diseño, bajo el título correspondiente (*imagen siguiente*) y se pueden activar/desactivar marcando o no la casilla.



Estos objetos se pueden reutilizar en el SPS modular. Por ejemplo, para aplicar las plantillas globales en el SPS modular basta con activarlas en la ventana Estructura del diseño. Los fragmentos de diseño, por su parte, se deben arrastra desde la ventana Estructura del diseño hasta la posición deseada. Las funciones JavaScript se asignan como controladores de eventos desde la ventana Propiedades a cada componente del diseño. Y las funciones XPath disponibles (y activadas) se pueden usar en expresiones XPath.

Para más información consulte el apartado Crear un archivo SPS modular 255.

Terminología

Cuando se usa un archivo SPS como módulo de un diseño SPS, decimos que el módulo SPS se **agrega** al archivo SPS. Los dos SPS se denominan ***módulo SPS agregado*** y ***archivo SPS modular (de destino)*** respectivamente. Cuando se agrega un módulo SPS, sus objetos también se añaden al archivo SPS modular. Estos objetos se denominan ***objetos modulares*** y los hay de varios tipos: plantillas globales, fragmentos de diseño, funciones JavaScript y componentes de diseño de página.

6.3.1 Objetos modulares

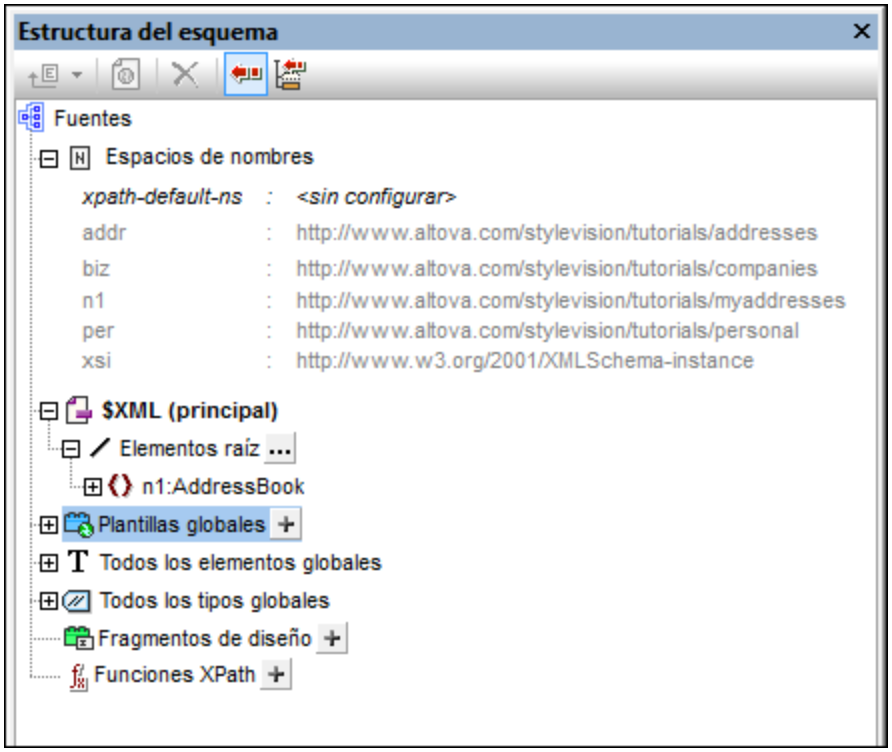
Este apartado enumera los objetos de los módulos SPS agregados 252 que están a disposición del archivo SPS modular 252 y explica cómo usar cada objeto en el archivo SPS modular. Consulte el apartado Crear un archivo SPS modular 255 para ver cómo se crea un SPS modular paso a paso. Para terminar, el presente apartado enumera los objetos de un módulo SPS agregado a los que el archivo SPS modular no tiene acceso.

- Declaraciones de espacios de nombres 253

- Plantillas globales 253
- Fragmentos de diseño 254
- Módulos agregados 254
- Scripts 255
- Estilos CSS 255
- Diseños de página 255
- Objetos modulares no disponibles 255

Declaraciones de espacios de nombres

Cada archivo SPS almacena una lista de identificadores URI de espacio de nombres con sus prefijos. Cuando se añade un módulo SPS, sus espacios de nombres se comparan con los de los esquemas fuente del archivo SPS modular de destino. Si un URI de espacio de nombres del módulo SPS agregado coincide con un URI de espacio de nombres del esquema fuente del archivo SPS modular, entonces el prefijo del esquema fuente del archivo SPS modular se adopta como prefijo del espacio de nombres del módulo SPS agregado. Si un URI de espacio de nombres del módulo SPS no coincide con ningún URI del esquema fuente del archivo SPS modular, entonces se emite un error.

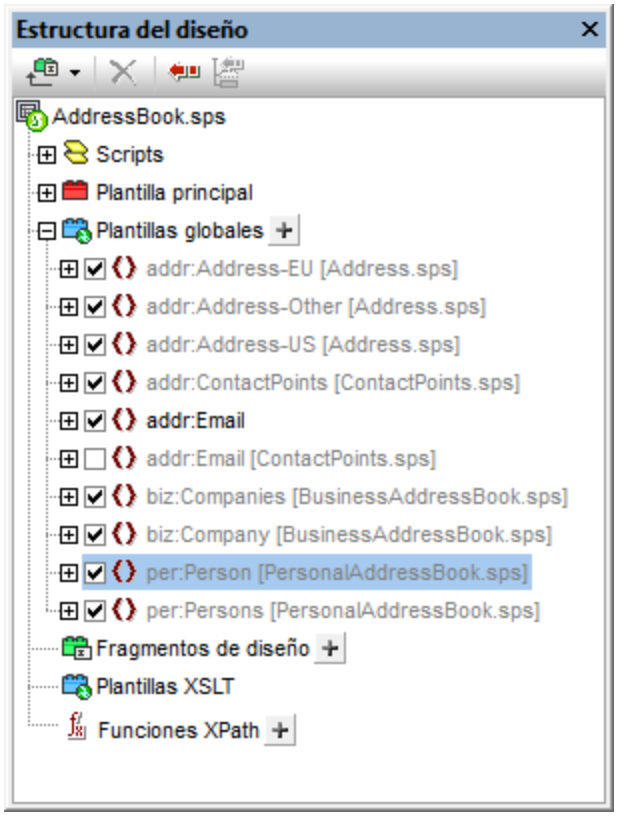


La imagen anterior muestra los espacios de nombres de un SPS, junto con sus prefijos, en la ventana Estructura del esquema. Estos espacios de nombres proceden de los esquemas fuente y no se pueden editar.

Plantillas globales

Las plantillas globales 266 del módulo SPS agregado están a disposición del archivo SPS modular y aparecen en la ventana Estructura del diseño 52 (*imagen siguiente*). Las plantillas globales se activan/desactivan automáticamente (marcadas o sin marcar) dependiendo del estado de activación del módulo agregado. Si desea crear una plantilla global para reemplazar una plantilla global de un módulo agregado, cree la nueva plantilla global haciendo clic en el icono situado junto a la entrada **`Plantillas globales`**. Aparece el

cuadro de diálogo "Agregar plantilla global nueva", donde puede seleccionar el elemento/atributo para el que desea crear la plantilla global. Otra opción es introducir una expresión XPath que seleccione el nodo necesario en el esquema. Cuando termine haga clic en **Aceptar** y StyleVision pregunta si desea activar la plantilla global nueva en lugar de la plantilla global del módulo agregado. Las plantillas globales se pueden activar/desactivar en cualquier momento con solo marcar/no marcar su casilla.



Observe que la plantilla principal de los módulos agregados no está disponible. Esto significa que si deesa reutilizar una plantilla, esta debe ser una plantilla global del módulo. Si no definió una plantilla global para un elemento determinado y se invoca el procesamiento para dicho elemento, se usará el procesamiento predeterminado del elemento (plantillas integradas XSLT).

Fragmentos de diseño

Los fragmentos de diseño 278 del módulo SPS agregado están a disposición del archivo SPS modular y aparecen en la ventana Estructura del diseño 52 (*imagen siguiente*). Cuando inserte un fragmento de diseño en el diseño, compruebe que lo inserta dentro del nodo de contexto adecuado.

Módulos agregados

Los módulos de los módulos SPS agregados también están a disposición del archivo SPS modular. Es decir, cuando se añade un módulo, están también disponibles todos sus módulos y los módulos de sus módulos, etc. Todos estos módulos juntos deben dar lugar a un modelo de contenido que sea válido según los esquemas fuente del archivo SPS modular al que se agregan. Todos los módulos se enumeran y se pueden gestionar desde la ventana Vista general del diseño 45.

 © 2016-2022 Altova GmbH

Scripts

Los scripts de todos los módulos SPS agregados están a disposición del archivo SPS modular y aparecen en la ventana Estructura del diseño 52. De hecho, los scripts de todos módulos agregados se recopilan en una biblioteca a la que puede acceder (desde el archivo SPS modular) desde el cuadro de diálogo "Propiedades".

Estilos CSS

Los estilos globales de los módulos SPS agregados están a disposición del archivo SPS modular y las reglas de estilos aparecen en la ventana Repositorio de estilos 391. Los archivos CSS también aparecen en la ventana Vista general del diseño 45. Igualmente, los archivos CSS externos a los que tiene acceso el módulo SPS agregado también están a disposición del archivo SPS modular.

Diseños de página

Los diseños de página de un módulo agregado están a disposición del archivo SPS modular y aparecen en la ventana Estructura del diseño 52.

Objetos modulares a los que no puede acceder el SPS modular

Estos son los objetos de los módulos agregados a los que el archivo SPS modular no puede acceder:

- ***Definiciones de parámetros:*** se ignoran.
- ***Esquemas fuente:*** el esquema fuente en el que se basa el módulo SPS se ignora. Tenga en cuenta que el modelo de contenido del elemento de documento del módulo SPS agregado debe estar dentro del modelo de contenido del archivo SPS modular. De lo contrario, no sería posible usar el módulo SPS correctamente. Si quiere, puede agregar un esquema definido por el usuario al archivo SPS modular. Este esquema adicional puede tomar el modelo de contenido de las plantillas globales añadidas.
- ***Archivo XML de trabajo y archivo de plantilla XML:*** se ignoran las referencias a estos archivos. El archivo SPS modular utiliza sus propios archivos XML de trabajo y de plantilla XML.
- ***Espacios de nombres predeterminados XPath:*** no se pasan al archivo SPS modular.

6.3.2 Crear un archivo SPS modular

Los archivos SPS modulares se crean en cuatro pasos:

1. Se diseña y se guarda el módulo SPS que queremos agregar 255.
2. Se añade el módulo 256 al archivo SPS donde deseamos usarlo (en adelante, el archivo SPS modular).
3. Se activan o desactivan los objetos modulares 258.
4. Se aplican los objetos donde sea necesario.

Módulo SPS que se debe agregar

Hay dos cosas que debe tener cuenta cuando cree un SPS para añadirlo a otro SPS:

1. Las plantillas que se podrán usar en el archivo SPS modular de destino 252 solamente pueden ser plantillas globales 266. Esto significa que las plantillas que desea volver a utilizar deben crearse como plantillas globales en el módulo SPS que desea agregar 252.

2. La estructura de documento definida en el módulo SPS que se debe agregar debe ser válida dentro del modelo de contenido definido por los esquemas fuente del SPS modular de destino 255. Si una plantilla agregada no está en el modelo de contenido definido por el esquema principal del SPS, puede definir su modelo de contenido en un esquema definido por el usuario.

El módulo SPS que desea agregar al SPS modular de destino debe estar basado en uno de estos esquemas:

- El esquema principal del archivo SPS modular de destino. En este caso, cuando se añade el módulo SPS, las plantillas globales añadidas formarán parte del modelo de contenido del esquema principal del SPS modular. El resultado de estas plantillas globales se puede editar en la vista Authentic.
- Un esquema que defina un modelo de contenido que sea parte del modelo de contenido definido por el esquema principal del archivo SPS modular. En este caso, cuando se añade el módulo SPS, las plantillas globales encajarán en el modelo de contenido del esquema principal del archivo SPS modular. El resultado de estas plantillas globales se puede editar en la vista Authentic.
- Un esquema que defina un modelo de contenido que **no forme parte** del modelo de contenido definido por el esquema principal del archivo SPS modular. Cuando se añade este módulo SPS, sus plantillas globales no formarán parte del modelo de contenido del esquema principal del archivo SPS modular. Sin embargo, se pueden usar para generar documentos de salida si se utiliza un esquema definido por el usuario que defina un modelo de contenido que incluya el modelo de contenido de las plantillas globales. Pero el resultado de la vista Authentic de estas plantillas globales no se puede editar.

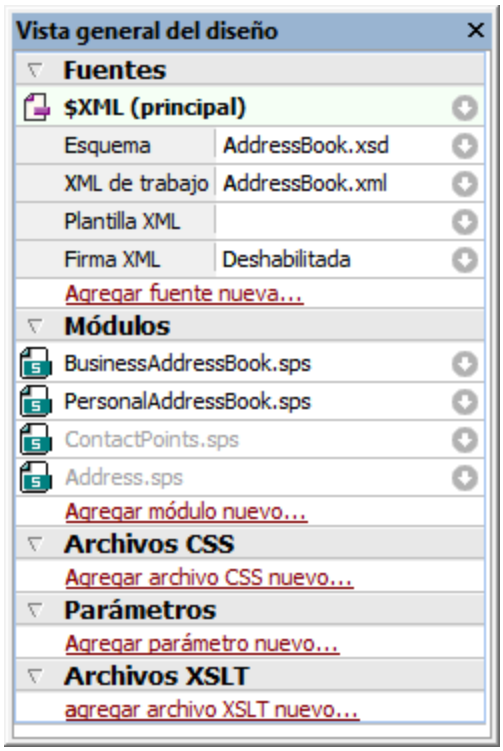
Cuando defina modelos de contenidos en sus esquemas, preste atención a los espacios de nombres 253 utilizados porque determinan los nombres expandidos de los nodos.

Si quiere, puede usar un archivo XML de trabajo 30 para probar el resultado del módulo SPS que desea agregar. La referencia a este archivo XML de trabajo será ignorada por el archivo SPS modular de destino 255.

Agregar el módulo SPS

Para agregar un módulo SPS al diseño SPS de destino haga clic en el vínculo `Agregar módulo nuevo...` de la ventana Vista general del diseño 45 (*imagen siguiente*). En el cuadro de diálogo que aparece busque el archivo SPS que desea añadir y después haga clic en **Abrir**.

© 2016-2022 Altova GmbH



El módulo se añade al SPS y aparece bajo el título **Módulos** de la Vista general del diseño. En la imagen anterior, por ejemplo, los módulos `BusinessAddressBook.sps` y `PersonalAddressBook.sps` se añadieron al archivo SPS `AddressBook.sps` (el archivo SPS activo). Todos los objetos modulares añadidos aparecen en la ventana Estructura del diseño. Los archivos CSS añadidos, sin embargo, aparecen en la ventana Vista general del diseño. Si los módulos añadidos incluyen a su vez otros módulos, estos módulos importados indirectamente también aparecen bajo el título **Módulos** de la Vista general del diseño, pero atenuados en color gris. Si pasa el puntero del ratón sobre los módulos, aparece información sobre el módulo, si se importó indirectamente, etc.

Para abrir uno de los módulos agregados o de los módulos importados indirectamente, haga clic con el botón derecho en el módulo y seleccione **Abrir módulo** en el menú contextual.

Orden de los módulos agregados

El orden en que se añaden y enumeran los módulos es importante a la hora de dar prioridad a los estilos CSS. Las reglas de estilo CSS de un módulo añadido más tarde (que aparece más abajo en la lista) tienen prioridad sobre las reglas de estilo definidas en módulos añadidos antes (que aparecen más arriba en la lista). Los estilos CSS del archivo SPS modular tienen prioridad sobre los estilos CSS de los módulos añadidos. Para cambiar la posición relativa de un módulo agregado (y por tanto el orden de prioridad de las reglas CSS), haga clic con el botón derecho en la Vista general del diseño y después seleccione **Subir** o **Bajar** en el menú contextual.

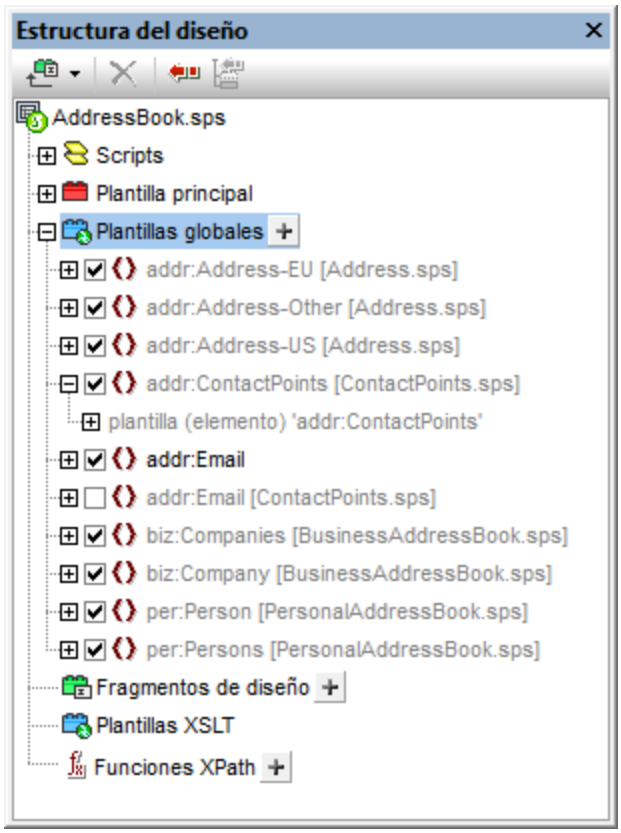
El orden de los módulos no importa a la hora de resolver conflictos entre scripts, plantillas globales, fragmentos de diseño y componentes de diseño de página.

Alertas de cambios en los archivos

Si uno de los archivos añadidos (ya sea un módulo SPS, un esquema o un archivo XML de trabajo) se modifica después de abrir el archivo SPS modular, aparece una notificación avisando de los cambios y preguntando si desea actualizar el archivo SPS modular.

Activar/desactivar el objeto añadido

Todos los objetos modulares de todos los módulos agregados (directa o indirectamente) se añaden al archivo SPS modular de destino y se enumeran bajo el título correspondiente en la ventana Estructura del diseño: **`Scripts`**, **`Plantillas globales`**, **`Fragmentos de diseño`**, **`Plantillas XSLT`** y **`Funciones XPath`**. Junto a estos objetos aparece una casilla (*imagen siguiente*) que sirve para activar o desactivar el objeto. Si el objeto se desactiva, es como si se eliminara del diseño SPS.



En la imagen anterior, por ejemplo, todas las plantillas globales utilizadas en el archivo SPS modular `AddressBook.sps` aparecen bajo el título **`Plantillas globales`**. Las plantillas que se añadieron a través de otros módulos (directa o indirectamente) aparecen atenuadas en color gris. Las que se crearon directamente en `AddressBook.sps` aparecen en negrita. En la imagen puede ver que en `AddressBook.sps` solamente se creó una plantilla global: `addr:Email`. Las demás plantillas globales se añadieron desde otros módulos y junto al nombre de la plantilla aparece el nombre del módulo en el que está definida.

Observe que hay dos plantillas globales para `addr:Email`, una en el archivo SPS modular `AddressBook.sps` y otra en el módulo agregado `ContactPoints.sps`. Si hay varias plantillas globales con el mismo nombre expandido, solo se puede usar una de ellas. Para seleccionar qué plantilla se usa, marque su casilla. Otra opción es activar la plantilla global desde su menú contextual en la vista **Diseño**. Este mecanismo puede ser

 © 2016-2022 Altova GmbH

útil cuando necesite (i) reemplazar una plantilla global añadida con otra creada en el archivo SPS modular o cuando (ii) se defina una plantilla global para un elemento en varios módulos agregados.

Para borrar una plantilla global que esté definida en el archivo SPS activo, selecciónela y haga clic en el botón **Eliminar**. Sin embargo, las plantillas globales que se definieron en un módulo agregado no se pueden eliminar del SPS modular. Para eliminarlas debe abrir el módulo SPS agregado y eliminarlas allí.

De la misma manera puede activar y desactivar scripts, fragmentos de diseño y componentes de diseño de página.

Aplicar o usar objetos modulares

En el archivo SPS modular de destino 252 puede diseñar sus plantillas como en un diseño SPS normal y corriente. Cada tipo de objeto modular se utiliza o aplica de una forma diferente. Antes de nada, compruebe que los objetos modulares que desea utilizar están activados 258.

Plantillas globales
Cuando quiera usar una plantilla global 266 de cualquier módulo SPS, primero debe comprobar si la plantilla global se aplica o no. Hay dos maneras de hacer esto:

- En la plantilla principal, especifique que la plantilla de elemento usa la plantilla global para ese elemento o copia esa plantilla global localmente. Estos dos comandos están en el menú contextual que aparece al hacer clic con el botón derecho en la etiqueta del elemento en la vista **Diseño**.
- En la plantilla principal, los marcadores de posición `(contenido)` y `(resto de contenido)` hacen que se apliquen las plantillas y, por tanto, se procesen las plantillas globales pertinentes.

Fragmentos de diseño
Si quiere usar un fragmento de diseño, arrástrelo desde la Estructura del esquema hasta la posición deseada de la plantilla global o principal. Compruebe que la posición donde coloca el fragmento de diseño es el nodo de contexto adecuado para el fragmento. Para más información consulte el apartado Fragmentos de diseño 278.

Scripts
Todas las funciones JavaScript (ya sean de un módulo SPS o del archivo SPS modular) se pueden usar como controladores de eventos y se pueden configurar desde la ventana Propiedades 499.

6.3.3 Ejemplo: una libreta de direcciones

La carpeta (Mis) Documentos 34, `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\ModularSPS`, contiene ejemplos de archivos SPS modulares. Los archivos de ejemplo de esta carpeta forman un proyecto: una libreta de direcciones modular, con contactos personales y de negocios. Este ejemplo no solo muestra los mecanismos utilizados para implementar los módulos, sino que ilustra las ventajas de usarlos.

- La libreta de direcciones está formada por dos módulos: (i) una libreta de direcciones de negocios y (ii) una libreta de direcciones personal. Cada una tiene un archivo SPS con un diseño diferente. En este caso los módulos se utilizan para formar una unidad de mayor tamaño. Los módulos son los componentes de esta unidad.
- Aunque los modelos de contenido de los dos módulos (la libreta de direcciones de negocios y la personal) tienen algunas diferencias, ambos tienen en común el módulo `ContactPoints`, que incluye

los principales datos de contacto: dirección, teléfono, fax y correo electrónico. Por tanto, el módulo `ContactPoints` puede ser compartido por las dos libretas de direcciones (la de negocios y la personal). En este caso el módulo se utiliza como unidad común compartida por varias unidades.

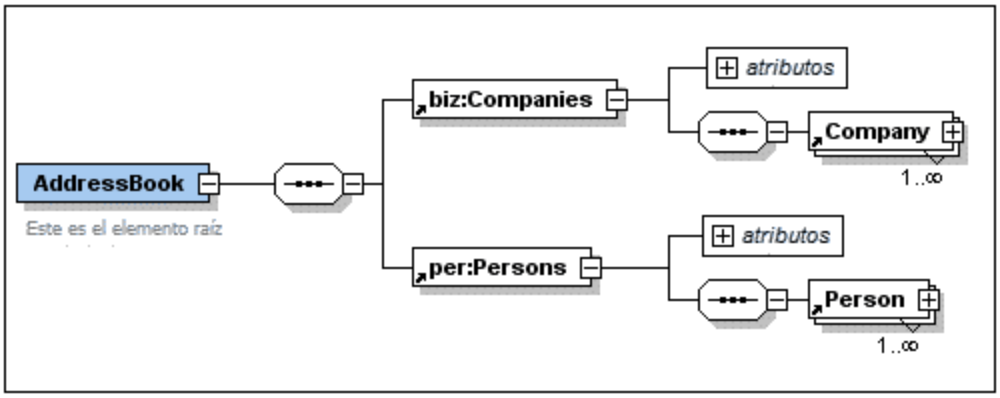
- Además, se pueden crear más módulos a partir del módulo `ContactPoints`. Por ejemplo, en el proyecto se creó el módulo `Address`, que contiene la dirección postal. Esta dirección postal puede tener tres modelos de contenido, dependiendo de si es una dirección de la UE, de EE UU o de otro lugar. El resultado de los tres modelos de contenido se define en un solo diseño SPS. No obstante, podrían tener SPS diferentes. En este caso, el uso de módulos aportaría aún más flexibilidad.

Para describir este proyecto de ejemplo dividimos la información en tres apartados:

- Los archivos de esquema 260
- Las fuentes de datos XML 261
- Los archivos SPS 262

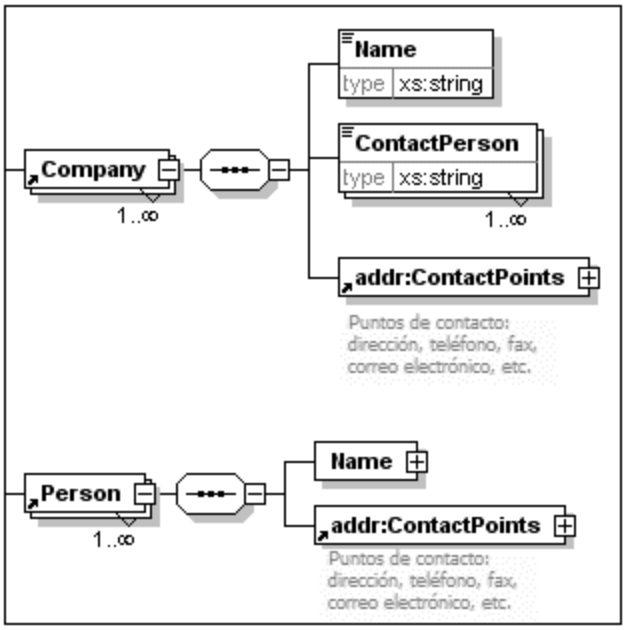
Los archivos de esquema

Cuando cree esquemas para archivos SPS modulares, lo más importante es no olvidarse de crear los elementos que quiere reutilizar como elementos globales. El esquema para la libreta de direcciones es `AddressBook.xsd`. Este esquema se construyó al importar los esquemas para la libreta de direcciones de negocios (`BusinessAddressBook.xsd`) y la libreta de direcciones personal (`PersonalAddressBook.xsd`). El esquema `BusinessAddressBook.xsd` ofrece un modelo de contenido para los contactos de negocios, mientras que el esquema `PersonalAddressBook.xsd` ofrece un modelo de contenido para los contactos personales (*imagen siguiente*).

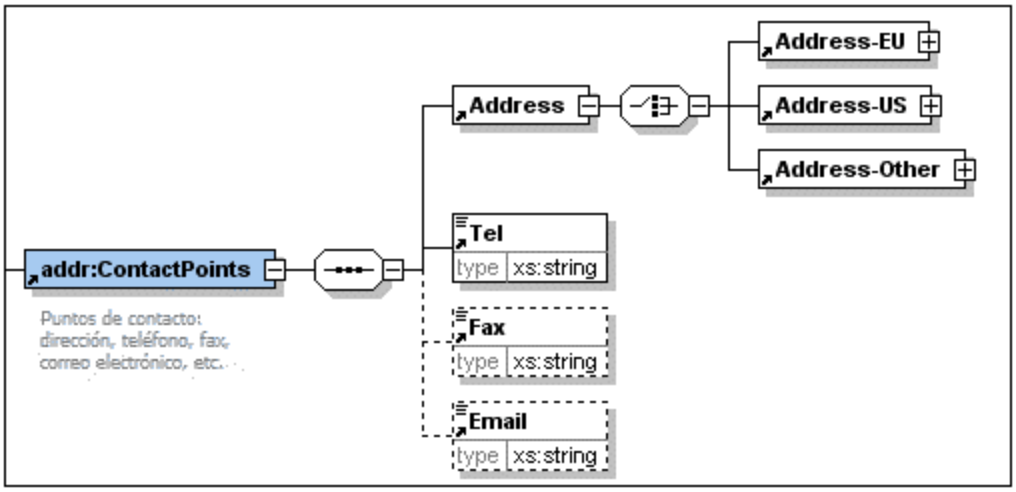


Ambos esquemas importan el esquema `ContactPoints.xsd` (*imagen siguiente*), que define un modelo de contenido para los datos de contacto.

© 2016-2022 Altova GmbH



Por último, el esquema `ContactPoints.xsd` (*imagen siguiente*) incluye el esquema `Address.xsd`, que define los tres tipos de modelos de contenido para las direcciones: direcciones de la EU, de EE UU y de otros lugares.



Las importaciones se usan cuando el esquema importado pertenece a un espacio de nombres distinto al del esquema de importación. Las inclusiones se usan cuando el esquema incluido pertenece a un espacio de nombres distinto al del esquema de inclusión.

Nota: las imágenes anteriores muestran el esquema correspondiente en la vista **Esquema** de Altova XMLSpy.

Las fuentes de datos XML

Los datos XML están en el archivo `AddressBook.xml`. Según la estructura de este archivo, el elemento `AddressBook` contiene los elementos secundarios `companies` y `persons`. Los modelos de contenido de estos

dos elementos se definen en los archivos de esquema `BusinessAddressBook.xsd` y `PersonalAddressBook.xsd`, respectivamente.

Hay dos archivos XML más, que corresponden a los esquemas `BusinessAddressBook.xsd` y `PersonalAddressBook.xsd`. Estos dos archivos XML, `BusinessAddressBook.xml` y `PersonalAddressBook.xml`, se usan como archivos XML de trabajo de los correspondientes archivos SPS.

Por tanto, los tres archivos XML son los archivos XML de trabajo 30 de estos módulos SPS:

- `AddressBook.xml => AddressBook.sps, ContactPoints.sps, Address.sps`
- `BusinessAddressBook.xml => BusinessAddressBook.sps`
- `PersonalAddressBook.xml => PersonalAddressBook.sps`

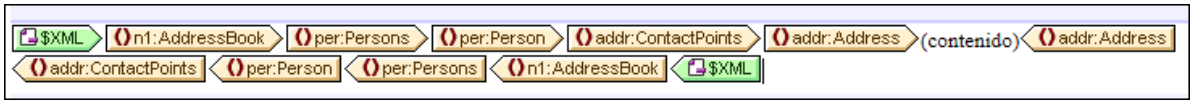
Los módulos SPS

Primero empezaremos por describir el módulo más básico del proyecto (`Address.sps`) y después describiremos todos los módulos que componen la libreta de direcciones (`AddressBook.sps`). Todos estos módulos SPS usan el esquema `AddressBook.xsd`.

Address.sps

Lo más importante es saber qué esquema y qué archivo XML de trabajo utiliza.

- El módulo `Address.sps` utiliza `AddressBook.xsd` como esquema, pero también podría utilizar `Address.xsd`, `ContactPoints.xsd`, `BusinessAddressBook.xsd` o `PersonalAddressBook.xsd` porque el elemento `Address` está presente en todos estos esquemas y estará disponible como elemento global. Cuando el módulo SPS se añade a otro módulo SPS, el esquema del módulo importado se ignora. Es decir, no importa qué esquema se utiliza si el archivo SPS se va a añadir a otro SPS como módulo.
- El archivo XML de trabajo es `AddressBook.xml`. Observe que la plantilla principal de `Address.sps` especifica que solo se debe procesar el elemento `Address`. Además, observe que se definieron plantillas globales para `Address-EU`, `Address-US` y `Address-Other`.



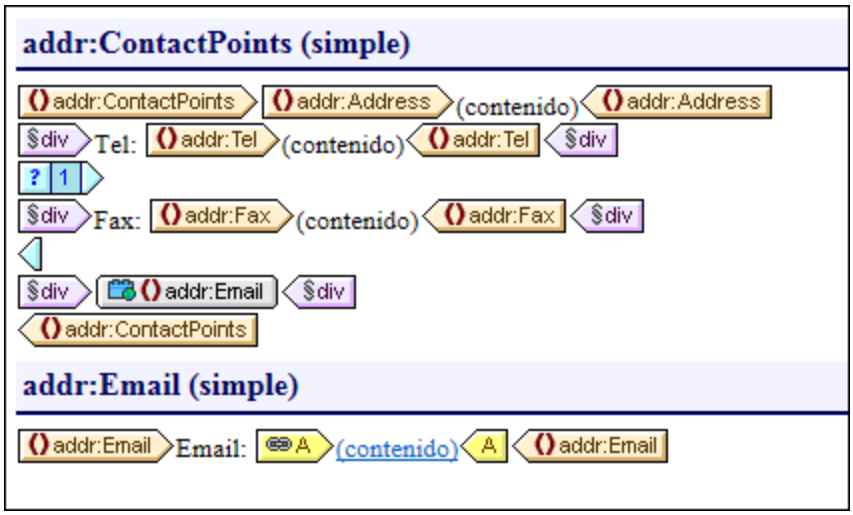
Como solamente se procesa el elemento `Address`, las vistas previas de los resultados solo muestran el resultado de `Address`. Cuando `Address.sps` se usa como módulo, las plantillas globales se añaden y la plantilla principal se ignora.

ContactPoints.sps

Este SPS importa el módulo anterior (`Address.sps`). Observe el uso de las plantillas globales dentro de otras plantillas y globales y de la plantilla principal.

- El módulo `ContactPoints.sps` utiliza `AddressBook.xsd` como esquema y `AddressBook.xml` como archivo XML de trabajo.
- `Address.sps` se añade como módulo a `ContactPoints.sps`, lo cual activa las plantillas globales de los elementos `Address-EU`, `Address-US` y `Address-Other`.
- Se definen las plantillas globales para los elementos `ContactPoints` y `Email`. Observe que la definición de `ContactPoints` utiliza la plantilla global de `Email` (*imagen siguiente*).

 © 2016-2022 Altova GmbH



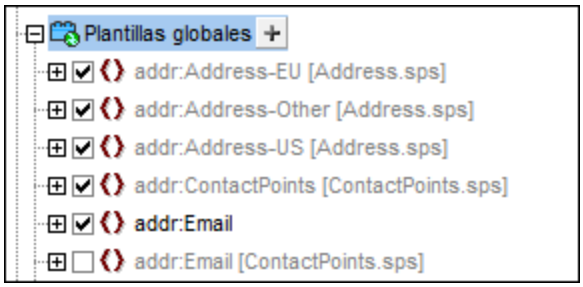
- La plantilla principal (necesaria para generar la vista previa) utiliza la plantilla global del elemento `ContactPoints`, lo cual permite obtener una vista previa del resultado de `ContactPoints`.

BusinessAddressBook.sps y PersonalAddressBook.sps
Estos archivos SPS importan un módulo cada uno, que a su vez importan otro módulo. Observe que la plantilla principal simplemente aplica las plantillas globales.

- Estos dos módulos usan el esquema `AddressBook.xsd`. Los archivos XML de trabajo de estos dos módulos son `BusinessAddressBook.xml` y `PersonalAddressBook.xml` respectivamente.
- El módulo `ContactPoints.sps` se añade a ambos módulos. Indirectamente se importa también el módulo `Address.sps`. Todas las plantillas globales de estos dos módulos estarán a disposición del archivo SPS modular de destino.
- En el módulo `BusinessAddressBook.sps` se definen plantillas globales para los elementos `Companies` y `Company`. Observe que la definición de `Company` utiliza la plantilla global de `ContactPoints`.
- En el módulo `PersonalAddressBook.sps` se definen plantillas globales para los elementos `Person` y `Persons`. La definición de `Person` utiliza la plantilla global de `ContactPoints`.

AddressBook.sps
Hay dos plantillas globales para el elemento `Email`. Cualquiera de las dos puede activarse.

- El diseño modular de destino `AddressBook.sps` utiliza el esquema `AddressBook.xsd` y el archivo de trabajo `AddressBook.xml`.
- Los módulos `BusinessAddressBook.sps` y `PersonalAddressBook.sps` se añaden al SPS modular de destino. Indirectamente se importan también los módulos `ContactPoints.sps` y `Address.sps`.
- Para el elemento `Email` se define una plantilla global. Esto significa que ahora hay dos plantillas globales para el elemento `Email`, una en el módulo `ContactPoints.sps` y otra en el módulo `AddressBook.sps` (*imagen siguiente*).



- En la lista **`Plantillas globales`** de la ventana Estructura del diseño (*imagen anterior*) puede seleccionar cuál de las dos plantillas globales debe activarse. En StyleVision no puede haber más de dos plantillas globales activas para el mismo elemento. La plantilla que esté activa se usa dentro de la plantilla global `ContactPoints`.
- La plantilla principal incluye contenido estático para el encabezado de los documentos de salida.

© 2016-2022 Altova GmbH

6.4 Plantillas y fragmentos de diseño

El documento SPS se compone de plantillas y es importante distinguir bien entre los diferentes tipos de plantilla que se pueden usar:

- ***Plantillas principales y globales:*** el documento SPS está formado por una plantilla principal 265 y, de manera opcional, por una o más plantillas globales 266. Desde la plantilla principal se puede hacer referencia a las plantillas globales.
- ***Plantillas nodo e iteradores de variables:*** plantillas que componen la plantilla principal y las plantillas globales. Una plantilla nodo 274 coincide con un nodo de un esquema fuente.
- ***Fragmentos de diseño:*** plantillas que se diseñan por separado para poder utilizarlas de nuevo en la plantilla principal o en las plantillas globales del documento SPS.

En esta sección describimos el papel que desempeñan las plantillas y los fragmentos de diseño en la estructura del documento SPS.

Tenga en cuenta que en esta sección no nos ocupamos de las propiedades de presentación 374 del diseño SPS, sino de su estructura. Además, en esta sección no se tienen en cuenta los componentes estructurales para medios impresos, como las portadas, los encabezados y los pie de página. Estos componentes se describen en la sección Diseñar medios impresos 419.

Nota: en la vista **Diseño** el archivo SPS puede incluir varias plantillas: la plantilla principal, varias plantillas globales, plantillas de diseño de página y fragmentos de diseño. Con los filtros para diseños 758, disponibles en la barra de herramientas 758, puede mostrar y ocultar cada tipo de plantilla en la vista **Diseño**.

6.4.1 Plantilla principal

La ***plantilla principal*** 32 determina la estructura del resultado de la transformación. Esto significa que el orden de los componentes de la plantilla principal es el orden de los componentes de la vista Authentic y de los resultados y documentos de salida.

En argot de programación, esto se conoce como *procesamiento de procedimientos*. El procesamiento comienza al principio de la plantilla y continúa de forma secuencial hasta el final. Durante este proceso, se procesan uno a uno los nodos del documento XML. Las plantillas que procesan estos nodos se llaman *plantillas locales* 32. Tras procesar una plantilla local, el procesador pasa al componente siguiente de la plantilla principal y así sucesivamente. De vez en cuando, un nodo hará referencia a una plantilla global 31 que debe utilizarse para procesar dicho nodo. En estos casos, tras ejecutar la plantilla global para ese nodo, el procesador vuelve a la posición de la plantilla principal desde donde se bifurcó y continúa en secuencia a partir del siguiente componente.

El punto de entrada para la plantilla principal es el nodo de documento 30 del esquema. StyleVision ofrece la opción de seleccionar varios documentos raíz (elementos de documento 30). Esto significa que dentro de la plantilla principal puede haber plantillas locales 32 para cada uno de los elementos de documento activos. La plantilla que se ejecuta durante el procesamiento es la del elemento que es el elemento de documento del documento XML de instancia que se está procesando.

6.4.2 Plantillas globales

En StyleVision puede definir una ***plantilla global*** 31 para cualquier nodo o tipo del esquema o para cualquier nodo especificado en un patrón XPath.

Una plantilla global especifica las instrucciones para el nodo o tipo seleccionados y se invoca por medio de una llamada desde la plantilla principal 32, desde fragmentos de diseño 278 o desde otras plantillas globales. El modelo de procesamiento es similar al de los lenguajes de programación declarativos, en el sentido de que se define una sola plantilla que se invoca varias veces. De este modo se puede usar una y otra vez una sola definición. Las plantillas globales se invocan en dos situaciones:

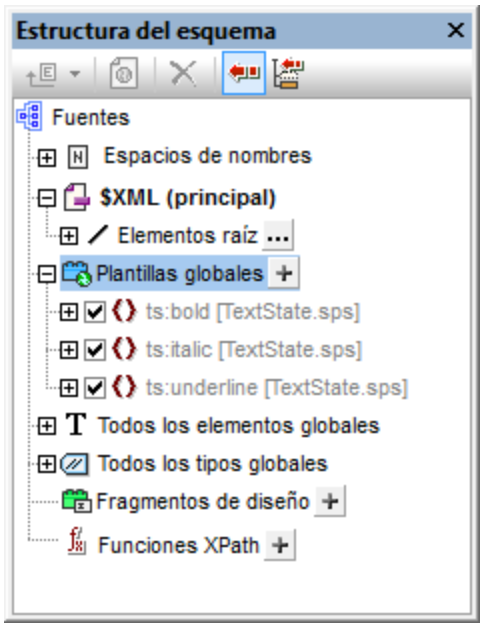
- Cuando un nodo o tipo de la plantilla principal 32 se configuró para hacer referencia a su propia plantilla global (clic con el botón derecho en el componente en el diseño y después en **Convertir en plantilla global**).
- Cuando se inserta un marcador de posición `(contenido)` 135 o `(resto-de-contenido)` 141 dentro de un elemento o tipo de una plantilla local 32 y el resto del contenido de dicho elemento o tipo incluye un nodo o tipo para el que existe una plantilla global 31.

Las plantillas globales son muy útiles si un nodo (o tipo) aparece dentro de varios elementos o en varias posiciones y para todas estas instancias del nodo (o tipo) basta un solo conjunto de instrucciones. Por ejemplo, imagine que el elemento `para` siempre debe tener el mismo formato, independientemente de si aparece en un elemento `chapter`, `section`, `appendix` o `blockquote`. En este caso lo más eficaz sería definir una plantilla global para el elemento `para` y después asegurarnos de que en la plantilla principal 32 se procese correctamente la plantilla global del elemento `para` (por ejemplo, incluyendo `//chapter/para` en la plantilla principal y especificando que `para` hace referencia a su plantilla global. O incluyendo `//chapter/title` y después `(contenido)` 135 o `(resto-de-contenido)` 141 de modo que el resto del contenido del elemento `chapter` se procese con las plantillas globales y predeterminadas disponibles). Además, puede definir una plantilla global para un tipo complejo (uno que define un modelo de dirección, por ejemplo) o incluso para un tipo simple (`xs:decimal`, por ejemplo). En estos casos, todas las instancias del tipo (complejo o simple) que invocan a la plantilla global de dicho tipo se procesarán de acuerdo con las reglas de la plantilla global.

Crear una plantilla global

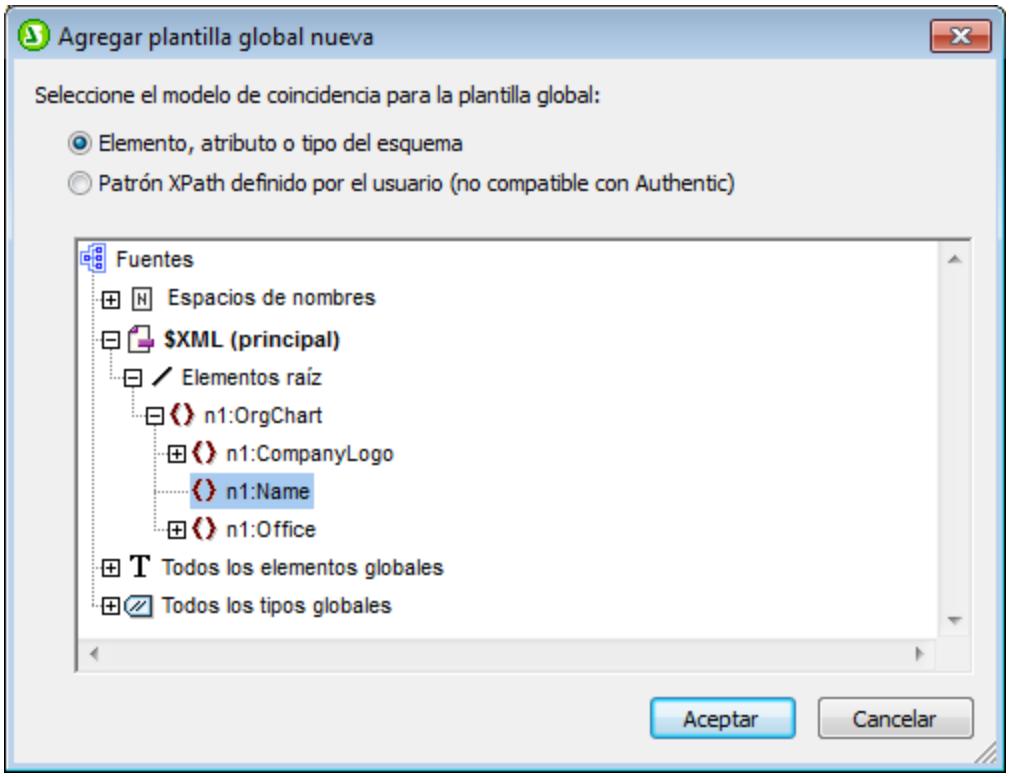
Puede crear plantillas globales para cualquier nodo/tipo del esquema o para cualquier nodo especificado en un patrón XPath. Las plantillas globales se crean en la ventana Estructura del esquema (*imagen siguiente*).

 © 2016-2022 Altova GmbH



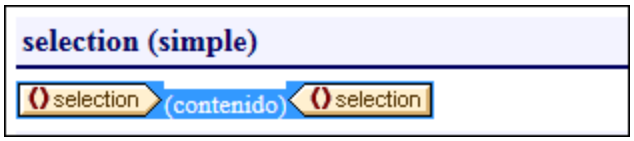
Las plantillas globales se pueden crear de varias maneras:

- *En la ventana Estructura del esquema*: haga clic en el icono **Agregar plantilla global nueva** situado a la derecha de la entrada `Plantillas globales` (*imagen anterior*). Aparece un diálogo llamado "Agregar plantilla global nueva" (*imagen siguiente*). En este cuadro de diálogo puede seleccionar un elemento/atributo/tipo del esquema o introducir un patrón XPath. (Recuerde que las plantillas globales creadas para nodos seleccionados con un patrón XPath no son compatibles con la vista Authentic). Para terminar haga clic en **Aceptar**. La plantilla se crea y se anexa a la lista de plantillas en la vista **Diseño**, donde se puede editar. En la ventana Estructura del esquema el nodo/tipo del esquema aparece marcado y con el signo +.



- *En la ventana Estructura del esquema:* haga clic con el botón derecho en el nodo/tipo del esquema (bajo la entrada `Elementos raíz`, `Todos los elementos globales` o `Todos los tipos globales`) y después seleccione el comando **Convertir en plantilla global o quitarla**. La plantilla se crea y se anexa a las plantillas ya existentes en la vista **Diseño**, donde se puede editar. En la ventana Estructura del esquema, el nodo/tipo del esquema aparece marcado y con el signo +.
- *En la vista **Diseño** a partir de otras plantillas de la plantilla principal*: haga clic con el botón derecho en la plantilla (en la vista **Diseño** o en la ventana Estructura del esquema) y seleccione el comando **Convertir en plantilla global** del menú contextual. Como resultado se crea una plantilla global a partir de la plantilla seleccionada (se anexa a las demás plantillas en la vista **Diseño**) y la plantilla está definida automáticamente para usar esta plantilla global (siga leyendo para obtener más información).

En la vista **Diseño** la plantilla global está debajo de la plantilla principal y se señala con una barra de color malva que contiene el nombre del nodo para el que se creó la plantilla global seguido del tipo (simple o complejo). A continuación puede ver la representación gráfica de una plantilla global



Note that the processing of the global template is user-defined and could include bot static and dynamic components, as well as the whole range of processing options available for processing of the main template.

Usar una plantilla global

Una vez creada, la plantilla global se utiliza cada vez que se inserte en el documento un nodo con el mismo nombre completo que la plantilla global. Cuando coloque el nodo en el diseño, seleccione el comando **Utilizar plantilla global** del menú contextual que aparece. Si en el diseño hay una plantilla local con un nombre completo idéntico al de un nodo para el que existe una plantilla global, puede utilizarse la plantilla global en lugar de la plantilla local. Para usar una plantilla global en lugar de una plantilla local, haga clic con el botón derecho en la plantilla local en la vista **Diseño** y seleccione el comando **Utilizar plantilla global**. Cuando se utiliza una plantilla global, se invocan sus instrucciones de procesamiento y la plantilla local las utiliza en tiempo de ejecución.

Cada vez que se utiliza la plantilla global en el diseño, se puede crear un patrón XPath en la plantilla global para filtrar el conjunto de nodos al que apunta. Para crear un filtro así, haga clic con el botón derecho en la etiqueta de la plantilla global en la vista **Diseño** y seleccione el comando **Editar filtro de XPath** 276 del menú contextual. Al hacer clic en este comando se abre el diálogo Editar expresión XPath 729, donde puede introducir la expresión necesaria.

Plantillas globales recursivas

Las plantillas globales pueden ser recursivas, es decir, pueden llamarse a sí mismas. Sin embargo, para evitar bucles sin fin en la vista Authentic, puede establecer una propiedad para limitar la profundidad de la llamada. Se trata de la propiedad *Profundidad máxima de llamada a plantillas*, disponible en la pestaña *Authentic* del cuadro de diálogo "Propiedades" del archivo SPS (**Archivo | Propiedades** 787). Esta propiedad define el máximo de llamadas a plantillas que se pueden hacer de forma recursiva durante el procesamiento para la vista Authentic. Si el número de llamadas a plantillas supera el número especificado en esta propiedad, StyleVision devuelve un error.

Crear una copia local de la plantilla global

Una vez creada la plantilla global, sus instrucciones de procesamiento se pueden copiar directamente en una plantilla local de la plantilla principal que tenga el mismo nombre completo. Esto se consigue haciendo clic con el botón derecho en la plantilla local y seleccione el comando **Crear copia local de la plantilla global**. Crear una copia local no es lo mismo que usar la plantilla global (en tiempo de ejecución) porque las instrucciones de procesamiento se copian solo de manera puntual. La plantilla global no tiene más influencia en la plantilla local. A partir de ese momento, la plantilla global o la plantilla local (o ambas) se pueden modificar independientemente la una de la otra. Por otro lado, si se especifica que una plantilla local debe *utilizar* una plantilla global (en tiempo de ejecución), todos los cambios realizados en la plantilla global se reflejarán en la plantilla local en tiempo de ejecución.

Activar y desactivar las plantillas globales

Para activar una plantilla global marque su casilla en la ventana Estructura del esquema. Para desactivarla vuelva a hacer clic en la casilla. Si una plantilla global está activada (opción predeterminada cuando se crea la plantilla global), se genera en la hoja de estilos XSLT. Si está desactivada, no se genera en la hoja de estilos XSLT pero se guarda en el diseño SPS. Las plantillas locales que usen una plantilla global desactivada no pueden hacer referencia a la plantilla global porque está desactivada y, por tanto, recurren a las plantillas predeterminadas de XSLT (que tienen el efecto conjunto de generar el contenido de los nodos de texto descendientes).

Las ventajas de poder activar/desactivar las plantillas globales son: (i) no es necesario borrar las plantillas globales que no se necesitan en un momento dado, basta con desactivarlas y volver a activarlas cuando se

necesiten y (ii) si los nombres de las plantillas de hojas de estilos importadas entran en conflicto con los de plantillas globales existentes, basta con desactivar la plantilla global.

Quitar una plantilla global

Para quitar una plantilla global haga clic con el botón derecho en la plantilla global (bien en la vista **Diseño**, bien en la ventana Estructura del esquema) y seleccione el comando **Convertir en plantilla global o quitarla** del menú contextual.

Plantillas globales simples y complejas

Hay dos tipos de plantillas globales: simples y complejas. Las plantillas globales complejas están disponibles por motivos de compatibilidad con versiones previas. Es decir, si una plantilla global de un diseño SPS creado con una versión de StyleVision anterior a la versión 2006 contiene una tabla o una lista y el diseño se abre en StyleVision versión 2006 o superior, la plantilla global se abre como plantilla global compleja.

Las plantillas globales complejas se diferencian de las plantillas globales simples en la forma en que procesan el nodo para el que se creó la plantilla global. Cuando se encuentra la primera instancia del nodo en el documento, la plantilla global compleja procesa las demás instancias del nodo inmediatamente después. En cambio, la plantilla global simple procesa cada instancia del nodo solamente cuando se encuentra dicha instancia del nodo.

Es importante tener en cuenta que una plantilla global simple se convierte automáticamente en una plantilla global compleja si se crea un formato predefinido 375 o una línea nueva **alrededor** del nodo de elemento para el que se creó la plantilla global. Como resultado la plantilla tendrá el comportamiento de una plantilla global compleja (*ver párrafo anterior*). Si prefiere que la plantilla global sea de tipo simple, elimine el formato predefinido 375 (arrastre el nodo fuera del formato predefinido y después elimine el formato predefinido) o elimine la línea nueva (elimine el componente en la ventana Estructura del diseño 52) según corresponda. Para evitar la conversión automática de las plantillas globales simples en plantillas globales complejas, asegúrese de añadir el formato predefinido 375 o la línea nueva dentro de las etiquetas del elemento para el que se creó la plantilla global.

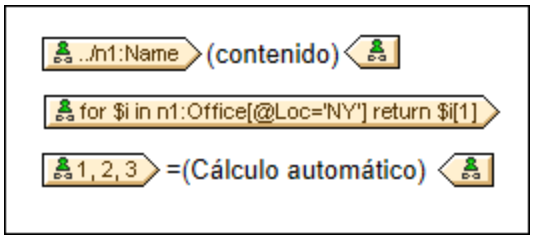
Plantillas globales en archivos SPS modulares

Cuando se añade un módulo SPS dentro de un archivo SPS 251, las plantillas del módulo agregado están disponibles y se pueden usar en el archivo SPS modular. Para más información consulte la sección Archivos SPS modulares 251.

6.4.3 Plantillas definidas por el usuario

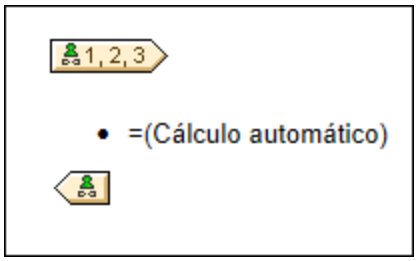
Las plantillas definidas por el usuario son plantillas para elementos generados por una expresión XPath creada por el usuario. Estos elementos pueden ser valores atómicos o nodos. En la imagen siguiente puede ver tres plantillas definidas por el usuario. Observe que las etiquetas de este tipo de plantillas tienen un icono en forma de peón verde. Las plantillas definidas por el usuario son muy prácticas porque ofrecen una gran flexibilidad. Sin embargo, no olvide que el contenido generado por plantillas definidas por el usuario **no se puede editar en la vista Authentic**.

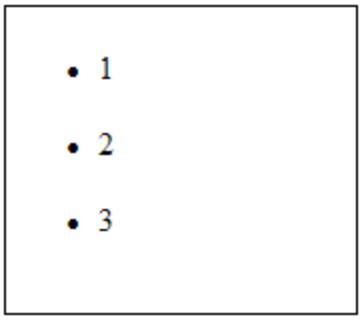
 © 2016-2022 Altova GmbH



La expresión XPath de las plantillas definidas por el usuario de la imagen hace esto:

- Selecciona un nodo de un esquema fuente. Con una expresión XPath puede alcanzar cualquier nodo de cualquier esquema fuente desde cualquier nodo de contexto. Si StyleVision puede apuntar inequívocamente al nodo especificado, la plantilla definida por el usuario se convierte automáticamente en una plantilla normal, lo cual habilita la edición en la vista Authentic. Si es una plantilla definida por el usuario, el icono de la etiqueta tendrá forma de peón verde.
- Selecciona un nodo que cumple con la condición especificada por la construcción `for` de XPath 2.0 y XPath 3.0. Dichas plantillas no pueden convertirse en plantillas normales porque la construcción `for` no permite a StyleVision resolver el destino inequívocamente con la información de esquema que tiene a su disposición.
- Selecciona una secuencia de valores atómicos `{1, 2, 3}`. Si está permitido crear una plantilla para un valor atómico, no podrá usar el marcador de posición `(contenido)` dentro de dicha plantilla. Esto se debe a que la instrucción `xsl:apply-templates` (que es la que genera el marcador de posición `(contenido)`) solamente se puede aplicar a elementos de nodo (no a valores atómicos). Sin embargo, lo que puede hacer es usar un cálculo automático junto con otro elemento de diseño como una lista. Por ejemplo, la plantilla definida por el usuario que aparece a continuación daría como resultado el texto de la derecha.





Nota: si el archivo SPS utiliza el modo XSLT 1.0, entonces la expresión XPath utilizada debe devolver un conjunto de nodos. De lo contrario se emite un error.

Ventaja de usar XPath para seleccionar nodos

La ventaja de usar expresiones XPath para seleccionar nodos del esquema (o lo que es lo mismo, plantillas definidas por el usuario) es que la potencia del mecanismo de selección de rutas de XPath se puede aprovechar para seleccionar cualquier nodo o secuencia de elementos, así como para filtrar o establecer condiciones para la selección de nodos. Como resultado, podrá apuntar a ciertos nodos del documento XML para una plantilla determinada. Por ejemplo, la expresión XPath `//Office/Department[@Location="NY"]` seleccionará solamente los nodos `Department` cuyo atributo `Location` tenga el valor `NY`. Para más ejemplos consulte el resto de este apartado.

Nota: si una expresión XPath contiene varios pasos de ruta de acceso, entonces la posición de los

paréntesis desempeña un papel importante (sobre todo en el caso de las agrupaciones y ordenaciones). Por ejemplo, la expresión XPath `/Org/Office/Dept` no se procesa igual que la expresión `(/Org/Office/Dept)`. En el caso de la primera expresión, sin paréntesis, el procesador recorre cada paso de la ruta. Para la segunda expresión, con paréntesis, todos los elementos `Dept` de todos los elementos `Office` se devuelven en un solo conjunto de nodos.

Paréntesis	Mecanismo XSLT subyacente	Efecto

No	`<xsl:for-each select="Org">`
	`  <xsl:for-each select="Office">`
	`  <xsl:for-each select="Dept">`
	`  ...`
	`  </xsl:for-each>`
	`  </xsl:for-each>`
	`</xsl:for-each>`
		Cada elemento `Office` tiene sus propios `Dept`. Puede realizar agrupaciones y ordenaciones dentro de cada `Office`.

Sí	`<xsl:for-each`
	`select="/Org/Office/Dept">`
	`  ...`
	`</xsl:for-each>`
		Los elementos `Dept` abarcan todos los elementos `Office` y todos los elementos `Org`.

Tenga en cuesta esta importante diferencia a la hora de realizar agrupaciones y ordenaciones.

Insertar una plantilla definida por el usuario

Para insertar una plantilla definida por el usuario:

1. Haga clic en el icono **Insertar plantilla definida por el usuario** de la barra de herramientas "Insertar elementos de diseño" y después en la posición del diseño donde desea insertar la plantilla. También puede hacer clic con el botón derecho en el diseño y seleccionar el comando **Insertar plantilla definida por el usuario** del menú contextual.
2. Se abre el cuadro de diálogo Editar expresión XPath 729. Escriba la expresión XPath que desea usar y haga clic en **Aceptar**. Observe que el nodo de contexto de la expresión XPath será el nodo donde hizo clic. Se crea una plantilla nodo vacía. A veces se crea un nodo unido. Cuando un nodo está unido, los nodos de instancia a los que se apunta están seleccionados como si estuvieran en el mismo nivel. Cuando el nodo no está unido (sino dividido en varios niveles jerárquicos) la selección del nodo se hace recorriendo cada nodo de instancia de cada nivel. El conjunto de nodos que la expresión devuelve en ambos casos (unido y dividido) es el mismo, a no ser que se especifique un criterio de agrupación o de ordenación. Para más información consulte el apartado Operaciones con plantillas nodo 274.

Editar el modelo de coincidencia para la plantilla

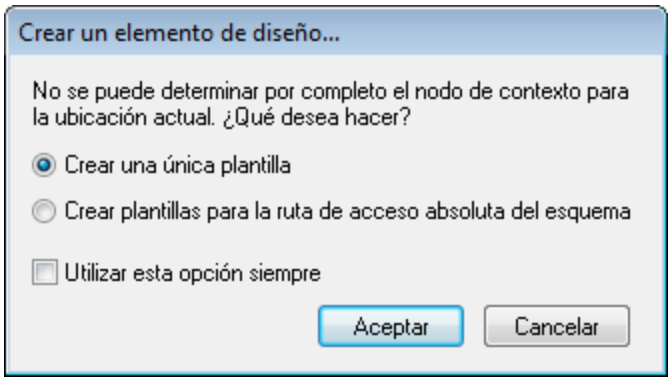
La selección de nodos de cualquier plantilla nodo (normal o definida por el usuario) se puede cambiar usando una expresión XPath y seleccionando otro modelo de coincidencia. Para editar el modelo de coincidencia de una plantilla nodo, haga clic con el botón derecho en la plantilla nodo y seleccione **Editar modelo de coincidencia** en el menú contextual. Se abre el cuadro de diálogo "Editar expresión XPath", donde puede introducir la expresión XPath para seleccionar otro nodo. Después haga clic en **Aceptar**.

Agregar nodos a plantillas definidas por el usuario

Si añade un nodo de esquema a una plantilla definida por el usuario creada para un nodo o secuencia que no se puede colocar en el contexto del esquema fuente, StyleVision no puede conocer el contexto para el nodo nuevo. En este caso aparece un aviso (*imagen siguiente*) preguntando cómo se debe hacer referencia al nuevo

 © 2016-2022 Altova GmbH

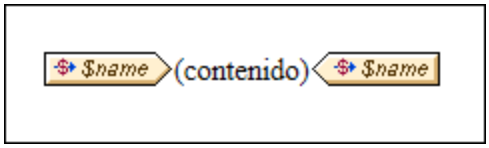
nodo: (i) por su nombre (con una ruta de acceso relativa) o (ii) por una ruta de acceso completa empezando por la raíz del esquema fuente.



Este aviso aparece siempre por defecto, pero se puede deshabilitar en la pestaña *Diseño* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**[894]).

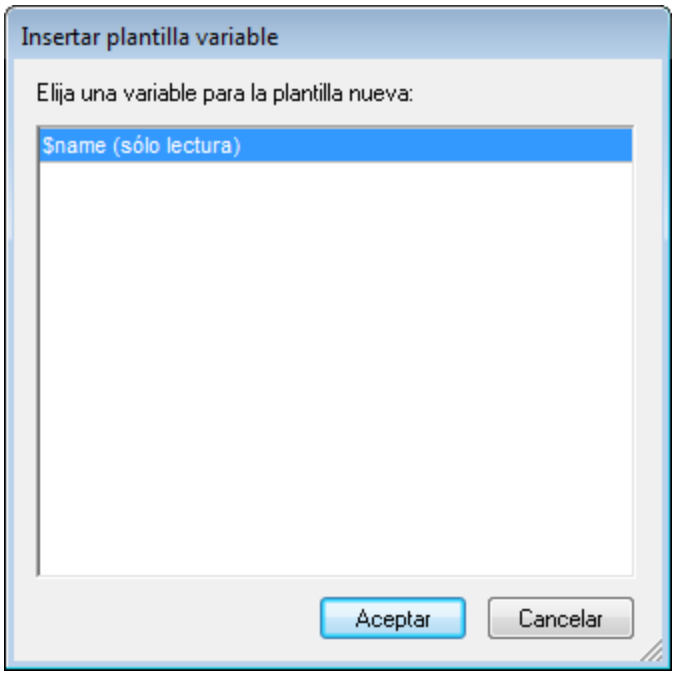
6.4.4 Plantillas variables

Una plantilla variable es una plantilla que apunta a una variable y que por defecto genera su contenido. Las plantillas variables se insertan con los comandos **Insertar | Insertar plantilla variable** o en **Incluir en | Plantilla variable**. Ambos comandos insertan una plantilla para una variable definida en el SPS. La plantilla variable (*imagen siguiente*) contiene por defecto el marcador de posición `(contenido)` y este marcador sirve para generar el contenido de la variable. Si quiere, puede insertar contenido adicional (estático o dinámico) dentro de la plantilla variable o modificarla como si fuera una plantilla normal y corriente. En su representación gráfica en la vista **Diseño** las etiquetas de la plantilla variable incluyen el símbolo del dólar (*imagen siguiente*).



Para insertar una plantilla variable:

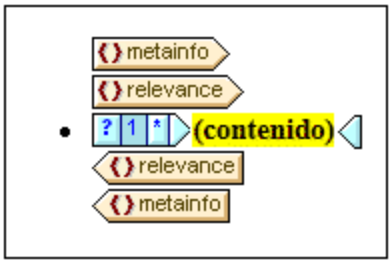
1. En la vista **Diseño** ponga el cursor en la posición donde desea insertar la plantilla variable.
2. Haga clic en el comando de menú **Insertar | Insertar plantilla variable**. Aparece el cuadro de diálogo "Insertar plantilla variable" (*imagen siguiente*).



3. El cuadro de diálogo enumera todas las variables y los parámetros declarados por el usuario [327] y definidos en el archivo SPS. Seleccione la variable para la que desea crear la plantilla.
4. Para terminar haga clic en **Aceptar**.

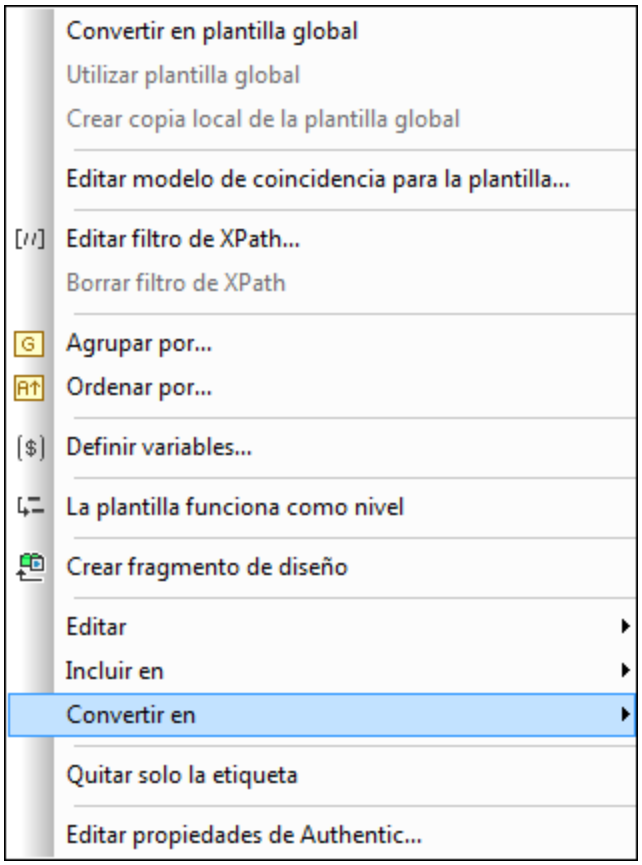
6.4.5 Operaciones con plantillas nodo

Una plantilla nodo es una plantilla del diseño que especifica cómo se debe procesar un nodo. En el diseño las plantillas nodo tienen etiquetas de apertura y cierre de color beige (*imagen siguiente*). El tipo de nodo se conoce por el símbolo que aparece en las pestañas del nodo (p. ej. llaves para los nodos de elemento y el signo igual para los nodos de atributo). En la imagen siguiente, por ejemplo, hay dos plantillas nodo de tipo elemento: `metainfo` y `relevance`. Para más información consulte el apartado *Nodos del documento XML* [724].



Las operaciones que se pueden realizar en las plantillas nodo están disponibles en el menú contextual (*imagen siguiente*) de la plantilla. Para abrir este menú contextual, haga clic con el botón derecho en la etiqueta de apertura o de cierre.

 © 2016-2022 Altova GmbH



A continuación describimos los comandos de este menú contextual:

- Plantillas globales 275
- Modelo de coincidencia para la plantilla 276
- Filtros de XPath 276
- Agrupar por, Ordenar por, Definir variables, La plantilla funciona como nivel 277
- Crear fragmento de diseño 277
- Quitar solo la etiqueta 277
- Editar, Incluir en, Convertir en 277
- Propiedades de Authentic 277

En el caso de algunas plantillas es posible que no estén disponibles todos los comandos y algunos aparezcan atenuados en color gris.

Plantillas globales: convertir en, utilizar y crear copia local

Una plantilla nodo de la plantilla principal se puede convertir en plantilla global o asociar a una plantilla global con estos comandos:

- **Convertir en plantilla global**: esta opción está disponible si la plantilla nodo representa un elemento definido como elemento global en el esquema. Al hacer clic en el comando se crea una plantilla global a partir de la plantilla nodo. La plantilla nodo de la plantilla principal usará esta plantilla global y sus etiquetas aparecerán en color gris (lo cual indica que utiliza la plantilla global).

- **Utilizar plantilla global**: si se definió una plantilla global con el mismo nombre completo que la plantilla nodo, la plantilla nodo usará el procesamiento de la plantilla global. Las etiquetas de la plantilla nodo aparecen en color gris.
- **Crear copia local de la plantilla global**: las instrucciones de procesamiento de una plantilla global cuyo nombre completo sea idéntico al de la plantilla nodo se copian físicamente en la plantilla nodo. La plantilla nodo es independiente de la plantilla global. En adelante la plantilla nodo y la plantilla global se pueden modificar independientemente una de la otra. Como la plantilla global no hace referencia a una plantilla global, sus etiquetas no cambian de color (siguen de color beige).

Para más información consulte el apartado Plantillas globales 266.

Editar modelo de coincidencia para la plantilla

El nodo para el que se creó la plantilla se puede cambiar con este comando. Al hacer clic en el comando se abre el cuadro de diálogo Editar expresión XPath 729, donde puede escribir la expresión XPath que selecciona otro nodo del esquema. También puede introducir una expresión XPath para convertir la plantilla en una plantilla definida por el usuario 148.

Editar / borrar filtro de XPath

Un filtro de XPath permite limitar el conjunto de nodos en el que se aplica una plantilla nodo. Los filtros de XPath también se pueden aplicar a plantillas globales 266.

Las plantillas nodo se aplican por defecto a los nodos (elementos/atributos) equivalentes al nodo para el que se creó la plantilla nodo (que tenga el mismo nombre y aparezca en la misma posición de la jerarquía del esquema). Por ejemplo, una plantilla nodo para el nodo `/Personnel/Office` seleccionará todos los elementos `/Personnel/Office`. Pero si creamos un filtro de XPath con la expresión `1` en el elemento `Office` (haciendo clic en el elemento `Office` y editando su filtro de XPath), se añade una expresión de predicado al elemento `Office`, de modo que la expresión XPath completa sería: `/Personnel/Office[1]`. Esta expresión XPath seleccionaría el primer secundario `Office` del elemento `Personnel` y, en la práctica, filtra los demás elementos `Office` y los deja fuera.

Los filtros se pueden añadir en cualquier plantilla nodo y en varias plantillas nodo del diseño. Esto permite tener expresiones XPath como: `/Personnel/Office[@country='US']/Person[Title='Manager']`, que seleccionaría todos los managers de las oficinas de EE UU de la compañía. En este ejemplo se creó un filtro en la plantilla nodo `Office` y otro en la plantilla `Person`. De modo que a cada instancia de una plantilla global, se le puede aplicar un filtro de XPath para limitar el conjunto de nodos al que se apunta.

Para añadir un filtro de XPath a una plantilla nodo, haga clic con el botón derecho en la plantilla y seleccione el comando **Editar filtro de XPath**. En el cuadro de diálogo introduzca una expresión XPath sin comillas, sin corchetes ni otros delimitadores. Puede introducir cualquier expresión XPath válida. Por ejemplo:

- `1`
- `@country='US'`
- `Title='Manager'`

Tras crear el filtro de XPath, en la etiqueta de apertura de la plantilla nodo aparece un símbolo en forma de filtro. En la imagen siguiente, por ejemplo, la plantilla nodo `synopsis` tiene un filtro.

Nota: cada plantilla nodo admite un filtro de XPath.

Agrupar por, Ordenar por, Definir variables, La plantilla sirve como nivel

Los mecanismos asociados a estos comandos se describen en las correspondientes secciones de esta documentación (haga clic en los enlaces):

- El comando **Agrupar por** sirve para agrupar las instancias del nodo representado por la plantilla nodo seleccionada. El mecanismo de agrupación se describe en la sección Agrupación 312.
- El comando **Ordenar por** sirve para ordenar las instancias del nodo representado por la plantilla nodo seleccionada. El mecanismo de ordenación se describe en la sección Ordenación 321.
- El comando **Definir variables** sirve para definir variables que están en el ámbito de la plantilla nodo seleccionada. En la sección Variables 332 se explica cómo trabajar con variables.
- El comando **La plantilla sirve como nivel** es un comando de conmutación que crea o quita un nivel en la plantilla nodo. Se pueden especificar varios niveles para estructurar el documento. Con esta estructura del documento se puede generar una tabla de contenido (TDC), numeración automática y referencias de texto. Estas características se describen en la sección Tabla de contenido (TDC) y Referencias 337.

Crear fragmento de diseño

Crea una plantilla de fragmento de diseño a partir de la plantilla seleccionada. El fragmento de diseño resultante se añade a los fragmentos de diseño situados al final del diseño y también en las ventanas Estructura del diseño y Estructura del esquema. El fragmento de diseño se aplica en la posición del diseño donde se creó.

Quitar solo la etiqueta

Este comando elimina solamente la etiqueta de formato o de la plantilla seleccionada. No elimina las etiquetas de formato descendientes ni los nodos descendientes. Este comando es muy útil si quiere eliminar una etiqueta de formato o una etiqueta de elemento primario sin eliminar todo lo que está dentro de la etiqueta (que es lo que pasaría con la operación **Eliminar**). No obstante, recuerde que al quitar un elemento primario puede que los nodos descendientes del elemento eliminado dejen de ser válidos. Cuando esto pasa, los nodos no válidos aparecen tachados en color rojo.

Editar, Incluir en, Convertir en

- **Editar**: este comando abre un submenú con los comandos **Cortar**, **Copiar**, **Pegar** y **Eliminar** de Windows.
- **Incluir en**: puede incluir la plantilla nodo dentro de estos componentes de diseño: párrafo 135, párrafo especial 137, numeración y viñetas 178, hipervínculo 364, condición 303, marcador TDC y nivel TDC 337.
- **Convertir en**: este comando sirve para (i) cambiar el nodo al que se aplica la plantilla o (ii) cambiar la forma en que se crea el nodo en el diseño. Esto se describe detalladamente en la sección Función Convertir en 215.

Editar propiedades de Authentic

Al hacer clic en este comando se abre la ventana "Propiedades" donde puede definir ciertas propiedades del nodo con instancias en la vista Authentic. Las propiedades de Authentic se describen en la sección Propiedades de nodos Authentic 462.

6.4.6 Fragmentos de diseño

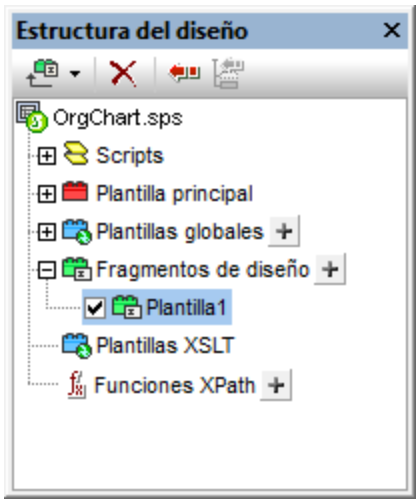
Los fragmentos de diseño son muy útiles para crear partes y volver a utilizarlas en otras posiciones del documento (igual que se reutilizan las funciones, por ejemplo). El mecanismo de uso es el siguiente:

1. Crear el fragmento de diseño en el archivo SPS 278.
2. Insertar el contenido del fragmento de diseño 279.
3. Insertar el fragmento de diseño en una posición de una plantilla 280.

Crear un fragmento de diseño

Siga estas instrucciones para crear un fragmento de diseño:

1. En la ventana Estructura del diseño o Estructura del esquema haga clic en el icono **Agregar fragmento de diseño nuevo**, situado a la derecha de la entrada **`Fragmentos del diseño`** (*imagen siguiente*). Esto añade un fragmento de diseño a la lista **`Fragmentos del diseño`** de la ventana.



   Observe que se crea una plantilla de fragmento de diseño en la vista **Diseño**. Esta plantilla se anexa a las plantillas que ya existen en el diseño y puede reconocerse por su encabezado verde. Si quiere ver solamente los fragmentos de diseño que están en el diseño, oculte la plantilla principal y las plantillas globales con los comandos **Mostrar/Ocultar** 758 de la barra de herramientas Filtro para diseños 758 de StyleVision. Además puede acceder rápidamente a los fragmentos de diseño desde la ventana Estructura del esquema.
2. Haga doble clic en el componente fragmento de diseño (en la ventana Estructura del esquema o Estructura del diseño), escriba un nombre nuevo y después pulse **Entrar**. El nuevo nombre del fragmento aparece en la ventana Estructura del diseño (*imagen siguiente*) y en la plantilla en la vista **Diseño**.

 © 2016-2022 Altova GmbH

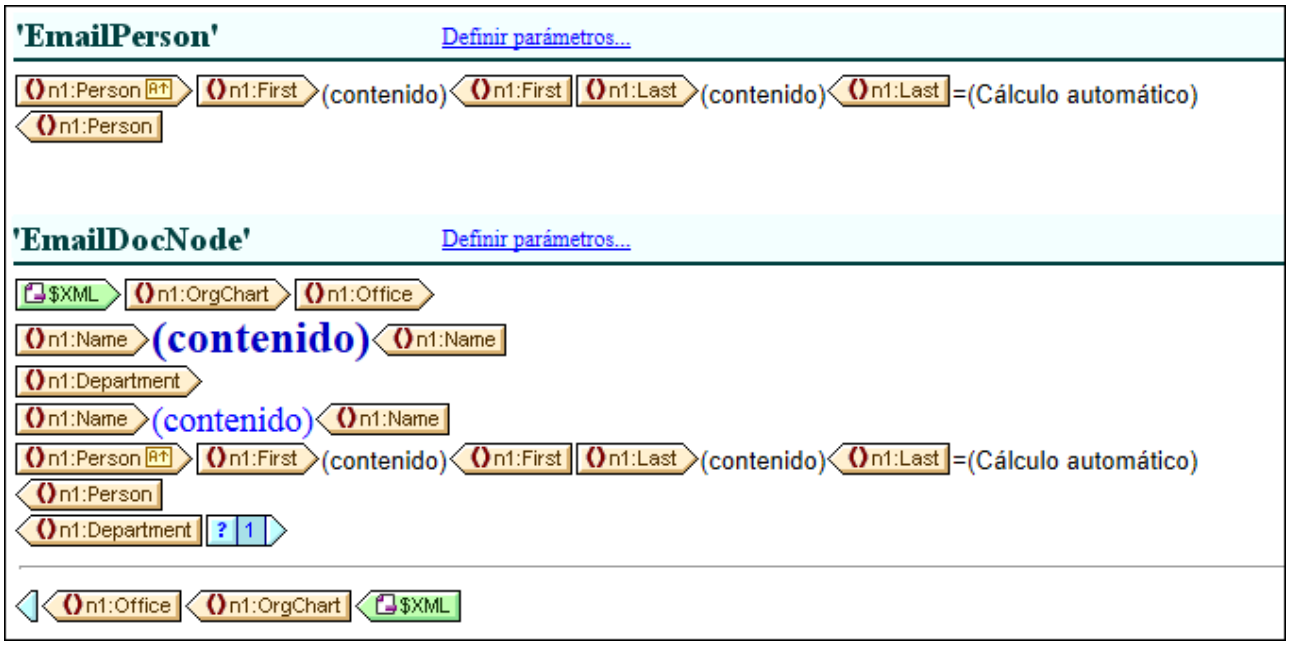
3. En la vista **Diseño** cree el contenido de la plantilla del fragmento de diseño (*ver subapartado siguiente*).

Nota: si quiere crear un fragmento de diseño a partir de una plantilla ya existente, haga clic con el botón derecho en la plantilla y seleccione el comando Crear fragmento de diseño en el menú contextual. Esto crea una plantilla de fragmento de diseño en esa posición del diseño. La plantilla de fragmento de diseño también se anexa a los fragmentos de diseño situados al final del diseño y aparece en las ventanas Estructura del diseño y Estructura del esquema. Si se crea de esta forma, el fragmento de diseño se aplica directamente en la posición donde se creó y no hace falta arrastrarlo desde la ventana Estructura del diseño o Estructura del esquema 280.

Insertar el contenido del fragmento de diseño

El contenido de la plantilla de fragmento de diseño se crea igual que en las demás plantillas 134. Para insertar contenido estático, ponga el cursor dentro del fragmento de diseño e inserte el contenido. Para insertar contenido dinámico, arrastre el nodo de esquema pertinente hasta el fragmento de diseño.

Cuando arrastre un nodo del esquema hasta el fragmento de diseño, hay una gran diferencia entre arrastrar el nodo desde (i) la lista **`Elementos globales`** y arrastrarlo desde (ii) la lista **`Elementos raíz`**. Si el nodo se arrastra desde la lista **`Elementos globales`**, se crea sin sus elementos antecesores (p. ej. en la imagen siguiente, el fragmento de diseño **EmailPerson**) y, por tanto, cuando se use en una plantilla deberá utilizarse dentro del contexto de su elemento primario. Por el contrario, si el nodo se arrastra desde la lista **`Elementos raíz`**, se crea dentro de una estructura que empieza con el nodo de documento (p. ej. en la imagen siguiente, el fragmento de diseño **EmailDocNode**) y, por tanto, puede utilizarse en cualquier parte de la plantilla.



La imagen anterior muestra dos fragmentos de diseño que generan exactamente el mismo resultado para el elemento `Person`. En el fragmento de diseño **EmailPerson**, el nodo `Person` se arrastró desde la lista **`Elementos raíz`** y se creó con una ruta de acceso absoluta (desde `$XML`, que es el nodo de documento).

Cuando inserte este fragmento de diseño en la plantilla principal, asegúrese de llamar a la plantilla **EmailPerson** desde un contexto que sea el primario del nodo `Person`. Para familiarizarse con el uso de los fragmentos de diseño, consulte y experimente con el archivo `Email.sps`, que está en la carpeta (Mis) Documentos [34], `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\DesignFragments`.

También puede definir un parámetro con un valor predeterminado en el fragmento de diseño. Al parámetro se le puede asignar un valor distinto en cada instancia del fragmento de diseño. Para más información consulte el apartado Parámetros para fragmentos de diseño [329].

Cuando termine de diseñar el documento SPS, observe que los componentes del diseño también se representan de forma gráfica en la ventana Estructura del diseño.

Insertar un fragmento de diseño en una plantilla

Para insertar un fragmento de diseño, arrástrelo desde la ventana Estructura del diseño o Estructura del esquema hasta la posición deseada del diseño. Coloque el fragmento de diseño en una posición que ofrezca el contexto adecuado. Si el contenido del fragmento de diseño se crea a partir de un elemento global, el contexto adecuado de la plantilla principal será el primario del nodo que se insertó en el fragmento de diseño. Consulte el subapartado anterior (Insertar el contenido del fragmento de diseño [279]) para obtener más información.

Otra opción es hacer clic con el botón derecho en la posición donde desea insertar el fragmento de diseño y hacer clic en **Insertar fragmento de diseño** en el menú contextual.

Nota: si en la plantilla principal se hace referencia a un fragmento de diseño y después se cambia el nombre de este fragmento de diseño, la referencia en la plantilla principal dejará de ser correcta y se emitirá un error XSLT. Para corregir esto, elimine la referencia original en la plantilla principal y cree una referencia nueva al fragmento de diseño.

 © 2016-2022 Altova GmbH

Fragmentos de diseño recursivos

Los fragmentos de diseño pueden ser recursivos, es decir, pueden llamarse a sí mismos. No obstante, para evitar bucles sin fin en la vista Authentic, puede establecer una propiedad para limitar la profundidad de la llamada. Esta propiedad (*Profundidad máxima de llamada a plantillas*) se configura en la pestaña *Authentic* del cuadro de diálogo "Propiedades" del archivo SPS (**Archivo | Propiedades** 787). Especifica el número máximo de llamadas a plantilla que se pueden hacer recursivamente durante el procesamiento para la vista Authentic. Si el número de llamadas a plantilla supera el número especificado en la propiedad *Profundidad máxima de llamada a plantillas*, StyleVision emite un error.

Eliminar un fragmento de diseño

Para eliminar un fragmento de diseño, selecciónelo en la ventana Estructura del esquema y haga clic en el icono **Eliminar** de la barra de herramientas.

Fragmentos de diseño en archivos SPS modulares

Cuando se añade un módulo SPS a un archivo SPS 251, los fragmentos de diseño del módulo agregado están a disposición del archivo SPS modular de destino. Para más información consulte la sección Archivos SPS modulares 251.

Archivo de ejemplo

En la carpeta (Mis) Documentos 34 encontrará un archivo SPS de ejemplo que utiliza fragmentos de diseño: `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\DesignFragments`.

6.5 Plantillas XSLT

En StyleVision puede importar archivos XSLT al diseño SPS y las plantillas XSLT de estos archivos XSLT estarán disponibles en forma de plantillas globales. Si durante el procesamiento del documento XML uno de los nodos XML coincide con un nodo de una plantilla XSLT importada, la plantilla XSLT importada se aplica al nodo. Si el archivo XSLT importado contiene plantillas con nombre, estas estarán disponibles y se podrán insertar en el diseño.

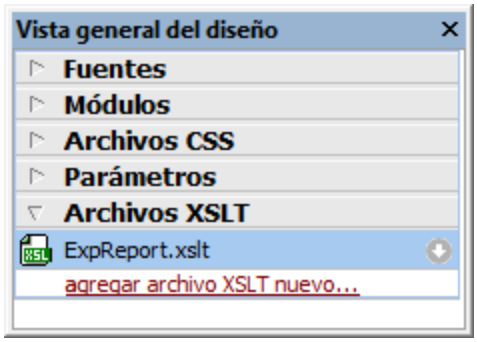
Nota:

- Las plantillas XSLT importadas no se pueden modificar en StyleVision.
- Las plantillas XSLT no son compatibles con la vista Authentic.

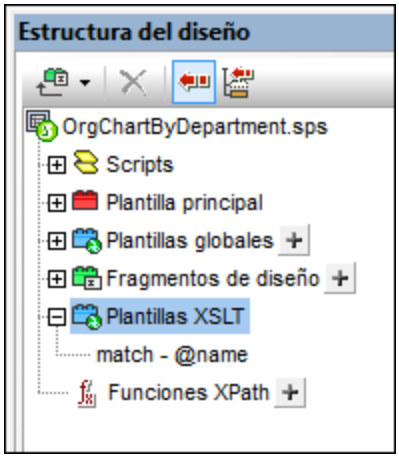
Importar el archivo XSLT

Siga estos pasos para importar un archivo XSLT:

1. En la ventana Vista general del diseño (*imagen siguiente*), haga clic en el vínculo `agregar archivo XSLT nuevo...`.



2. Aparece el cuadro de diálogo "Abrir". Busque el archivo XSLT necesario, selecciónelo y haga clic en **Abrir**. El archivo XSLT se importa en el diseño SPS. A la hoja de estilos XSLT se añade una instrucción `xsl:import` y en la ventana Estructura del diseño (*imagen siguiente*), bajo la entrada `Plantillas XSLT`, se enumeran las plantillas XSLT del archivo XSLT importado.



© 2016-2022 Altova GmbH

Las plantillas XSLT importadas pueden ser de dos tipos: (i) plantillas de coincidencia (que llevan la nota `match` en la Estructura del diseño) y (ii) plantillas con nombre (que llevan la nota `name` en la Estructura del diseño). Además, en la Estructura del diseño: (i) las plantillas con atributo `match` aparecen con el valor de su atributo `select` y (ii) las plantillas con nombre aparecen con el valor de su atributo `name`.

Plantillas de coincidencia

Las plantillas de coincidencia (`xsl:template match=`) se usan cuando, durante el procesamiento, una plantilla aplica plantillas a un nodo del documento XML y para ello se selecciona la plantilla de coincidencia que se debe aplicar. Esto ocurre cuando el nombre completo del nodo XML coincide con el nombre completo de la plantilla de coincidencia importada. Si en el diseño SPS se creó una plantilla global con el mismo nombre completo, esta plantilla global tiene prioridad sobre la plantilla importada. Si hay varios archivos XSLT importados, tiene prioridad el archivo XSLT que se importó más recientemente (que aparece más abajo en la lista **Archivos XSLT** de la ventana Vista general del diseño) y así sucesivamente.

Plantillas con nombre

Las plantillas con nombre (`xsl:template name=`) se puede arrastrar desde la ventana Estructura del diseño hasta cualquier posición del diseño. En la posición de destino elegida se crea un elemento `xsl:call-template` (*imagen siguiente*) que llama a la plantilla con nombre.

```
<xsl:call-template name="BoldItalic" />
```

Como resultado se implementa la plantilla con nombre en la correspondiente posición del diseño. Esto es muy útil si quiere insertar contenido que sea independiente tanto del documento XML de instancia como de la hoja de estilos XSLT.

6.6 Múltiples documentos de salida

Un diseño SPS de StyleVision puede generar varios documentos de salida diferentes a la vez: un documento de salida principal y varios documentos de salida adicionales. Esto puede ser muy útil si quiere crear resultados divididos en módulos. Para definir los diferentes documentos de salida en el diseño basta con insertar plantillas de documento nuevo (*imagen siguiente*). El contenido de cada documento de salida se coloca dentro de la correspondiente plantilla `Documento nuevo`.

Las plantillas `Documento nuevo` se pueden crear en cualquier parte del diseño lo cual permite definir módulos en cualquier nivel. Por ejemplo, un informe sobre las sucursales de una organización internacional puede tener varios documentos de salida, uno por cada nivel de la organización: (i) mundial, (ii) continental, (iii) nacional, (iv) regional y (v) por sucursal. Por ejemplo, puede generar un documento de salida para cada sucursal o un informe para todas las sucursales de cada país. Por cada documento de salida deberá insertar una plantilla `Documento nuevo` en el diseño. Para aprender a estructurar el documento correctamente consulte el apartado Plantillas de documento nuevo y estructura del diseño 286.

Esta característica de múltiples documentos de salida es compatible con los formatos de salida HTML, RTF y texto. En la vista Authentic, que es una vista de edición, el contenido aparece en un solo documento.

Esta sección se divide en varios apartados:

- Insertar una plantilla de documento nuevo 285: cómo crear una plantilla que genere un documento de salida diferente.
- Plantillas de documento nuevo y estructura del diseño 286: información sobre dónde se debe insertar la plantilla de documento nuevo en el diseño.
- URL de las plantillas de documento nuevo 286: criterios a la hora de especificar la URL de las plantillas de documento nuevo.
- Vista previa de archivos y documentos de salida 289: información sobre problemas relacionados con la ubicación de los documentos de salida.

© 2016-2022 Altova GmbH

- Propiedades y estilos de los documentos [291]: describe las propiedades de los documentos y las opciones de estilo.

6.6.1 Insertar una plantilla de documento nuevo

Hay dos maneras de insertar plantillas de documento nuevo en el diseño SPS:

- La plantilla de documento de salida nuevo **se puede insertar** directamente en cualquier posición del diseño. En este caso se añade primero la plantilla `Documento nuevo` y después su contenido. Para insertar una plantilla `Documento nuevo` ponga el cursor en la posición deseada y seleccione el comando de menú **Insertar | Insertar documento nuevo**. Otra opción es hacer clic con el botón derecho y después en **Insertar documento nuevo**.
- La otra manera de insertar la plantilla de documento nuevo es **incluir contenido** del diseño en una plantilla `Documento nuevo`. En este caso la plantilla `Documento nuevo` abarcará el contenido incluido, que se puede modificar o ampliar. Para incluir contenido en una plantilla `Documento nuevo`, resalte el contenido y después seleccione el comando de menú **Incluir en | Documento nuevo**. Otra opción es seleccionar el contenido, hacer clic con el botón derecho y después en **Incluir en | Documento nuevo**.

En la imagen siguiente puede ver un ejemplo de plantilla `Documento nuevo`.

En esta imagen puede ver que:

1. Las etiquetas de la plantilla `Documento nuevo` contienen la URL (ruta de acceso y nombre) del documento de salida que se generará. El sufijo del nombre de archivo se generará automáticamente según el tipo de archivo del formato de salida. Por ejemplo, para el formato de salida HTML, se anexará el sufijo `.html` al nombre de archivo en la URL. Consulte el apartado URL de las plantillas de documento nuevo [286] para obtener más información.

2. La plantilla `Documento nuevo` contiene una **Sección inicial de documento**. Si quiere, puede añadir más secciones de documento. Para ello siga las instrucciones de la sección Secciones del documento 420.

6.6.2 Plantillas de documento nuevo y estructura del diseño

Cuando cree múltiples documentos de salida, debe crear las plantillas `Documento nuevo` en los nodos adecuados del documento raíz. Por tanto, cuando diseñe múltiples documentos de salida debe tener en cuenta tanto la estructura de los documentos de salida 286 como la estructura de entrada (del documento XML de origen) 286.

Documento de salida principal y documentos de salida adicionales (estructura de salida)

Cuando se añade la primera plantilla `Documento nuevo` al diseño, todo el contenido del diseño que esté fuera de esta plantilla se asigna automáticamente a un documento distinto. Este otro documento se interpreta como el documento de salida principal y en las vista previas de StyleVision se denomina *Documento de salida principal.*

En los documentos de salida generados (creados con los comandos de menú **Archivo | Guardar archivos generados**), el nombre del documento de salida principal será el nombre que le asigne cuando genere los documentos de salida con el comando **Guardar archivos generados**. El nombre de los documentos de salida adicionales se toma del nombre que se les asignó en las URL de sus correspondientes plantillas `Documento nuevo`.

Plantillas de documento nuevo y estructura del documento de origen

Cuando se crea una plantilla `Documento nuevo` en el diseño, la posición jerárquica donde se crea desempeña un papel importante. Hay dos posibilidades:

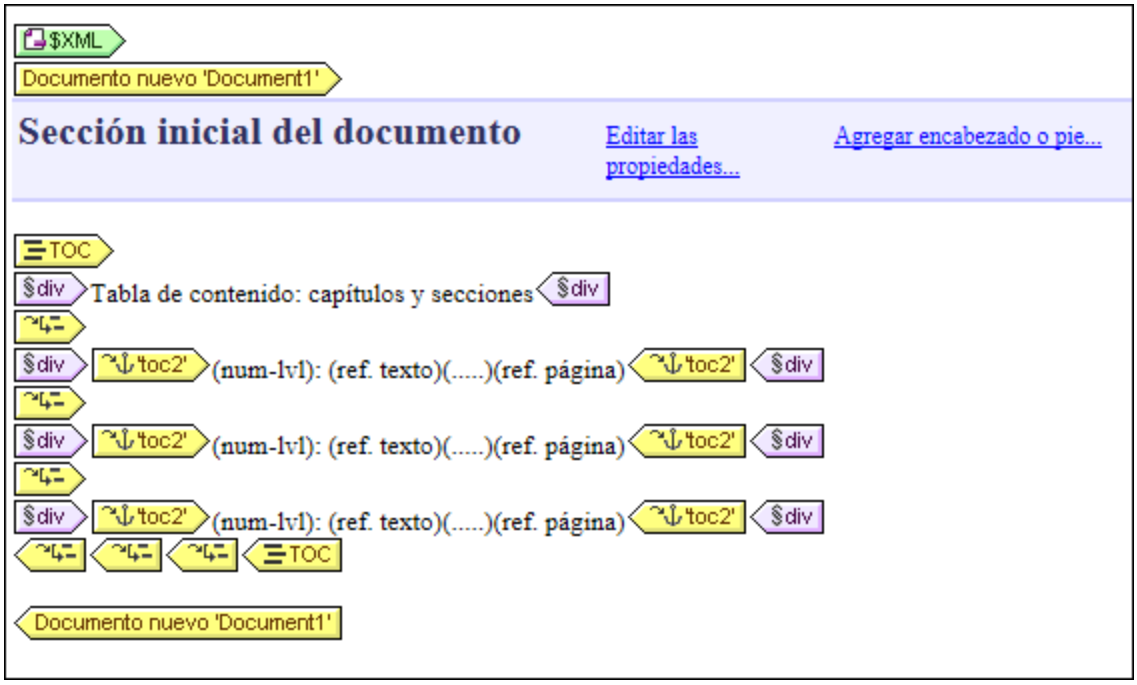
1. La plantilla `Documento nuevo` se crea en un nodo que **se procesa una sola vez**. En este caso la plantilla `Documento nuevo` también se procesa una sola vez. Esto significa que el nombre de archivo de la propiedad URL de la plantilla `Documento nuevo` puede ser un nombre estático.
2. La plantilla `Documento nuevo` se crea en un nodo que **se procesa varia veces**. Como resultado, la plantilla `Documento nuevo` se procesará tantas veces como se procese el nodo. Por ejemplo: un elemento `Oficina` con varios elementos secundarios llamados `Departamento` (para los departamentos de la oficina). Supongamos que creamos una plantilla `Documento nuevo` dentro del nodo `Departamento`. Como el nodo `Departamento` se procesará varias veces (tantas veces como elementos `Departamento` tenga el elemento `Oficina`), la plantilla `Documento nuevo` también se procesará varias veces, tantas como elementos `Departamento` existan en el XML de origen. Esto significa que el nombre de archivo de la propiedad URL de la plantilla `Documento nuevo` debe ser un nombre dinámico. De lo contrario, los documentos de salida creados para los elementos `Departamento` tendrían el mismo nombre de archivo.

6.6.3 URL de las plantillas de documento nuevo

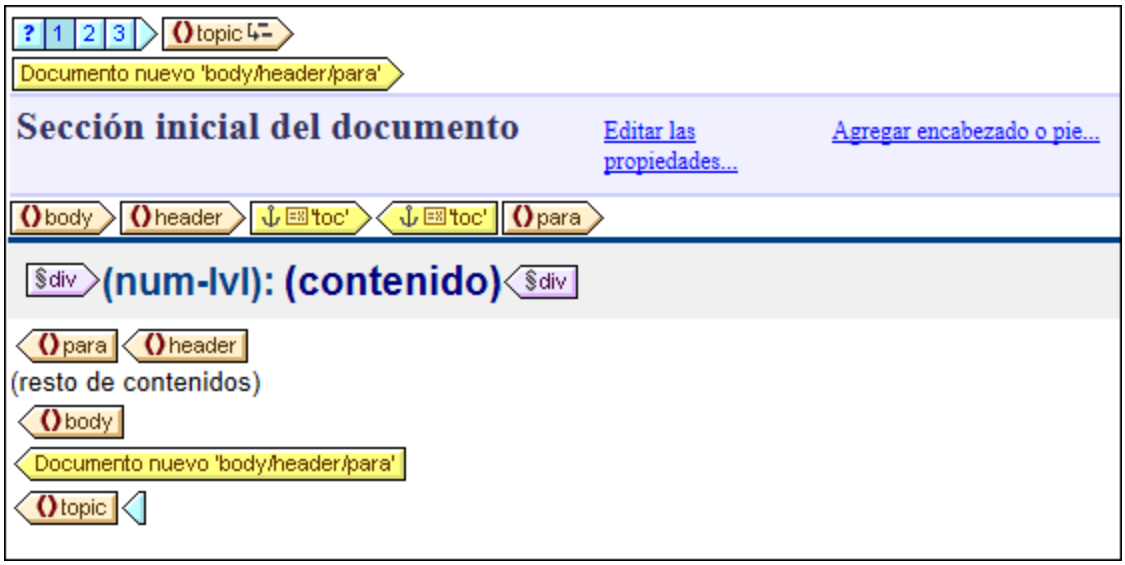
Este apartado describe la relación entre las URL de las plantillas 287 `Documento nuevo` 287 y la estructura del diseño 287, cómo editar las URL 288 y cómo crear vínculos entre varios documentos de salida 289.

Direcciones URL de las plantillas de documento nuevo

Si la plantilla `Documento nuevo` se procesa una sola vez (ver *apartado anterior* 286), entonces la propiedad URL de la plantilla puede ser una URL estática. La plantilla de la imagen siguiente, por ejemplo, se procesará una sola vez porque está dentro del elemento de documento (`$XML`). La URL tiene asignado el valor estático `Document1`. Este valor será el nombre de archivo del documento de salida. Como no se añadió la ruta al nombre de archivo, el archivo se generará en el mismo directorio que el documento de salida principal (consulte Vista previa de archivos y documentos de salida 289). Si la URL incluyera una ruta de acceso, el documento de salida se guardaría en la ubicación especificada por la ruta de acceso.

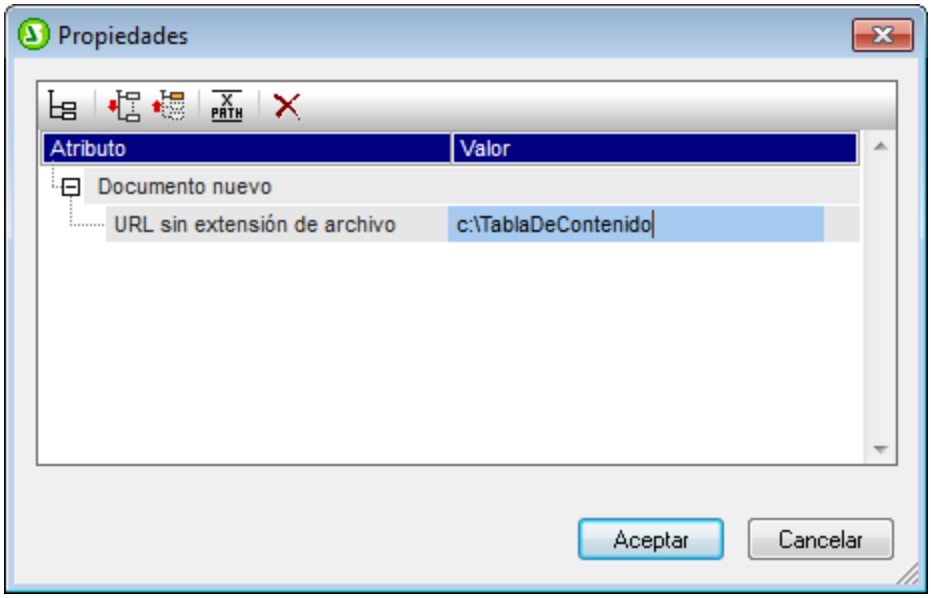


Si, por el contrario, la plantilla `Documento nuevo` se procesa varias veces, generando varios documentos de salida (ver *apartado anterior* 286), entonces la propiedad URL de la plantilla debe ser una URL dinámica seleccionada con una expresión XPath. En la imagen siguiente, por ejemplo, la URL de la plantilla `Documento nuevo` es la expresión XPath: `body/header/para`. La plantilla `Documento nuevo` está dentro del elemento topic así que se procesará cada vez que se procese este elemento. Con cada iteración en el elemento `topic`, el contenido del secundario `body/header/para` del elemento `topic` se asignará como URL de la plantilla `Documento nuevo`. Como resultado se crea un documento nuevo por cada elemento `topic`. Cada uno de estos documentos tendrá un nombre distinto, el del elemento `body/header/para` (es decir, el texto del título de cada tema).



Editar la URL

Cuando se añade una plantilla `Documento nuevo` al diseño, la plantilla se crea con una URL predeterminada: la cadena de texto estática `DocumentX` (siendo `X` un número entero). Si quiere editar la URL, haga clic con el botón derecho en la etiqueta de la plantilla `Documento nuevo` y seleccione el comando **Editar URL...** del menú contextual. Se abre el diálogo "Propiedades" (*imagen siguiente*) donde puede editar el campo *Valor* de la propiedad `URL sin extensión de archivo`.



Si quiere introducir una URL estática, escriba el texto en el campo *Valor*. Si prefiere usar una URL dinámica, haga clic en el campo *Valor* y después en el botón **XPath** de la barra de herramientas del diálogo. Aparece otro cuadro de diálogo donde puede introducir la XPath que desea usar. Recuerde que: (i) el nodo de contexto para la expresión XPath es el nodo donde se insertó la plantilla `Documento nuevo` y (ii) para anexar la ruta de acceso a la expresión XPath debe utilizar la función `concat()` de XPath. Por ejemplo: `concat('C:\MiSalida\', body\header\para)`. Esta expresión XPath generaría la cadena de dirección URL: `C:`

`\MiSalida\nombreArchivo`. La extensión de archivo se generará automáticamente según el formato de salida elegido.

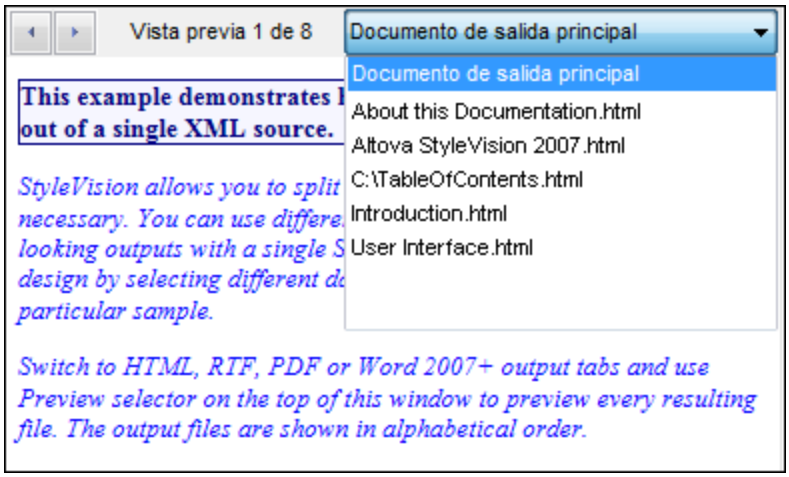
Crear vínculos entre los documentos

Puede crear vínculos entre los documentos de salida por medio de marcadores e hipervínculos 364. Los marcadores 364 se pueden insertar en el encabezado de la plantilla `Documento nuevo` o en cualquier otra posición de la plantilla. Después, en los demás documentos, puede crear hipervínculos 366 que apunten al marcador. Si necesita insertar marcadores en un nodo que se procesa varias veces, compruebe que el nombre del marcador se genera de forma dinámica. De lo contrario (si le asigna un nombre de marcador estático), en el documento de salida habrá varios nodos con el mismo nombre de marcador.

También puede usar una tabla de contenido (TDC) 337 para crear vínculos entre los documentos. La tabla de contenido puede estar en un documento distinto (por ejemplo, en el documento de salida principal) y apuntar a los demás documentos de salida (mientras que los documentos de salida apuntan a su vez a la tabla de contenido del documento principal).

6.6.4 Vista previa de archivos y documentos de salida

En las vistas previas de StyleVision puede ver cada uno de los múltiples documentos de salida que se especificaron en el diseño (*imagen siguiente*).

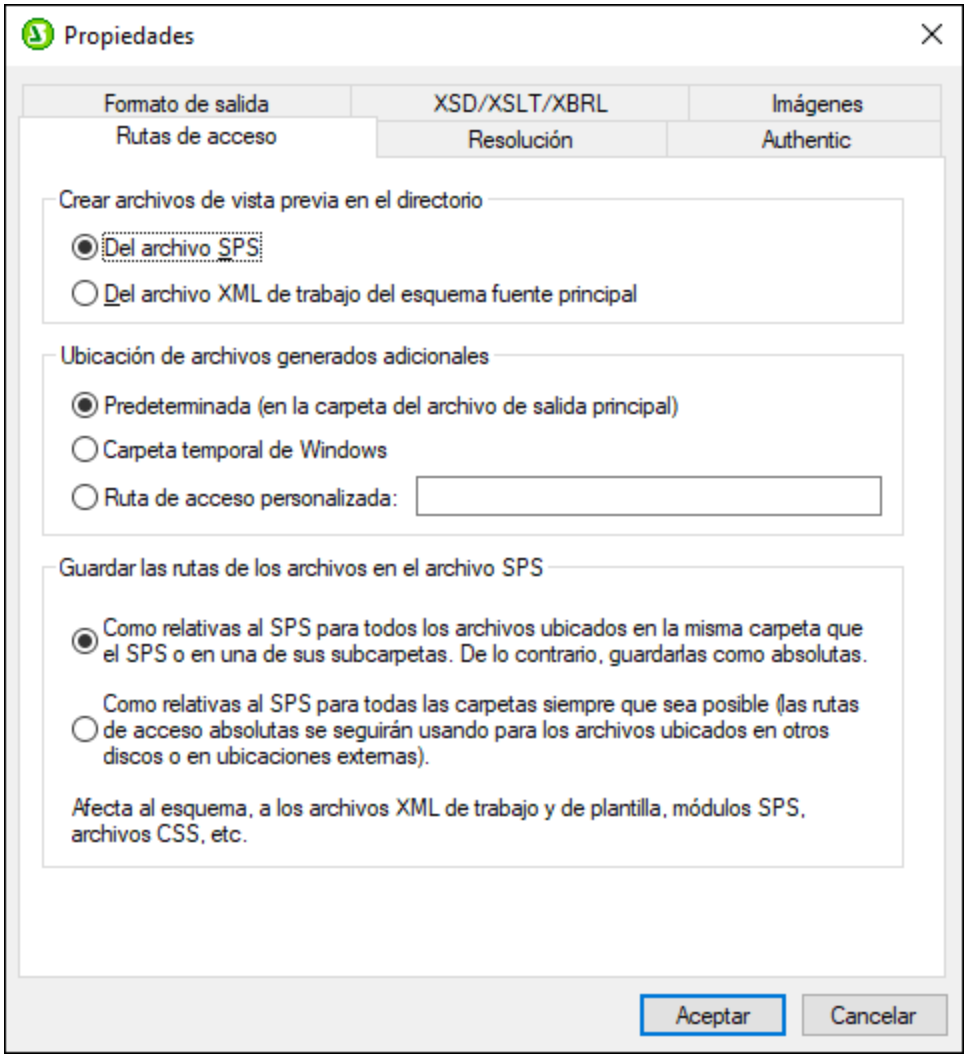


Por ejemplo, la imagen anterior muestra la vista **HTML** de un documento SPS diseñado para generar múltiples documentos de salida. Para invocar los diferentes documentos desde la ventana de vista previa: (i) haga clic en los botones de flecha situados en la esquina superior izquierda para avanzar por los documentos disponibles o (ii) seleccione el documento pertinente en la lista desplegable del cuadro combinado (*imagen anterior*). Observe que la lista desplegable muestra la URL completa (la ruta más el nombre de archivo).

Ubicación de los archivos de vista previa

Los archivos de vista previa se crean por defecto en el directorio donde se creó el archivo SPS. Puede cambiar la ubicación predeterminada en la pestaña *Rutas de acceso* del cuadro de diálogo "Propiedades" (*imagen siguiente*) del archivo SPS (que se abre haciendo clic en **Archivo | Propiedades**). Aquípuede especificar como ubicación alternativa la del directorio del archivo XML de trabajo. Si la URL de una plantilla `Documento`

`nuevo` contiene una ruta de acceso, la ubicación especificada en esta ruta se usará como ubicación de los respectivos archivos de vista previa. Si la ubicación no se encuentra, se emite un error. Si tiene pensado crear vínculos entre los documentos de salida, debería conocer la ubicación de destino de estos archivos.



En la pestaña *Rutas de acceso* del cuadro de diálogo "Propiedades" (*imagen anterior*) también puede especificar dónde se guardan los archivos adicionales temporales (como archivos de salida, imágenes, gráficos, etc.). No olvide que, si la URL de una plantilla `Documento nuevo` contiene una ruta de acceso, se usará la ubicación indicada por esta ruta.

Generar documentos de salida (rutas de acceso, etc.)

Siga estas instrucciones para generar los documentos de salida:

1. Pase el ratón por el comando de menú **Archivo | Guardar archivos generados** y haga clic en el formato de salida que desea generar.
2. Aparece el cuadro de diálogo "Guardar el archivo generado como". Busque la carpeta donde desea guardar el archivo generado.
3. Escriba el nombre del documento de salida principal y haga clic en **Guardar**.

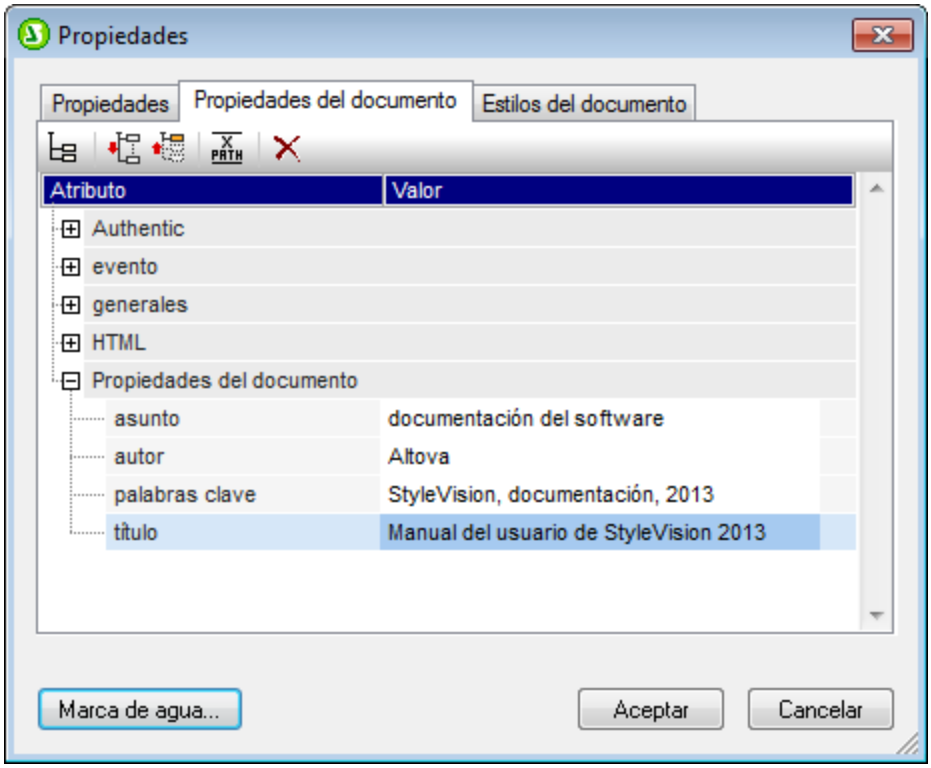
 © 2016-2022 Altova GmbH

Una vez terminado el proceso aparece una ventana de información que enumera la ubicación de todos los documentos de salida, así como de los demás archivos generados (imágenes, gráficos, etc.).

El documento de salida principal se guardará en la ubicación seleccionada en el cuadro de diálogo "Guardar el archivo generado como". Los demás documentos de salida creados con las plantillas `Documento nuevo` que no tengan una ruta de acceso en su URL se guardarán en la misma carpeta que el documento de salida principal. Si añadió una ruta de acceso al nombre de archivo en la URL de la plantilla `Documento nuevo`, el documento de salida se guardará en dicha ubicación. Si la ubicación no existe, se emite un error.

6.6.5 Propiedades y estilos de los documentos

En el diseño SPS puede dividir los resultados en varios documentos de salida. A cada documento de salida se le pueden asignar propiedades y estilos de documentos propios. Estas propiedades y estilos se configuran en las pestañas *Propiedades del documento* y *Estilos del documento* (*imagen siguiente*) del cuadro de diálogo "Propiedades" de la **Sección inicial del documento**. Para abrir el diálogo "Propiedades", haga clic en el enlace `Editar las propiedades...` que aparece en la barra de la **Sección inicial del documento** del documento que desea configurar. Las propiedades y estilos se aplican a todo el documento de salida.



En la pestaña *Propiedades del documento*, el grupo *Propiedades del documento* permite introducir metadatos para el documento. Estos metadatos se guardarán en el documento de salida de acuerdo con el formato de salida elegido. Por ejemplo, en los documentos de salida HTML, las propiedades se almacenan en las etiquetas `META` del elemento `HEAD`.

Los estilos de documento se describen en el apartado Establecer valores de estilos 400.

7 Archivos SPS: características avanzadas

Las secciones Archivos SPS: contenido 134 y Archivos SPS: estructura 219 de esta documentación describen el contenido y la estructura básicos de los archivos SPS. Sin embargo, a menudo puede ser necesario modificar o manipular el contenido o la estructura de los datos de origen. Por ejemplo, quizás necesite ordenar un grupo de nodos (nodos con datos personales, por ejemplo) según cierto criterio (por orden alfabético, por ejemplo). O quizás necesite agrupar todos los clientes de una BD según su ciudad de residencia. O a lo mejor necesita sumar las ventas de un producto en determinada ciudad.

Estas y otras operaciones pueden completarse con ayuda de las características avanzadas de StyleVision:

- Cálculos automáticos 293: un potente mecanismo basado en expresiones XPath para manipular datos, presentar los datos manipulados en los documentos de salida y actualizar nodos del documento XML con el resultado del cálculo.
- Condiciones 303: el procesamiento de las plantillas y su contenido puede estar condicionado a las estructuras de datos o valores del XML o al resultado de una expresión XPath.
- Agrupaciones 312: puede definir un procesamiento para un grupo de elementos seleccionados mediante una expresión XPath.
- Ordenaciones 321: puede ordenar un conjunto de elementos XML según varias claves de ordenación.
- Parámetros y variables 327: los parámetros se declaran a nivel global con un valor predeterminado. Estos valores se pueden reemplazar en tiempo de ejecución con valores pasados a la hoja de estilos desde StyleVision Server. Las variables se pueden definir en el SPS y a estas variables se les puede hacer referencia en el archivo SPS.
- Tabla de contenido (TDC) y referencias 337: puede construir tablas de contenido (TDC) en varias posiciones del documento de salida en todos los formatos de salida compatibles. El mecanismo de las tablas de contenido selecciona primero los componentes a los que se debe hacer referencia en la TDC y después hace referencia a estos componentes marcados. Otras características del SPS que utilizan referencias son: (i) la numeración automática 359 (los nodos que se repiten en el documento se pueden numerar automáticamente y a la numeración se le puede dar formato); (ii) las referencias de texto 363 (el texto del documento se puede marcar para hacerle referencia desde otras partes del documento) y (iii) los marcadores e hipervínculos 364 (los marcadores señalan puntos clave del documento de salida a los que se puede apuntar por medio de hipervínculos. Los hipervínculos también pueden enlazar con recursos externos usando diferentes métodos para determinar el URI de destino, estático, dinámico, ambos o entidades URI sin analizar). Todos estos mecanismos de referencia se describen en esta sección.

 © 2016-2022 Altova GmbH

7.1 Cálculos automáticos

Los **cálculos automáticos** sirven para (i) mostrar el resultado de una evaluación XPath en cualquier posición del documento de salida y (ii) para actualizar un nodo del documento XML principal (el documento XML que se edita en la vista Authentic) con el resultado de la evaluación XPath.

El mecanismo de los cálculos automáticos es ideal para:

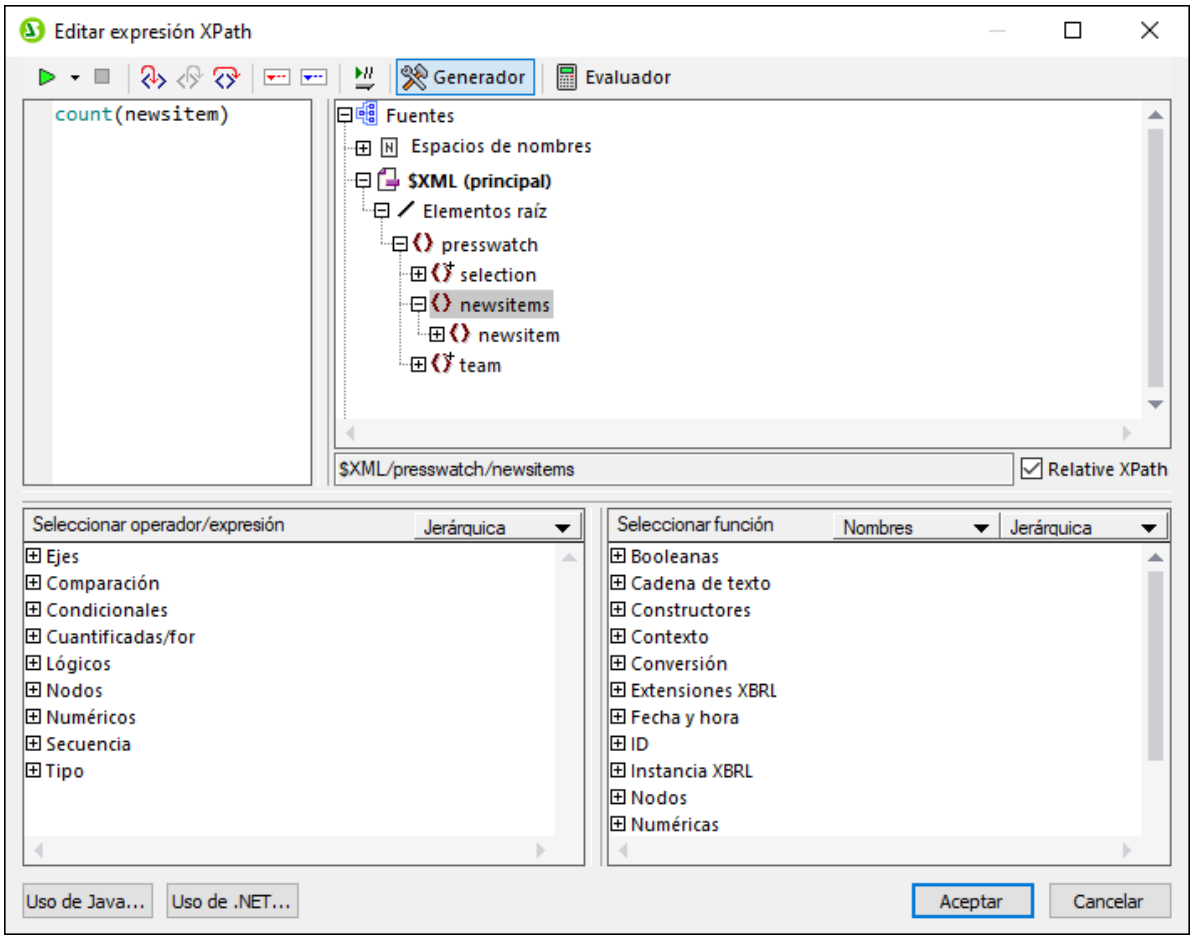
- Insertar cálculos basados en operaciones con valores de datos dinámicos. Por ejemplo, puede contar el número de elementos `Empleado` de un elemento `Oficina` (con `count(Empleado)`), o sumar los valores de todos los elementos `Precio` de cada elemento `Factura` (con `sum(Precio)`), o unir los elementos `Nombre` y `Apellido` de un elemento `Persona` (con `concat(Nombre, ' ', Apellido)`). De este modo puede generar datos nuevos a partir de datos dinámicos del documento XML y enviar el resultado generado al documento de salida.
- Mostrar información derivada de la estructura del documento. Por ejemplo, puede usar la función `position()` de XPath para insertar números de fila dinámicamente en una tabla dinámica o para numerar dinámicamente las secciones de un documento. La ventaja es que se puede generar información de forma dinámica a partir de la estructura cambiante del documento.
- Insertar datos de documentos XML externos. La función `doc()` de XPath 2.0 permite acceder a la raíz de documento de documentos XML externos y, por tanto, permite insertar el contenido de nodos del documento XML externo en el documento de salida.
- Actualizar el valor de los nodos del documento XML principal. Por ejemplo, el nodo `Destinatario` se puede actualizar con una expresión XPath como `concat(Título, ' ', Nombre, ' ', Apellido)`.
- Presentar el contenido de un nodo en cualquier posición del diseño.

7.1.1 Editar y mover cálculos automáticos

Crear cálculos automáticos

Siga estas instrucciones para crear un cálculo automático:

1. Ponga el cursor en la posición donde desea que aparezca el resultado del cálculo automático y haga clic en el comando de menú **Insertar | Cálculo automático**. En el submenú que aparece seleccione **Valor** (si quiere que el resultado aparezca como texto sin formato), **Campo de entrada** (si quiere que aparezca dentro de un campo de entrada o cuadro de texto) o **Campo de entrada multilínea** (si quiere que aparezca en un cuadro de texto multilínea). Es decir, el resultado del cálculo automático aparece como valor o en un campo de entrada. En la vista Authentic el resultado del cálculo automático no se puede editar. Tras elegir una de estas tres opciones aparece el cuadro de diálogo "Editar expresión XPath" (*imagen siguiente*).

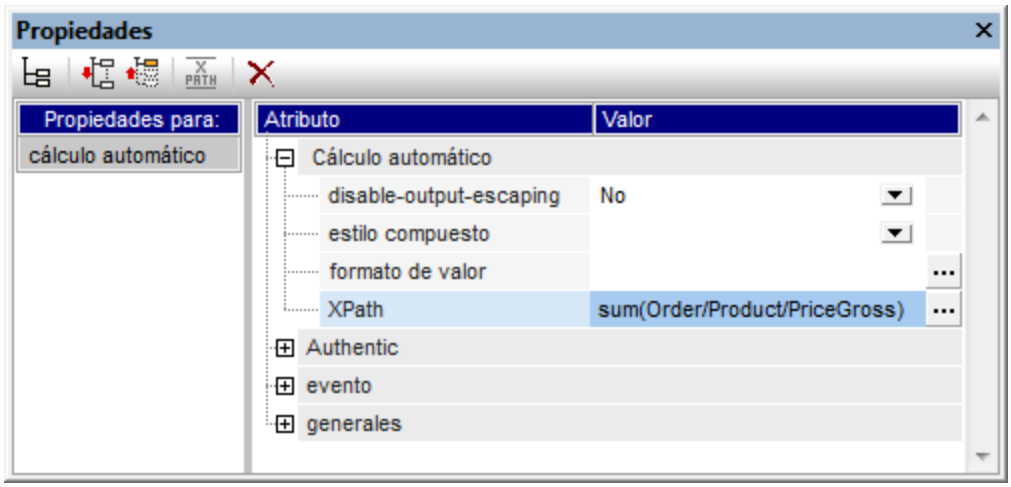


2. En el panel *Expresión XPath* (en la parte superior del diálogo) puede teclear la expresión XPath para el cálculo automático. También puede construir una expresión haciendo doble clic en los nodos, en los operadores o en las funciones de los paneles inferiores del cuadro de diálogo. Es importante conocer el nodo de contexto del punto de inserción. El nodo de contexto aparece resaltado en el árbol del esquema fuente (por ejemplo, en la imagen anterior el nodo de contexto es `newsitems`). Si seleccionó la versión XSLT 1.0 para su diseño SPS, debe usar expresiones XPath 1.0. Si seleccionó la versión XSLT 2.0 o XSLT 3.0, entonces debe usar expresiones XPath 2.0 o XPath 3.0. Para conocer mejor este cuadro de diálogo consulte la sección Cuadro de diálogo "Editar expresión XPath" 729.
3. Además, si quiere copiar el valor del cálculo automático en un nodo del documento XML, puede seleccionar el nodo con una expresión XPath. Para aprender a actualizar nodos con el resultado del cálculo automático consulte el apartado Actualizar nodos con cálculos automáticos 296.

Cuando termine de construir su expresión XPath haga clic en **Aceptar**. En la vista **Diseño** se inserta el cálculo automático. Para ver su resultado cambie a la vista Authentic o cualquier otra vista previa.

Editar los cálculos automáticos

Si quiere editar su expresión XPath, seleccione el cálculo automático y en la ventana Propiedades (*imagen siguiente*) haga clic en el botón **Editar** de la propiedad `XPath` (del grupo *Cálculo automático*). Al hacer clic en este botón aparece el cuadro de diálogo Editar expresión XPath 729 (*imagen anterior*), donde puede realizar los cambios.



Dar formato a los cálculos automáticos

Puede aplicar formatos predefinidos y estilos CSS a los cálculos automáticos como si fueran contenido de texto: seleccione el cálculo automático y aplique el formato. Además, puede especificar el formato de entrada 380 de un cálculo automático que sea de tipo de datos numérico o fecha. Esto se hace en la propiedad `formato de valor` de la ventana Propiedades (grupo de propiedades *Cálculo automático*).

Recuerde que puede incluir retornos de carro y saltos de línea (CR/LF) en la expresión XPath. Si el cálculo automático está incluido en el tipo de párrafo especial `pre`, el resultado de un CR/LF generará una línea nueva en el documento de salida (excepto en formato RTF). Un ejemplo de expresión XPath con CR/LF sería:

```
translate('a;b;c', ';', codepoints-to-string(13))
```

Mover los cálculos automáticos

Para mover un cálculo automático a otra posición del diseño, haga clic en el cálculo automático (para seleccionarlo) y arrástrelo a la nueva posición. También puede usar las operaciones cortar/copiar y pegar. Sin embargo, no olvide que deberá cambiar la expresión XPath si el nodo de contexto de la nueva posición es diferente al de la posición anterior.

Resumen

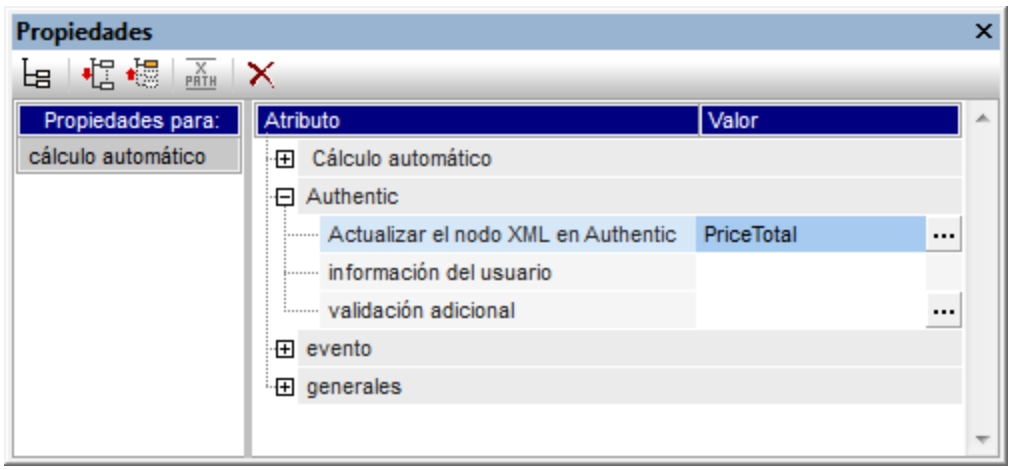
Estos son los puntos más importantes que debe tener en cuenta sobre los cálculos automáticos:

- Puede insertar cálculos automáticos en cualquier posición del diseño.
- La posición donde inserte el cálculo automático determina el nodo de contexto para la expresión XPath.
- En la vista Authentic el cálculo automático se vuelve a evaluar cada vez que cambia uno de los valores relacionados con el cálculo (es decir, cualquier nodo que forme parte de la expresión XPath).
- El resultado de los cálculos automáticos no se puede editar en la vista Authentic ni en las demás vistas previas.
- Puede actualizar cualquier nodo del documento XML con el resultado del cálculo automático.

7.1.2 Actualizar nodos con cálculos automáticos

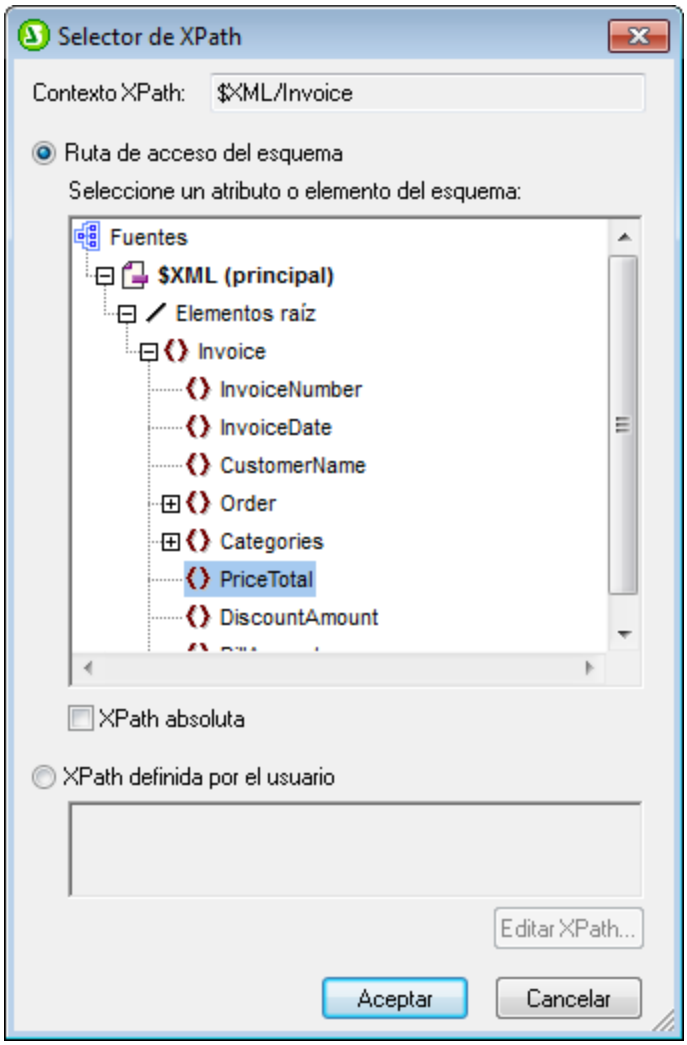
El valor (o resultado) de un cálculo automático se puede copiar en un nodo del documento XML principal (el documento que se edita en la vista Authentic). Estas son las instrucciones:

1. Cree el cálculo automático siguiendo las instrucciones del apartado Editar y mover cálculos automáticos 293.
2. Seleccione el cálculo automático y, en la ventana Propiedades (*imagen siguiente*), haga clic en el botón **Edición** de la propiedad `Actualizar el nodo XML en Authentic` (en el grupo de propiedades *Authentic*).



3. Aparece el cuadro de diálogo "Selector de XPath" (*imagen siguiente*). Seleccione el nodo que desea actualizar y haga clic en **Aceptar**. Si quiere que la expresión XPath sea absoluta (es decir, que empiece en la raíz del documento), marque la casilla *XPath absoluta*.

© 2016-2022 Altova GmbH



Si lo prefiere, también puede seleccionar el botón de opción *XPath definida por el usuario* e introducir la expresión XPath a mano.

La expresión XPath debe seleccionar **un solo nodo**. Si la expresión XPath selecciona varios nodos, no se actualizará ninguno.

NOTA IMPORTANTE
Para poder actualizar un nodo del documento XML principal deben cumplirse dos condiciones:

1. La expresión XPath para el cálculo automático debe incluir al menos un valor que esté relacionado con un nodo XML, es decir, un valor dinámico. Por ejemplo, `Price*1.2`. Esta expresión afecta al elemento `Price` y por tanto es dinámica. Si la expresión XPath contiene un valor estático (por ejemplo, `string("Nanonull, Inc.")`), la función de actualización de nodos XML no funcionará.
2. Debe modificar en la vista Authentic un nodo utilizado en la expresión XPath. Por ejemplo, si la expresión XPath es `Price*1.2` y el cálculo automático está configurado para actualizar el nodo `VATPrice`, entonces debe modificar el elemento `Price` en la vista Authentic para que el nodo `VATPrice` se pueda actualizar.

Cambiar el nodo que se debe actualizar y cancelar la actualización

Para cambiar el nodo que se debe actualizar, haga clic en el botón **Edición** de la propiedad `Actualizar el nodo XML en Authentic` (en la ventana Propiedades, grupo de propiedades *Authentic*) y después seleccione el nuevo nodo en el cuadro de diálogo "Selector de XPath" que aparece. Para eliminar la propiedad `Actualizar el nodo XML en Authentic`, haga clic en el botón **Eliminar** de la barra de herramientas de la ventana Propiedades.

¿Debería incluir el cálculo automático o el contenido del nodo actualizado en el resultado del diseño?

Si no se trata de una plantilla condicional, puede usar el cálculo automático o el contenido del nodo actualizado en el resultado del diseño. Es irrelevante cuál de los dos utiliza porque la actualización del nodo ocurre inmediatamente después de la evaluación del cálculo automático. Ahora bien, debería tener e cuenta que se utiliza una fuente diferente en cada caso.

Ocultar el cálculo automático

A menudo necesitará utilizar un cálculo automático para actualizar un nodo con el valor del cálculo automático. En este caso, puede darse una de estas dos situaciones:

- Quiere mostrar el resultado una sola vez, pero no puede ocultar el cálculo automático porque si lo oculta no se puede evaluar. Si no se trata de una plantilla condicional, es mejor incluir el cálculo automático en el resultado del diseño pero no incluir el contenido del nodo actualizado.
- No quiere mostrar el resultado sino solamente actualizar el nodo. En esta situación lo mejor es aplicar formato de texto al cálculo automático para que no sea visible en el documento de salida (por ejemplo, aplicándole el color de fuente blanco al texto del cálculo automático sobre un fondo blanco).

7.1.3 Cálculos automáticos basados en nodos actualizados

Si quiere crear un cálculo automático que use un nodo actualizado por otro cálculo automático, tiene dos opciones:

- Crear ambos cálculos automáticos en la misma plantilla.
- Crear cada cálculo automático en una plantilla diferente.

Cálculos automáticos en la misma plantilla

Cuando dos cálculos automáticos están en la misma plantilla, el diseño SPS sigue este procedimiento:

1. Se modifica un nodo utilizado en la expresión XPath del cálculo automático.
2. Se leen todos los valores de los nodos del documento XML y se ejecutan todos los cálculos automáticos.
3. Suponiendo que el primer cálculo automático se ejecute correctamente, el nodo XML especificado se actualiza (llamémosle `Nodo-A`). El segundo cálculo automático (basado en el `Nodo-A`) se ejecuta pero usa el valor del `Nodo-A` antes de que el nodo se actualice. Esto se debe a que el SPS lee el valor del `Nodo-A` antes de que este se actualice.

© 2016-2022 Altova GmbH

4. Si a partir de este momento se edita el documento XML o si se cambia de vista (de la vista Authentic a otra vista previa), entonces los valores de los nodos se vuelven a leer y los cálculos automáticos se ejecutan.
5. Ahora se lleva a cabo el segundo cálculo automático. Si el objetivo de este cálculo automático es actualizar otro nodo más, deberá cambiar uno de los nodos utilizados en la expresión XPath para que dicha actualización pueda producirse.

El intervalo temporal entre la actualización del `Nodo-A` y la evaluación del segundo cálculo automático con el valor actualizado del `Nodo-A` puede confundir al usuario de la vista Authentic. Por ello es mejor que la expresión XPath del segundo cálculo automático contenga la expresión XPath del primer cálculo automático (en lugar del nodo actualizado). Como resultado el segundo cálculo automático se ejecutará con los datos de entrada del primer cálculo automático y llevará a cabo el cálculo automático como parte de su propio cálculo automático. De este modo el cálculo automático se evalúa independientemente del contenido del `Nodo-A`.

Ejemplo

El primer cálculo automático calcula el importe del IVA de un producto usando los nodos de (i) precio neto y (ii) tasa de IVA. Después actualiza el nodo que contiene el importe del IVA. El segundo cálculo automático calcula el producto bruto (precio neto + IVA) y después actualiza el nodo que contiene el precio bruto.

- El cálculo automático necesario para calcular el IVA del producto sería: `NetPrice * VATRate div 100`. Cuando se introduce la tasa de IVA del producto, el cálculo automático se ejecuta y actualiza el nodo `VATAmount`.
- Si el cálculo automático para calcular el precio bruto es: `NetPrice + VATAmount`, entonces el cálculo automático se ejecutará con el valor de `VATAmount` que se leyó antes de que `VATAmount` se actualizara.
- Si, por el contrario, el cálculo automático para calcular el precio bruto fuera: `NetPrice + (NetPrice * VATRate div 100)`, entonces el cálculo automático se ejecutará con el valor de `VATRate` y acutalizará el nodo `GrossPrice`. El nodo actualizado `VatAmount` no se incluyó en el segundo cálculo automático.

Para ver un ejemplo concreto consulte el apartado Ejemplo: una factura 299.

Cálculos automáticos en plantillas diferentes

Si los cálculos automáticos están en plantillas diferentes, el nodo actualizado por el primer cálculo automático se puede usar en el segundo cálculo automático. Esto se debe a que los cálculos automáticos y los nodos se ejecutan/actualizan por separado en cada plantilla. Para ver un ejemplo concreto consulte el apartado Ejemplo: una factura 299.

7.1.4 Ejemplo: una factura

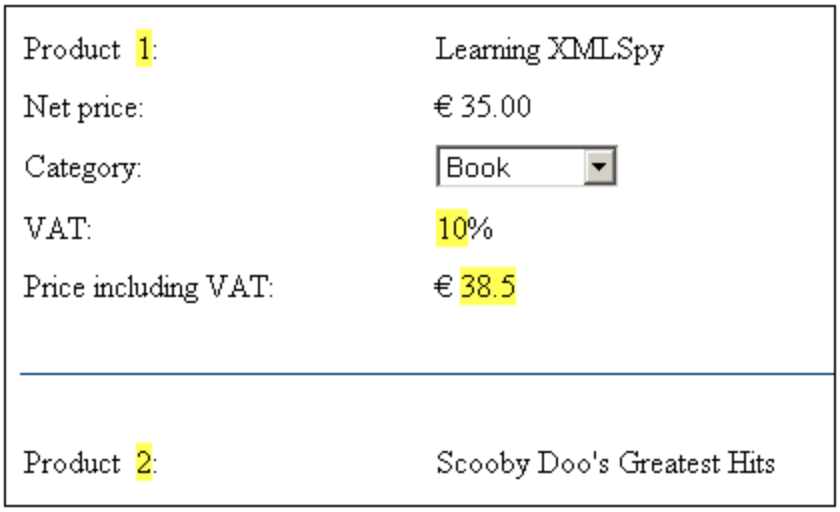
El archivo SPS de ejemplo `Invoice.sps` de la carpeta (Mis) Documentos 34, `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\Auto-Calculations\`, demuestra cómo usar cálculos automáticos para:

- Contar nodos
- Seleccionar un nodo en función de los datos introducidos por el usuario de la vista Authentic
- Actualizar el contenido de un nodo con el resultado del cálculo automático
- Usar el resultado de un cálculo automático en otro cálculo automático

En el archivo SPS de ejemplo, los cálculos automáticos aparecen resaltados en amarillo (*imagen siguiente*).

Contar nodos

En el archivo SPS de ejemplo cada producto de la lista está numerado según su posición en la lista de productos solicitados por el cliente (`Product 1`, `Product 2`, etc). Esta numeración se consigue gracias a un cálculo automático (*imagen siguiente*).



En este caso concreto la expresión XPath `position()` sería suficiente para obtener la numeración correcta. Otra manera muy fácil de obtener la posición de un nodo es contar el número de elementos del mismo nivel anteriores y sumarle uno. La expresión XPath sería: `count(preceding-sibling::Product)+1`. Este último método es ideal para los contextos en los que es difícil o imposible usar la función `position()`. Puede probar este cálculo automático en el archivo de ejemplo eliminando productos y añadiendo productos nuevos.

Seleccionar un nodo en función de los datos introducidos por el usuario

En el archivo de ejemplo el usuario selecciona la categoría de productos (`Book`, `CD`, `DVD`, or `Electronics`) en un cuadro combinado. Esta selección se introduce en el nodo `//Product/Category` del documento XML. Después un cálculo automático utiliza este valor para remitirse a una "tabla de búsqueda" del documento XML e identificar el nodo que contiene el porcentaje de IVA para esta categoría de productos. La expresión de este cálculo automático es:

```
for $i in Category return /Invoice/Categories/Category[. = $i]/@rate.
```

En el documento de salida aparece la tasa de IVA en la posición donde se insertó el cálculo automático. En el archivo SPS de ejemplo la tabla de búsqueda se almacena en el mismo documento XML que los datos de la factura. No obstante, esta tabla también se puede almacenar en un documento distinto, en cuyo caso habría que acceder a ella con la función `doc()` de XPath 2.0. Para probar este cálculo automático seleccione otro tipo de producto en cualquier cuadro combinado *Category* y verá cómo el valor de la tasa de IVA cambia (`Book=10%`; `CD=15%`; `DVD=15%`; `Electronics=20%`).

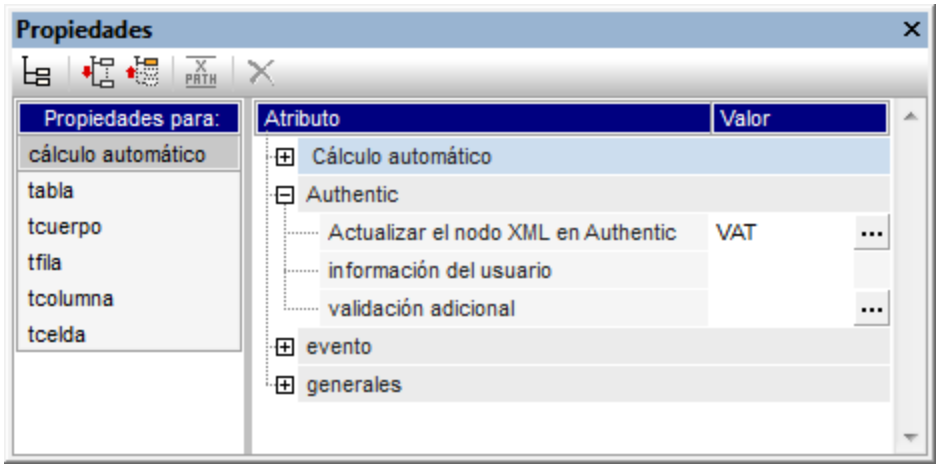
Actualizar el contenido de un nodo con el resultado de un cálculo automático

La tasa de IVA que el cálculo automático obtiene de la tabla de búsqueda es un valor dinámico que se almacena temporalmente en memoria (para usarla en la vista Authentic). El resultado del cálculo automático,

 © 2016-2022 Altova GmbH

sin embargo, se puede almacenar en el nodo `VAT` del documento XML. Esto tiene dos ventajas: (i) no hace falta que el usuario introduzca el contenido del nodo VAT porque el cálculo automático lo introduce automáticamente y (ii) cada vez que se modifique la tabla de búsqueda, los cambios se aplicarán al nodo VAT cuando se abra el archivo `Invoice.xml` en Authentic. Para configurar un cálculo automático para que actualice un nodo debe seguir estas instrucciones:

1. Seleccione el cálculo automático en la vista **Diseño**.
2. En la ventana Propiedades (*imagen siguiente*) expanda el grupo de propiedades *Authentic* y haga clic en el botón **Edición** de la propiedad `Actualizar el nodo XML en Authentic`.
3. Aparece un cuadro de diálogo. Seleccione el nodo `VAT` y después haga clic en **Aceptar**.



A partir de ese momento cuando el usuario de la vista Authentic seleccione una categoría de productos diferente en el cuadro combinado, el cálculo automático obtendrá la tasa de IVA (remitiéndose a la tabla de búsqueda), mostrará la tasa de IVA y actualizará el nodo `VAT` del documento XML.

Usar en un cálculo automático un nodo actualizado por medio de otro cálculo automático

La tasa de IVA obtenida con el cálculo automático (ver más arriba) se necesita para calcular el precio bruto (precio neto + IVA) de cada producto. La fórmula necesaria se podría derivar así:

`Gross Price = Net Price + VAT-amount`
Puesto que `VAT-amount = Net Price * VAT-percentage div 100`
Entonces `Gross Price = Net Price + (Net Price * VAT-percentage div 100)`

El precio neto de un producto se obtiene del nodo `PriceNet`. La tasa de IVA se calcula con el cálculo automático descrito más arriba y dicho cálculo automático actualiza el nodo `VAT`. El contenido del nodo `VAT` se puede usar en otro cálculo automático para generar el precio bruto. Para ello usaríamos esta expresión XPath:

```
PriceNet  + (PriceNet * (VAT div 100))
```

La expresión XPath se puede ver y editar en la ventana Propiedades 293. Para probar el cálculo automático que calcula el precio bruto cambie el precio o la categoría de producto. Observe que el precio bruto (con IVA incluido) también cambia.

Product 6:	A Short History of the American Century
Net price:	€ 20.00
Category:	DVD
VAT:	15%
Price including VAT:	€ 23

Price Total: € 358.9

En el archivo SPS de ejemplo el cálculo automático que calcula el precio bruto actualiza el nodo `PriceGross` del documento XML.

Los nodos `PriceGross` actualizados se pueden usar en un cálculo automático que sume los precios de todos los productos comprados. La expresión XPath sería: `sum(Order/Product/PriceGross)`. Este cálculo automático actualiza el nodo `PriceTotal`. Para probarlo modifique los precios de los diferentes productos y verá el efecto que esto tiene en el precio total.

Ejercicio práctico

Ahora intente añadir dos cálculos al archivo SPS:

1. Primero inserte un cálculo automático que calcule un descuento sobre el importe total. Si el precio de la compra (el precio total) supera los 100, 300 o 600 euros, se debe aplicar un descuento del 5%, 10% y 12% respectivamente. Inserte la cantidad del descuento en el documento de salida (*imagen siguiente*) y actualice el nodo `DiscountAmount` del documento XML.
2. Después inserte un cálculo automático que calcule el importe final tras deducir el descuento. Esta cantidad será igual al precio total menos el descuento (que se calculó en el cálculo automático anterior). Inserte el importe final de la factura en el documento de salida (*imagen siguiente*) y actualice el nodo `BillAmount` del documento XML.

Configure estos cálculos automáticos de modo que la vista Authentic del documento tenga este aspecto.

Price Total:	€	358.90	**Volume Discounts**
Less Volume Discount:	€	35.89	Over € 100 = 05% Over € 300 = 10%
Bill Amount:	€	323.01	Over € 600 = 12%

Estos dos cálculos automáticos aparecen en el archivo de ejemplo `InvoiceWithDiscounts.sps`, que está en la carpeta (Mis) Documentos[34] `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\Auto-Calculations`.

7.2 Condiciones

Puede insertar condiciones en cualquier posición del diseño SPS, tanto en la plantilla principal como en plantillas globales. Una condición es un componente formado por una o más ramas, estando cada rama definida por una expresión XPath. Por ejemplo, imagine una condición compuesta por dos ramas. La expresión XPath de la primera rama comprueba si el valor del atributo `Ubicación` del nodo de contexto es "`EE UU`". La expresión XPath de la segunda rama comprueba si el valor del atributo `Ubicación` es "`UE`". Cada rama contiene una plantilla que denominamos *plantilla condicional*. Cuando un nodo se procesa con una condición, se ejecuta la primera rama cuyo resultado sea `true`, es decir, se procesa su plantilla condicional y el procesador sale de la condición. Es decir, no se evalúan más ramas. De este modo, puede usar distintas plantillas dependiendo del valor de un nodo. Por ejemplo, podría usar una plantilla para los datos de EE UU y otra para EU.

Este apartado de la documentación abarca estos temas:

- Definir condiciones 303: cómo crear condiciones en el diseño y cómo crear ramas condicionales.
- Editar condiciones 307: cómo editar las expresiones XPath de una condición.
- Condiciones basadas en el resultado 307: cómo utilizar condiciones para generar resultados diferentes para cada formato de salida.
- Condiciones y cálculos automáticos 308: aspectos que debe tener en cuenta cuando use condiciones junto con cálculos automáticos.

7.2.1 Definir condiciones

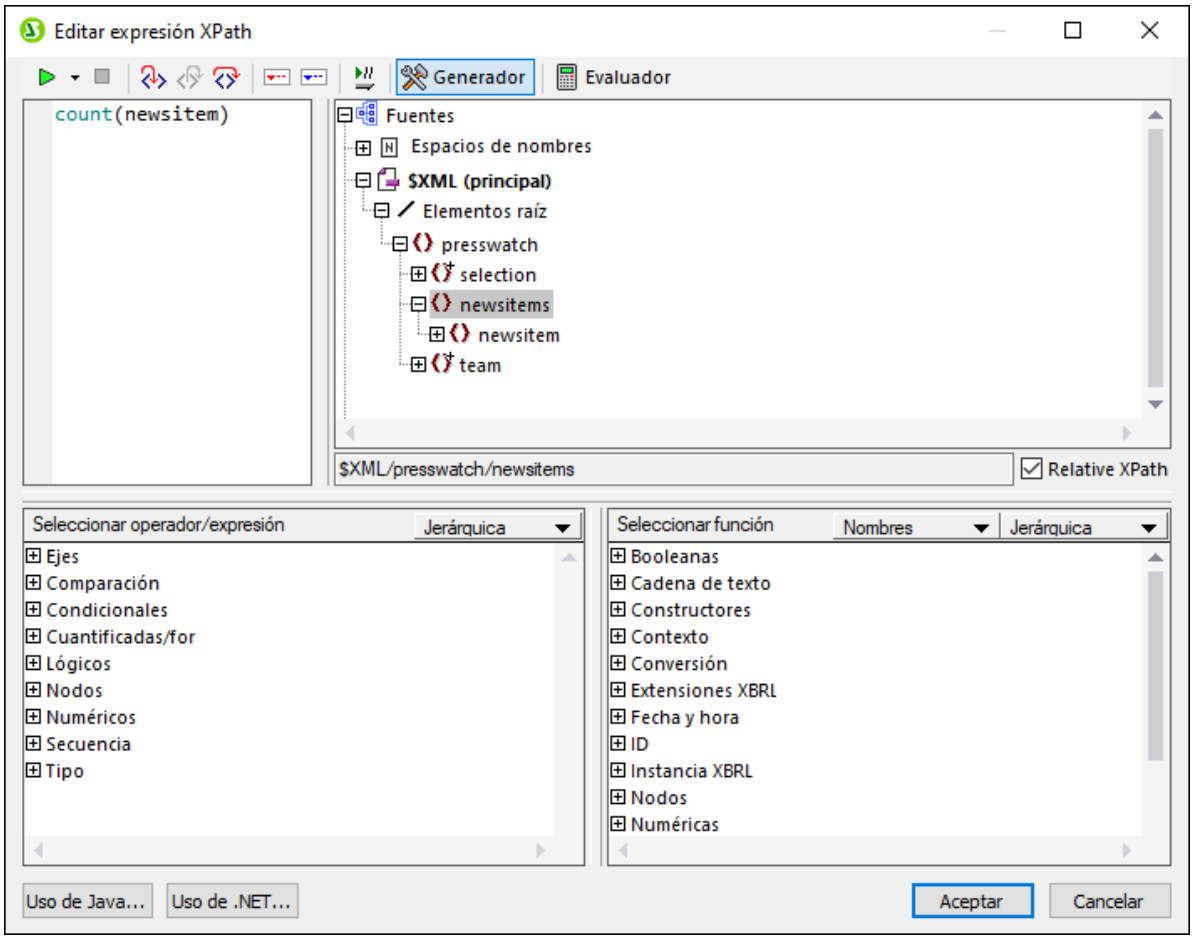
El proceso de definición de condiciones tiene tres pasos:

1. Primero se crea la condición con su primera rama.
2. Después se crean las demás ramas que ofrecen alternativas de procesamiento.
3. Finalmente se crean y editan las plantillas dentro de cada rama condicional.

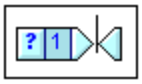
Crear la condición con su primera rama

Siga estas instrucciones para definir una condición:

1. Ponga el cursor en cualquier punto del diseño o seleccione un componente del diseño y después haga clic en el comando de menú **Insertar | Condición**. Aparece el cuadro de diálogo Editar expresión XPath 729 (*imagen siguiente*).



2. En el panel *Expresión XPath* (panel superior del cuadro de diálogo) escriba a mano la expresión XPath para la primera rama condicional. También puede generar una expresión XPath haciendo doble clic en los nodos, operadores y funciones de los paneles inferiores del cuadro de diálogo. Es importante conocer cuál es el nodo de contexto del punto de inserción y por eso, al abrir el cuadro de diálogo, el nodo de contexto aparece resaltado en el árbol de esquemas fuente.
3. Cuando termine de construir su expresión XPath haga clic en **Aceptar**. La condición se crea en el diseño con su primera rama, cuya expresión XPath es la que introdujo en el cuadro de diálogo. Si la condición se insertó en un punto de inserción de texto, la primera rama estará vacía (no tendrá dentro ninguna plantilla, como en la imagen siguiente). Si la condición se insertó estando seleccionado un componente, la condición se crea alrededor del componente y este pasa a ser la plantilla de la primera rama.



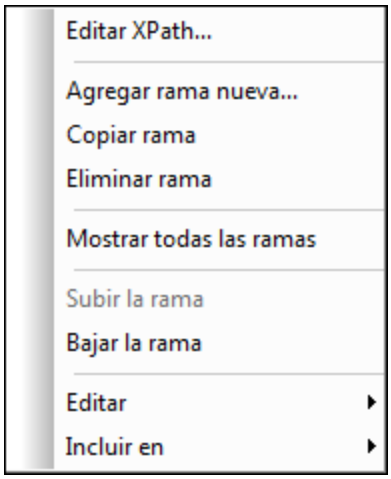
Para seleccionar toda la condición, haga clic en la celda que tiene el signo de interrogación. Para seleccionar la primera rama, haga clic en la celda que tiene el número 1.

Tras crear la condición con su primera rama (que puede tener dentro una plantilla o no) podrá crear tantas ramas adicionales como quiera.

© 2016-2022 Altova GmbH

Crear las demás ramas de la condición

Las demás ramas de la condición se deben crear una a una. Para crear una rama más, haga clic con el botón derecho y seleccione una de estas opciones en el menú contextual: (i) **Agregar rama nueva...** si quiere crear una rama sin plantilla y (ii) **Copiar rama** si quiere copiar una plantilla ya existente en la nueva rama.



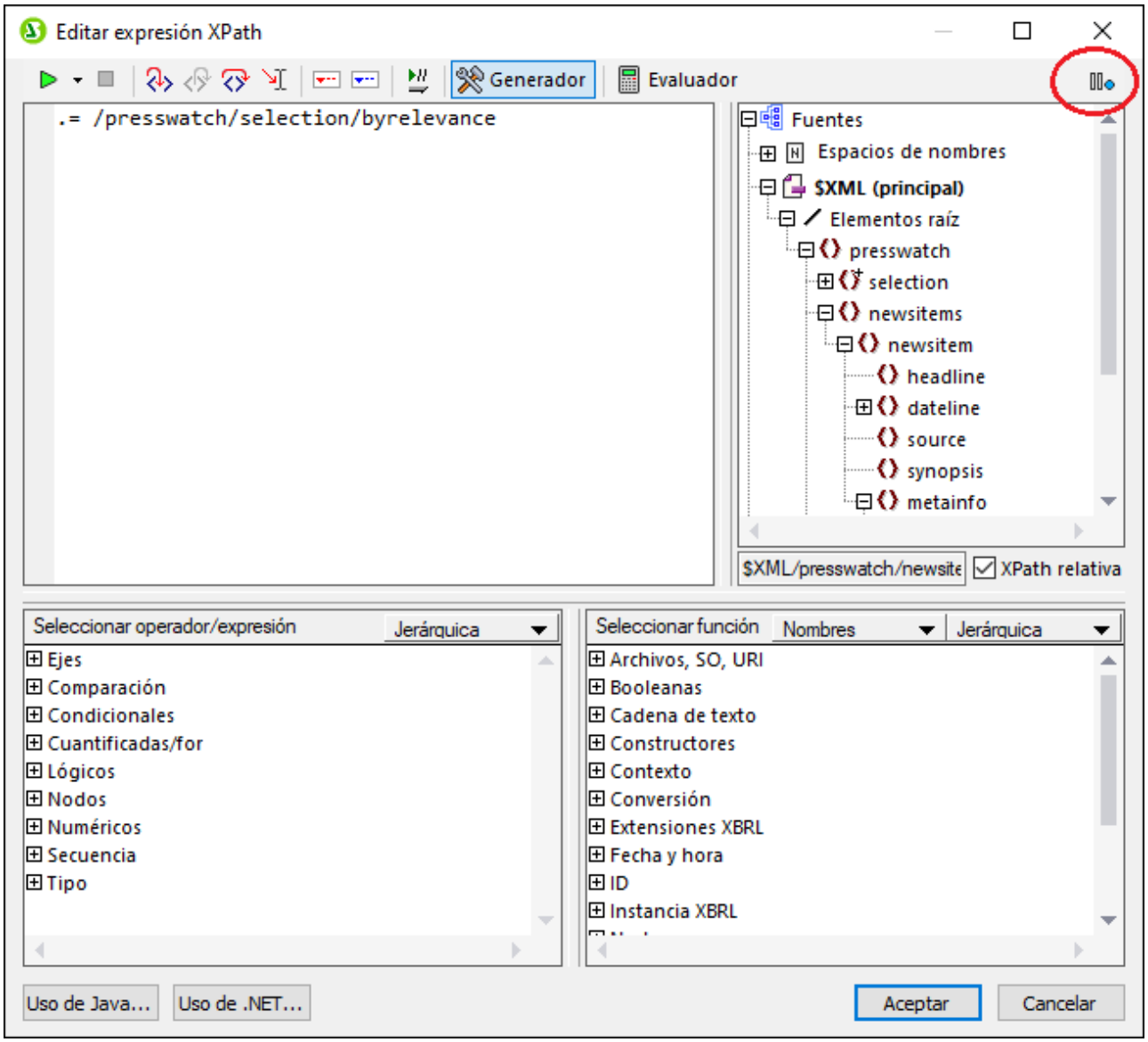
Para crear una rama nueva, haga clic con el botón derecho en cualquier rama condicional y seleccione **Agregar rama nueva** en el menú contextual. Aparece el cuadro de diálogo Editar expresión XPath 729. Tras introducir la expresión XPath, haga clic en **Aceptar** para añadir una rama nueva vacía a la condición. En el diseño verá que aparece una celda nueva con un número igual al número de la última rama + 1.

Para crear una copia de una rama, haga clic con el botón derecho en la rama que quiere copiar y seleccione **Copiar rama**. Aparece el cuadro de diálogo Editar expresión XPath 729 con la expresión XPath de la rama copiada. Tras modificar la expresión XPath, haga clic en **Aceptar** para añadir una rama nueva a la condición. Esta nueva rama contiene una copia de la plantilla de la rama que se copió. En el diseño verá que ahora hay una celda nueva con un número igual al número de la última rama + 1.

La rama Otherwise

La rama `Otherwise` es una alternativa de amplio alcance que especifica cierto tipo de procesamiento (una plantilla) en caso de que ninguna de las ramas definidas tenga el resultado `true`. Sin la rama `Otherwise`, sería necesario crear ramas para todas y cada una de las situaciones posibles o aceptar que el procesador salga de la plantilla condicional sin ejecutar ninguna rama.

Para insertar una rama `Otherwise`, utilice el comando **Agregar rama nueva** o **Copiar rama** (según corresponda) y después en el diálogo Editar expresión XPath 729 marque la casilla *Otherwise* situada en la esquina superior derecha del diálogo (*imagen siguiente*).



Subir y bajar ramas

El orden de las ramas de la condición es importante porque el procesador ejecuta la primera rama cuyo resultado sea `true` y después sale de la condición. Para subir o bajar ramas en la jerarquía de la condición, haga clic con el botón derecho en la rama que desea mover y seleccione **Subir rama** o **Bajar rama**.

Eliminar una rama

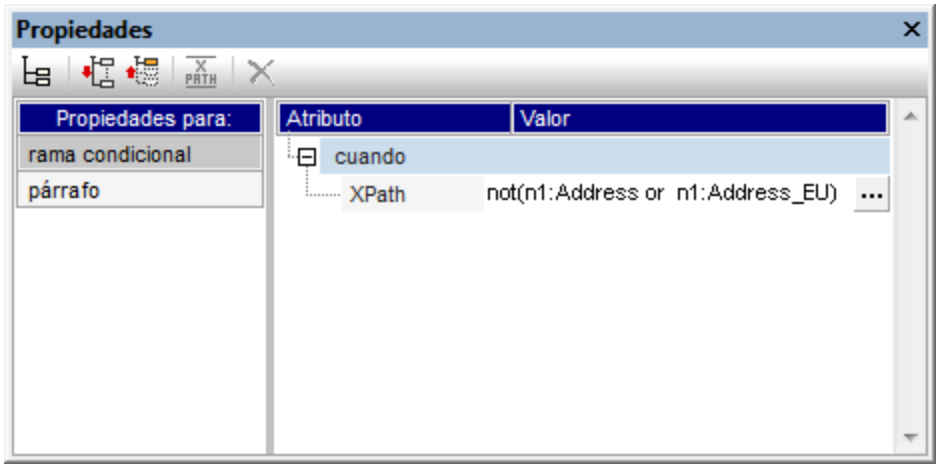
Haga clic con el botón derecho en la rama que desea eliminar y seleccione el comando **Eliminar rama**.

© 2016-2022 Altova GmbH

7.2.2 Editar condiciones

Siga estas instrucciones para editar la expresión XPath de una rama condicional:

1. Seleccione la rama condicional (no la condición).
2. En la ventana Propiedades (columna *Propiedades para:*), seleccione el componente **`rama condicional`** (*imagen siguiente*).



3. Haga clic en el botón **Edición** ... de la propiedad `XPath` (grupo de propiedades *cuando*). Al hacer clic en el botón se abre el diálogo Editar expresión XPath 729, donde puede editar la expresión XPath para esa rama de la condición.

7.2.3 Condiciones basadas en los resultados

Cada componente del documento de diseño puede procesarse de una manera distinta dependiendo del formato de salida (vista Authentic, RTF y HTML). Por ejemplo, en StyleVision puede crear una condición para procesar contenido para la vista Authentic y otra condición para procesar el mismo contenido para el formato de salida HTML. O imagine que quiere incluir parte del texto en la vista Authentic, pero no en el formato de salida HTML. Para ello puede crear una condición con una rama de procesamiento para la vista Authentic y ninguna rama para el formato HTML de salida.

Nota: puede crear condiciones para cada formato de salida alrededor de componentes o partes de componentes. Así conseguirá una gran flexibilidad en la forma en que se estructuran los distintos documentos de salida.

Crear condiciones para cada formato de salida

Siga estas instrucciones para crear condiciones para cada formato de salida:

1. En la vista **Diseño** seleccione el componente (o resalte la parte del documento) al que quiere dar un formato distinto dependiendo del formato de salida.
2. Haga clic con el botón derecho y en el menú contextual seleccione **Incluir en | Condición basada en el resultado**. Esto inserta una condición con cuatro ramas, una por cada formato de salida (vista

Authentic, RTF, texto y HTML) y todas con el mismo contenido: el componente seleccionado. Para ver qué formato de salida corresponde a cada rama, pase el puntero del ratón por la etiqueta de la rama o mire su expresión XPath (en la ventana Propiedades, clic en el componente **`rama condicional`**).

3. Dentro de cada rama pude definir el procesamiento deseado para el componente en el formato de salida de la rama. Si no quiere procesar el componente seleccionado en algún formato de salida, elimine la rama de dicho formato (selecciónela y pulse **Suprimir** o haga clic con el botón derecho en la rama y seleccione **Eliminar rama**).

Nota: si lo prefiere, también puede crear primero la condición basada en el resultado y después insertar el contenido (estático o dinámico) en cada rama.

Editar las ramas de una condición basada en el resultado

La expresión XPath de una rama de una condición basada en el resultado es `$SV_OutputFormat = 'format'`, siendo `format` uno de estos: `Authentic`, `RTF`, `texto` o `HTML`. La expresión XPath de las ramas condicionales se puede editar en la ventana Propiedades (componente **`rama condicional`** y haciendo clic en el botón **Edición** 307). Por ejemplo, puede combinar el formato HTML con la vista Authentic en una sola rama (con la expresión XPath `$SV_OutputFormat = 'Authentic' or $SV_OutputFormat = 'HTML'`).

También puede (a) eliminar ramas de la condición, (b) crear una rama `otherwise` y (c) subir y bajar las ramas para cambiar su prioridad relativa. Para más información sobre cómo realizar estas operaciones, consulte los apartados Definir condiciones 303 y Editar condiciones 307.

Usar el parámetro **`$SV_OutputFormat`**

En el archivo XSLT generado para cada formato de salida se crea el parámetro global `$SV_OutputFormat`, al cual se asigna el valor del formato de salida correspondiente (es decir, `Authentic`, `RTF` o `HTML`). Este parámetro se puede reemplazar con solo pasar otro valor al procesador en tiempo de ejecución. Por ejemplo, imagine que quiere crear dos resultados HTML diferentes, uno de los cuales se seleccionará en tiempo de ejecución. En este caso podría crear las ramas condicionales `$SV_OutputFormat = 'HTML-1'` y `$SV_OutputFormat = 'HTML-2'`. Después en tiempo de ejecución puede enviar al procesador el valor de parámetro necesario (`HTML-1` o `HTML-2`).

7.2.4 Condiciones y cálculos automáticos

Cuando combine condiciones y cálculos automáticos siempre debe tener en cuenta que:

- StyleVision solamente evalúa los cálculos automáticos de **condiciones visibles** (la rama que da `true` como resultado)
- StyleVision evalúa los cálculos automáticos **antes de** evaluar las condiciones.

Por tanto siga siempre estas directrices cuando trabaje con condiciones y cálculos automáticos:

1. Si un cálculo automático actualiza un nodo y dicho nodo forma parte de una condición (bien por ser parte de la expresión XPath de una rama o por ser el contenido de una plantilla condicional), es recomendable que el cálculo automático esté fuera de la condición. Esto garantiza la visibilidad del cálculo automático (independientemente de qué rama condicional sea visible) y la actualización del nodo cada vez que se desencadene el cálculo automático. Si está dentro de una rama no visible, el cálculo automático no se desencadena y el nodo no se actualiza.

 © 2016-2022 Altova GmbH

2. Si no puede evitar colocar el cálculo automático dentro de una condición, entonces asegúrese de que (i) el cálculo está en todas las ramas condicionales y de que (ii) las ramas condiciones cubren todas las situaciones posibles, sin olvidarse de ninguna posibilidad. De lo contrario, existe el riesgo de que, si no está en ninguna plantilla visible, el cálculo automático no se desencadene.
3. Si necesita usar un cálculo automático distinto para cada condición, asegúrese de que se cubren todas las condiciones posibles para cada cálculo automático.
4. Recuerde que el orden en que se definieron las condiciones de una plantilla condicional es importante. La primera condición cuyo resultado sea `true` se ejecuta primero. La condición `Otherwise` es una práctica manera de cubrir todas las posibilidades.

7.3 Presencia condicional

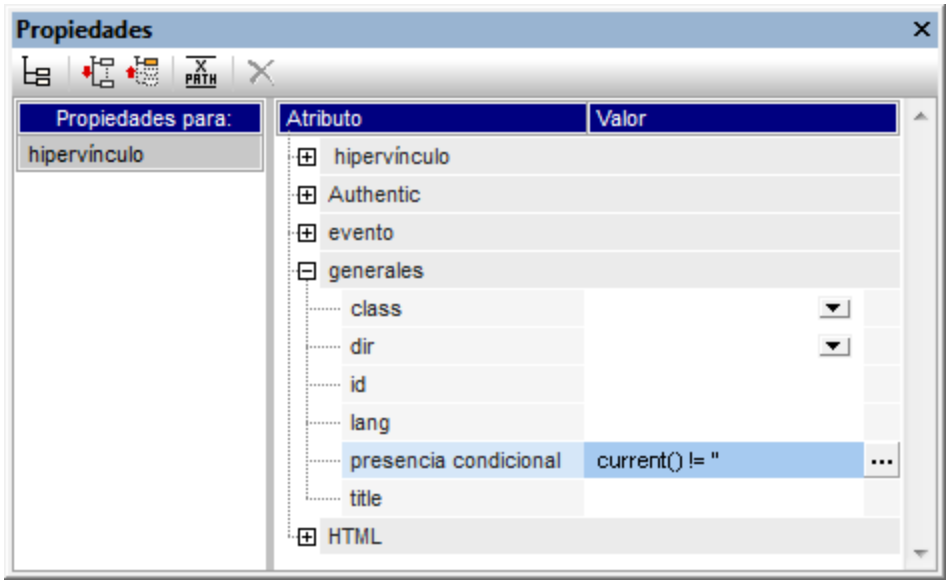
La característica de *presencia condicional* sirve para crear ciertos componentes de diseño solamente si se cumple cierta condición. Estos son los componentes de diseño que se pueden crear si la condición se cumple, es decir, que pueden tener presencia condicional:

- Hipervínculos 366
- Marcadores 364
- Documentos nuevos 285
- Elementos definidos por el usuario 151
- Niveles de TDC 340

Especificar la presencia condicional

Siga estas instrucciones para especificar que un componente debe crearse solamente si se cumple determinada condición:

1. Seleccione el componente de diseño (*ver la lista anterior*).
2. En la ventana Propiedades del componente de diseño (*imagen siguiente*) expanda el grupo de propiedades *Generales* y haga clic en el icono ... de la propiedad `presencia condicional`.



3. Aparece el cuadro de diálogo Editar expresión XPath 729. Introduzca la expresión XPath que define la condición que debe cumplirse para que el componente de diseño se implemente en los documentos de salida.
4. Cuando termine haga clic en **Aceptar**.

Nota: si se cumple la condición, el componente de diseño se implementa. Si no se cumple, el componente no se implementa pero su contenido se genera en los documentos de salida (aunque el componente no esté presente). En el ejemplo de la imagen anterior, por ejemplo, se creó un hipervínculo de presencia condicional. La condición comprueba que el nodo actual no esté vacío. Si el nodo no está vacío, el resultado de la condición es `true` y el hipervínculo se crea en el documento de salida. El texto del hipervínculo se deriva del contenido del componente de diseño `Hipervínculo`. La URL del

hipervínculo se especifica en el grupo de propiedades *Hipervínculo*. Si el resultado de la condición es `false`, el texto del hipervínculo se incluye en el documento de salida, pero como texto sin formato y no como hipervínculo.

Lo mismo ocurre con los demás componentes de diseño que pueden tener presencia condicional. Es el componente el que se crea si la condición se cumple. El contenido del componente de diseño se crea siempre, se cumpla o no la condición.

7.4 Agrupaciones

Las agrupaciones de StyleVision son compatibles con diseños SPS que utilizan **XSLT 2.0 o XSLT 3.0** y con los formatos de salida HTML, texto y RTF. **La vista Authentic no es compatible con esta característica.**

Las agrupaciones sirven para procesar componentes (nodos por lo general) por grupos. Por ejemplo, imagine que tiene un inventario de automóviles. En este inventario hay un elemento llamado `auto` que almacena los datos de cada automóvil. Si el elemento `car` tuviera un atributo `marca`, por ejemplo, podría agrupar los automóviles por marcas. Esto puede ser muy útil en multitud de casos. Por ejemplo:

- Puede reunir todos los automóviles de una misma marca en el documento de salida, bajo el título del nombre de la marca.
- Puede realizar operaciones dentro de un grupo y presentar por separado los resultados de cada grupo. Por ejemplo, puede enumerar por separado el número de modelos disponible de cada marca.

Por último puede procesar un grupo en subgrupos. Por ejemplo, dentro de cada marca, puede agrupar los automóviles por modelo y después por año.

Criterios de agrupación

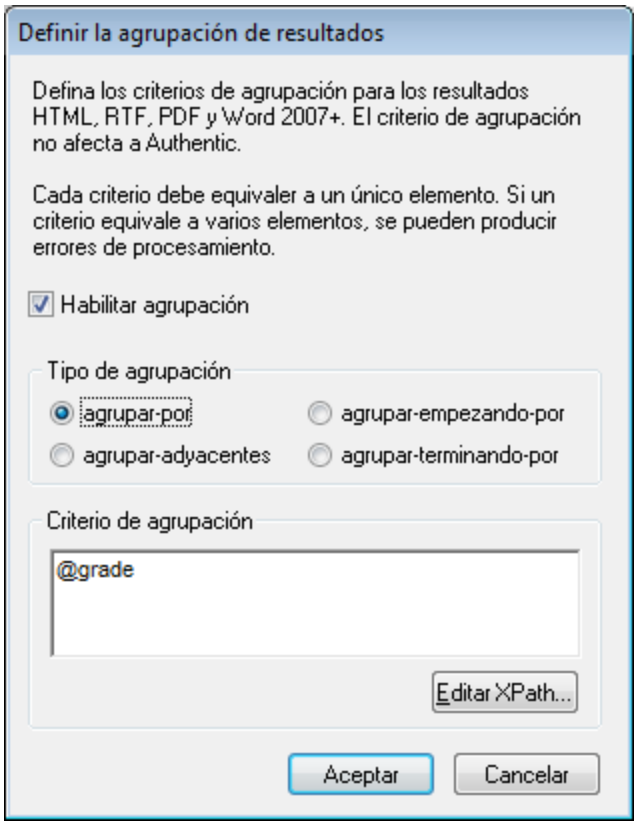
Los componentes se pueden agrupar usando dos criterios generales: (i) una clave de agrupación (que comprueba el valor de un nodo) y (ii) la posición relativa de los componentes. Estos son los criterios de agrupación disponibles:

- ***agrupar-por***: agrupa los componentes en función de una clave definida mediante XPath. Por ejemplo, los elementos `auto` se pueden agrupar en función de su atributo `marca`. La agrupación se establece en el elemento `auto` y una expresión XPath selecciona el atributo `marca`.
- ***agrupar-adyacentes***: combina una clave de agrupación con el criterio de posición. Todos los elementos adyacentes que tengan el mismo valor que la clave de agrupación se incluyen en el mismo grupo. Si el valor de la clave de agrupación no es el mismo que el del elemento anterior, este elemento inicia un grupo nuevo.
- ***agrupar-empezando-por***: empieza un grupo nuevo cuando un nodo coincide con un patrón XPath definido. Si no coincide con el patrón XPath definido, el nodo se asigna al grupo actual.
- ***agrupar-terminando-por***: termina un grupo cuando un nodo coincide con un patrón XPath definido. El nodo que coincide con el patrón es el último nodo del grupo. El nodo siguiente empieza un grupo nuevo. Si el nodo posterior al que comienza un grupo no coincide con el patrón XPath definido, el nodo se asigna al grupo actual.

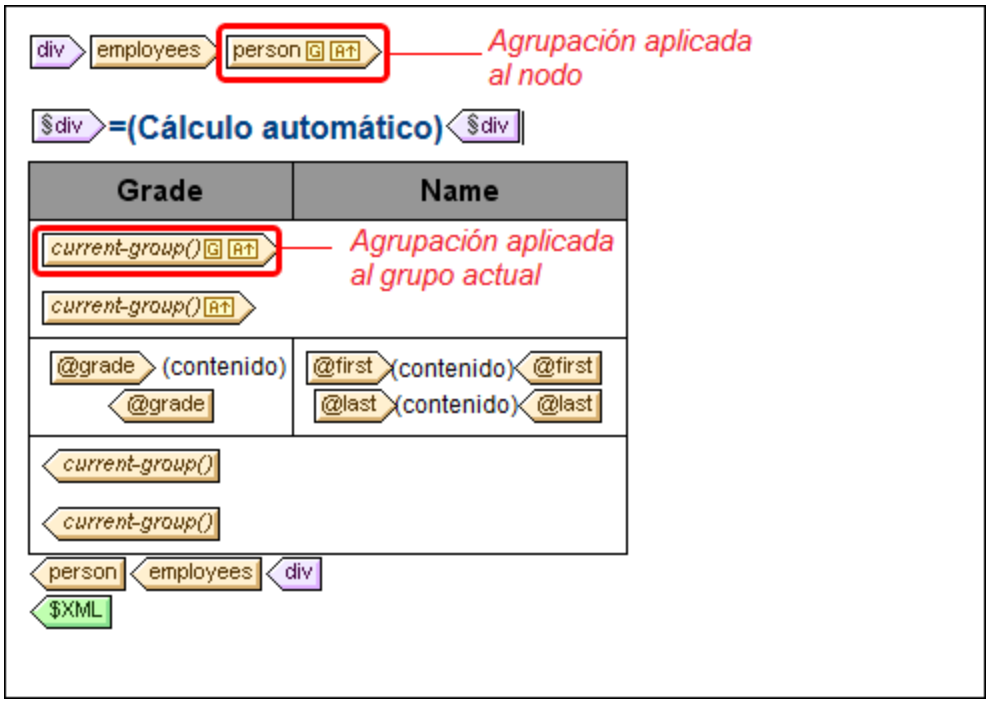
Crear grupos

Puede definir criterios de agrupación tanto en un nodo como en la plantilla *current-group*. Para crear un grupo haga clic con el botón derecho en el nodo o en la plantilla *current-group*. En el menú contextual que aparece seleccione el comando **Agrupar por**. Se abre el cuadro de diálogo "Definir la agrupación de resultados" (*imagen siguiente*).

 © 2016-2022 Altova GmbH



En este cuadro de diálogo marque la casilla *Habilitar agrupación* y después seleccione el *Tipo de agrupación*. En el cuadro de texto *Criterio de agrupación* debe escribir la expresión XPath que define la clave de agrupación (para las opciones *agrupar-por* y *agrupar-adyacentes*) o el patrón de coincidencia (para las opciones *agrupar-empezando-por* y *agrupar-terminando-por*). Al hacer clic en **Aceptar** aparece un cuadro de diálogo que pregunta si desea ordenar el grupo alfabéticamente (en orden ascendente). Si lo prefiere, puede ordenar los grupos más tarde. La imagen siguiente muestra nodos y plantillas *current-group* a los que se aplicaron criterios de agrupación.



En el ejemplo de la imagen anterior, se agrupó el nodo `person` y también los grupos resultantes. Por ejemplo, si los elementos `person` se agrupan por departamento, los departamentos se pueden ordenar por orden alfabético ascendente. Los grupos creados a partir de este criterio se agrupan otra vez creando una agrupación en la plantilla *current-group()*. De este modo puede agrupar los elementos `person` primero por departamento y luego por puesto.

Ordenar grupos

Tras confirmar una definición de agrupación aparece un aviso preguntando si desea ordenar los grupos en orden ascendente o no. Esta ordenación también se puede crear, modificar o eliminar más tarde.

Para definir, modificar o eliminar la ordenación, haga clic con el botón derecho en la plantilla de agrupación y seleccione **Ordenar por**. Aparece el cuadro de diálogo Definir el criterio de ordenación de los resultados 321. En la sección de este manual Ordenación 321 encontrará más información. Lo más importante que debe recordar es que para ordenar grupos en función de su criterio de agrupación, debe seleccionar la función XPath `current-grouping-key()` como criterio de ordenación. Para ver un ejemplo consulte los apartados siguientes.

Ver y editar las opciones de agrupación/ordenación

Para ver y editar las opciones de agrupación/ordenación de una plantilla, haga clic con el botón derecho en la plantilla y seleccione **Agrupar por** o **Ordenar por** respectivamente. Esto abre el respectivo cuadro de diálogo donde puede ver/editar las opciones de configuración.

Plantillas definidas por el usuario

Las plantillas definidas por el usuario 270 son plantillas que se aplican a componentes seleccionados por medio de una expresión XPath definida por el usuario. Los nodos seleccionados por la expresión XPath de una plantilla definida por el usuario también se pueden agrupar. En este caso la agrupación se aplica a la plantilla definida por el usuario.

© 2016-2022 Altova GmbH

7.4.1 Ejemplo: agrupar-por (Persons.sps)

El ejemplo `Persons.sps` se basa en el esquema `Persons.xsd` y utiliza el archivo XML de trabajo `Persons.xml`. El archivo SPS está en la carpeta (Mis) Documentos[34] `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\Grouping\Persons\`. La estructura del documento XML es esta: un elemento de documento `employees` que puede contener un número ilimitado de empleados `person`. Cada empleado `person` tiene una estructura como esta:

```
<person first="Vernon" last="Callaby" department="Administration" grade="C"/>
```

En este diseño los empleados se agrupan por departamentos. Cada departamento se representa en una tabla distinta y los departamentos aparecen en orden alfabético ascendente. Dentro de cada tabla de departamento, los empleados se agrupan por puestos (en orden alfabético ascendente) y, dentro de cada puesto, los empleados aparecen en orden alfabético ascendente según su apellido.

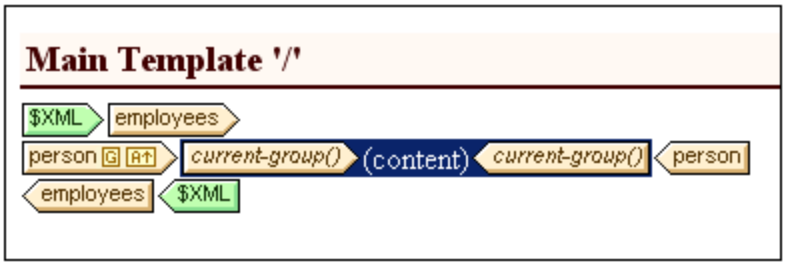
Estrategia

La estrategia para crear estos grupos pasa por definir la agrupación en el elemento `person` y la clave de agrupación en el atributo `department`. Esto permite ordenar los elementos `person` por grupos según el valor del atributo `department`. Si especifica ordenar los grupos, los departamentos aparecerán en orden alfabético ascendente. Como los departamentos deben aparecer en tablas distintas, se crea una tabla con la plantilla *current-grouping* (basada en la clave de agrupación del atributo `department`). Después, dentro de cada grupo ordenado de elementos Person, indicamos que la clave de ordenación es el atributo `grade`.

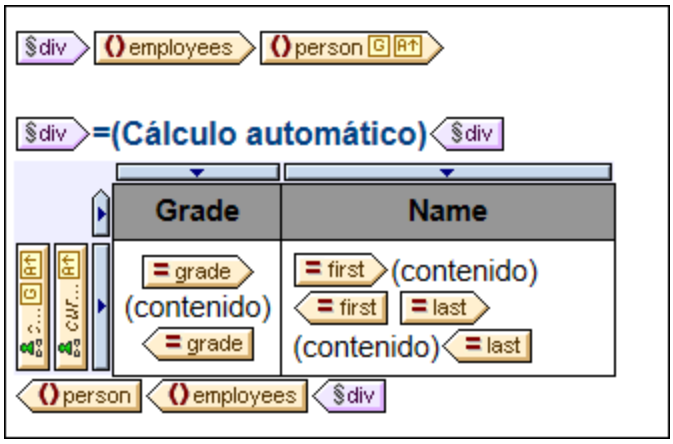
Crear el diseño SPS

El diseño se creó de esta manera:

1. Arrastre el elemento `Person` desde la estructura del esquema y al soltarlo seleccione el comando **Crear contenido**.
2. Haga clic con el botón derecho en la etiqueta del elemento `person` y en el menú contextual seleccione el comando **Agrupar por**.
3. En el cuadro de diálogo "Definir la agrupación de los resultados" seleccione la opción *agrupar-por*, escriba la expresión XPath `@department` en el cuadro de texto *Criterio de agrupación* y después haga clic en **Aceptar**.
4. Aparece un cuadro de diálogo preguntando si los grupos deben ordenarse. Hacemos clic en **Sí** porque queremos que los grupos aparezcan en orden alfabético ascendente (puede establecer, modificar o eliminar la ordenación más tarde si quiere).
5. Como cada grupo (departamento) debe aparecer en una tabla distinta, cree una tabla a partir del grupo actual. Para ello, haga clic con el botón derecho en la etiqueta `current-group()` (*imagen siguiente*) y seleccione el comando **Convertir en | Tabla**. Después seleccione los atributos secundarios `@last` y `@grade` como columnas de la tabla.



6. Reorganice el contenido de las columnas y celdas de la tabla de modo que la primera columna contenga `@grade` y la segunda columna contenga los nodos `@first` y `@last` (*imagen siguiente*).
7. El siguiente paso es agrupar los elementos del grupo actual (nodos agrupados por departamento) por puestos. Para ello creamos una agrupación para el atributo `grade` en la plantilla *current-group()*. Acepte la ordenación predeterminada. Se crea una plantilla *current-group()* nueva (*imagen siguiente*).
8. Ordene este nuevo grupo (que es un subgrupo de personas agrupadas por puesto) por apellidos (atributo `last`).



9. Aplique formato a la tabla.
10. Inserte un título encima de la tabla. Como cada tabla representa un departamento, puede obtener el nombre del departamento de forma dinámica desde el contexto actual. Esto puede hacerse con un cálculo automático que llame a la función `current-grouping-key()` de XPath 2.0/3.0.
11. Repita todo el proceso para crear el mismo resultado pero esta vez agrupe a los empleados primero por su puesto y después por departamento.

Para ver/editar las opciones de agrupación/ordenación de una plantilla, haga clic con el botón derecho en una plantilla y en el menú contextual seleccione **Agrupar por** / **Ordenar por.** En ambos casos aparece el cuadro de diálogo respectivo donde puede ver/editar las opciones.

7.4.2 Ejemplo: agrupar-por (Scores.sps)

El ejemplo `Scores.sps` está basado en el esquema `Scores.xsd` y utiliza el archivo XML de trabajo `Scores.xml`. Está disponible en la carpeta (Mis) Documentos[34], `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\Grouping\Scores\`. La estructura

del documento XML es esta: un elemento de documento `results` contiene un elemento `group` o más y un elemento `match` o más. Cada elemento `match` representa un partido y tiene esta estructura:

```
<match group="A" date="2007-10-12">
       <team name="Brazil" for="2" points="3"/>
       <team name="Germany" for="1" points="0"/>
</match>
```

El diseño se divide en tres partes (*imagen siguiente*): (i) los resultados de los partidos de cada día (los elementos se agrupan por `//match/@date`), (ii) los resultados de los partidos de cada grupo (los elementos se agrupan por `//match/@group`) y (iii) tablas de grupo con un resumen de la posición de cada grupo (una tabla dinámica creada con el elemento `group`, con cálculos automáticos para calcular los datos necesarios).

Match Results: Day-by-Day

2007-10-12

Brazil - Germany	2 - 1
Italy - Holland	2 - 2

2007-10-13

Argentina - France	2 - 0
England - Spain	0 - 0

Match Results: By Group

Group A

Brazil - Germany	2 - 1
Italy - Holland	2 - 2
Brazil - Italy	1 - 2
Germany - Holland	2 - 2
Brazil - Holland	1 - 0
Germany - Italy	1 - 1

Group Tables

Group A

Team	P	W	D	L	F	A	Pts
Brazil	3	2	0	1	4	3	6
Italy	3	1	2	0	5	4	5
Germany	3	0	2	1	4	5	2
Holland	3	0	2	1	4	5	2

Estrategia

Para las dos secciones del diseño que contienen los resultados de los partidos, agrupamos los partidos por fecha y por torneo. Para los miembros de cada grupo (fecha y torneo) creamos tablas sin contorno. Los partidos que se jugaron en una fecha aparecen en una tabla y los resultados de un grupo del torneo aparecen en otra tabla (por ejemplo, los partidos del grupo A). Para la última sección, donde aparece la posición de cada

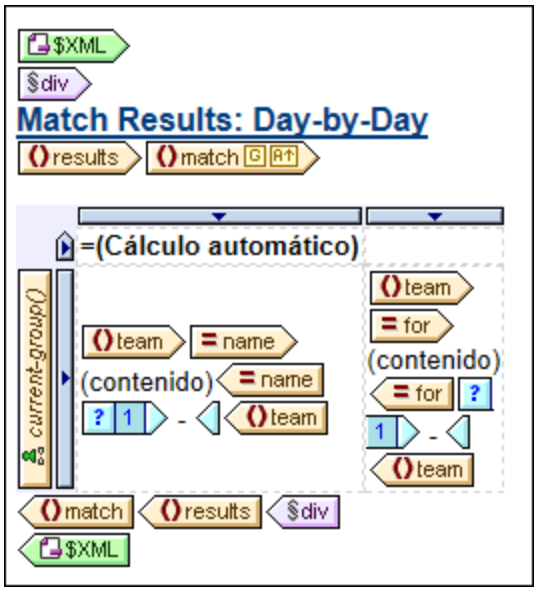
© 2016-2022 Altova GmbH

grupo, creamos una tabla dinámica a partir del elemento `group` donde los cálculos dinámicos generan el valor de los datos necesarios.

Crear el diseño SPS

El diseño se creó así:

1. Arrastre el elemento `/results/match` desde la estructura del esquema y al soltarlo en el diseño seleccione **Crear contenido**.
2. Haga clic con el botón derecho en la etiqueta del elemento `match` y en el menú contextual seleccione **Agrupar por**.
3. En el cuadro de diálogo "Definir la agrupación de los resultados" seleccione la opción *agrupar-por* y en el cuadro de texto *Criterio de agrupación* escriba la expresión XPath `@date`. Después haga clic en **Aceptar**.
4. Aparece un cuadro de diálogo preguntando si se deben ordenar los grupos. Hacemos clic en **Sí** para que los grupos aparezcan en orden alfabético ascendente. (Si quiere, esto se puede definir más tarde.)
5. Como cada grupo (una fecha) debe crearse en una tabla distinta, cree una tabla a partir de la plantilla *current-group*. Para ello haga clic con el botón derecho en la etiqueta de `current-group()`, seleccione el comando **Convertir en | Tabla** y después seleccione los nodos descendientes `team/@name` y `team/@for` como columnas de la tabla (*imagen siguiente*).



6. En cada celda cree un guión, que debe aparecer en el resultado a no ser que el partido sea el último del grupo. Para ello use una plantilla condicional con esta condición `position() != last()`. El resultado de esta plantilla condicional sería `Brazil - Germany` o `2 - 1`, por ejemplo.
7. Inserte un cálculo automático en el título para obtener la clave de agrupación actual para cada grupo (expresión XPath: `current-grouping-key()`).
8. Aplique el formato deseado a la tabla.
9. Para agrupar los partidos por grupos, repita todo el proceso pero agrupe los partidos por el atributo `group` de `match`.
10. Para las tablas de los grupos (en la tercera sección del diseño), que contienen la posición de cada equipo del grupo, cree una tabla dinámica a partir del elemento `/results/group`. Añada las columnas que sean necesarias (con los comandos **Tabla | Anexar columna** o **Tabla | Insertar columna**). Cree cálculos automáticos en cada columna para calcular la posición del equipo (3 puntos por victoria,

1 punto por empate y 0 puntos por derrota). Y para terminar ordene la tabla por orden ascendente según el total de puntos obtenido. Para ver las expresiones XPath utilizadas para obtener estos resultados, haga clic con el botón derecho en el cálculo automático o en la plantilla y seleccione **Editar XPath** u **Ordenar por** respectivamente.

© 2016-2022 Altova GmbH

7.5 Ordenación

La característica *ordenación* es compatible con los formatos de salida HTML texto y RTF. **La vista Authentic no es compatible con esta característica.**

En StyleVision puede ordenar nodos del mismo nivel con el mismo nombre completo según varias claves de ordenación. Por ejemplo, todos los elementos `Persona` (del elemento primario `Compañía`, por ejemplo) se pueden ordenar según su secundario `Apellido`. La clave de ordenación debe ser un nodo del documento y suele ser un nodo descendiente (elemento o atributo) del nodo que deseamos ordenar. En nuestro ejemplo, la clave de ordenación es el elemento secundario `Apellido`.

Si en el conjunto de nodos que intentamos ordenar hay dos elementos con clave de ordenación que tienen el mismo valor, se puede incluir una clave de ordenación más. Por ejemplo, además de la clave de ordenación del secundario `Apellido`, podríamos especificar una clave de ordenación más en el secundario `Nombre`. De este modo, los elementos `Persona` con el mismo `Apellido` se ordenarán según su `Nombre`. Por tanto, en el diseño SPS puede definir varias instrucciones (cada una con una clave de ordenación diferente) para la misma ordenación.

La plantilla se aplica al conjunto ordenado y los resultados se envían ordenados al documento de salida. Puede usar ordenaciones con los formatos de salida HTML y RTF.

Plantillas definidas por el usuario

Las plantillas definidas por el usuario 270 son plantillas que se aplican a elementos/atributos seleccionados con una expresión XPath definida por el usuario. Los nodos seleccionados por la expresión XPath de una plantilla definida por el usuario también se pueden ordenar. En este caso, la ordenación se aplica a la plantilla definida por el usuario.

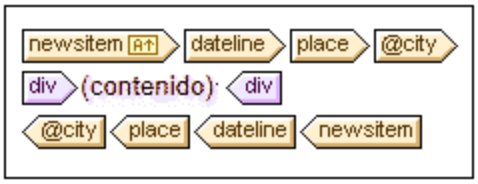
Temas de esta sección

- Descripción del mecanismo de ordenación 321.
- Ejemplo 323 de ordenación.

7.5.1 El mecanismo de ordenación

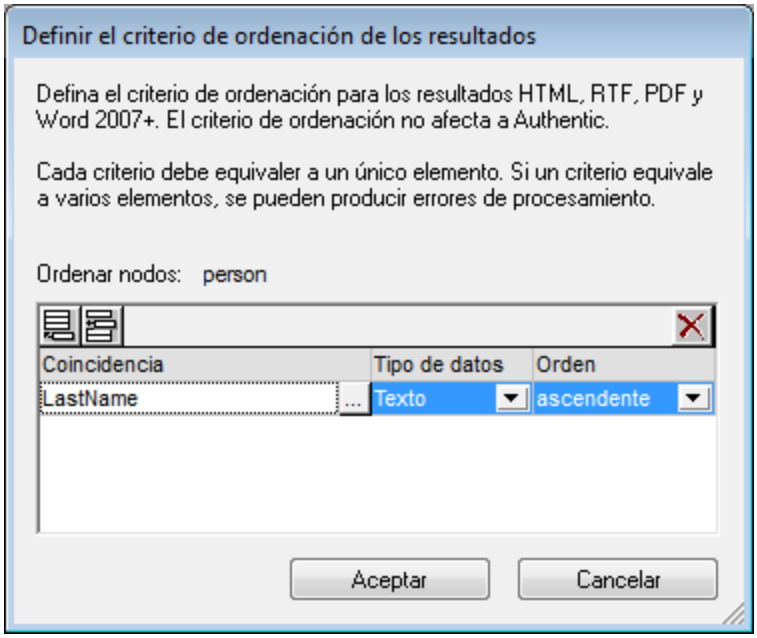
La ordenación de un nodo de elemento del esquema tiene dos pasos:

1. En la vista **Diseño** seleccione el nodo de elemento del esquema que desea ordenar. Recuerde que se ordenarán las instancias de **este** elemento del documento XML. Por ejemplo, observe la estructura que aparece en la imagen siguiente.



Cada elemento `newsitem` tiene un elemento `dateline`, que a su vez tiene un elemento `place` con un atributo `city`. Los nodos `@city` de todos los elementos `newsitem` deben aparecer en orden alfabético en el documento de salida. Si seleccionamos `place` o `dateline` para la ordenación, solamente obtenemos un elemento porque `place` o `dateline` solamente aparecen una vez dentro de su elemento primario. Sin embargo, si seleccionamos `newsitem`, obtenemos varios elementos `newsitem` para ordenar. Es decir, en este caso la ordenación debe definirse en el elemento `newsitem` usando la clave de ordenación `dateline/place/@city`.

2. Tras seleccionar el elemento al que se aplicará la ordenación, haga clic con el botón derecho en el elemento. En el menú contextual que aparece haga clic en **Ordenar por** y se abre el cuadro de diálogo "Definir criterio de ordenación de los resultados" (*imagen siguiente*). En este cuadro de diálogo puede insertar o anexar instrucciones de ordenación.



Cada instrucción de ordenación contiene: (i) una clave de ordenación (en la columna *Coincidencia*), (ii) el *Tipo de datos* que debe usarse para el criterio de ordenación (texto o número) y (iii) el *Orden* (ascendente o descendente). Recuerde que el orden en que define las instrucciones de ordenación es importante porque el mecanismo de ordenación utiliza las instrucciones una a una, empezando por la primera hasta llegar a la última instrucción si hay varios elementos con el mismo valor. Puede definir todas las instrucciones de ordenación que quiera.

Para ver un ejemplo de ordenación consulte el apartado Ejemplo: múltiples claves de ordenación 323.

Plantillas definidas por el usuario

Las plantillas definidas por el usuario[270] son plantillas que se aplican a elementos/atributos seleccionados con una expresión XPath definida por el usuario. Los nodos seleccionados por la expresión XPath de una plantilla definida por el usuario también se pueden ordenar. En este caso, la ordenación se aplica a la plantilla definida por el usuario.

Nota sobre las claves de ordenación

La expresión XPath de la clave de ordenación debe seleccionar **un solo nodo** por cada instancia de elemento y no un conjunto de nodos (XPath 1.0) o una secuencia de elementos (XPath 2.0/3.0). La clave para cada elemento debe dar como resultado un valor de cadena o un valor numérico.

En los diseños SPS que usan **XSLT 2.0 o XSLT 3.0**, si la clave de ordenación devuelve una secuencia de nodos, se emite un error de procesamiento XSLT. Por ejemplo, con el nodo de contexto `Person`, la expresión XPath `../Person/LastName` devolvería un error porque esta expresión devuelve todos los elementos `LastName` que forman parte del primario `Person` (suponiendo que haya más de un elemento `Person`). La expresión XPath correcta, con `Person` como nodo de contexto, sería `LastName` (porque solamente hay un nodo `LastName` por cada elemento `Person`).

En los diseño SPS que usan **XSLT 1.0**, la especificación XSLT 1.0 exige que cuando el selector de la clave de ordenación devuelva un conjunto de nodos, debe utilizarse el valor de texto del primer nodo. Por tanto, StyleVision no devuelve ningún error si la expresión XPath selecciona varios nodos para la clave de ordenación. StyleVision utiliza el texto del primer nodo e ignora los demás. Sin embargo, puede que el primer nodo seleccionado no sea la clave de ordenación deseada. Por ejemplo, la expresión XPath `../Person/LastName` del ejemplo anterior no devolvería un error pero tampoco realizará la ordenación porque es el mismo valor para cada elemento del bucle de ordenación (el valor de texto del primer nodo `LastName`). Sin embargo, una expresión del estilo `location/@*`, realizará la ordenación, usando el primer atributo del secundario `location` como clave de ordenación. De todas maneras debería evitar este tipo de expresiones porque es mejor seleccionar el criterio de ordenación con mayor precisión (seleccionando un solo nodo).

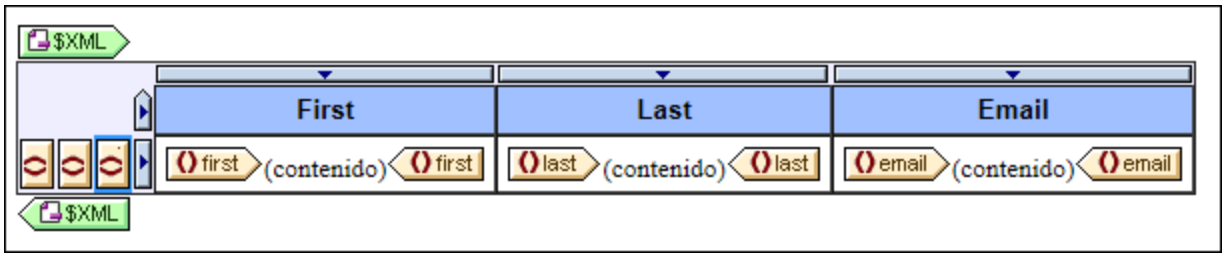
7.5.2 Ejemplo: múltiples claves de ordenación

En este ejemplo (ver carpeta (Mis) Documentos[34], `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\Sorting\SortingOnTwoTextKeys.sps`) se enumeran en una tabla los miembros de un equipo. Cada miembro aparece en una fila de la tabla, con su nombre, apellido y dirección de correo electrónico. Imagine que queremos ordenar la lista por orden alfabético, primero por apellido y después por nombre.

Cuando la lista está todavía sin ordenar, el orden de los elementos en el documento de salida es el orden que tienen los elementos `member` en el documento XML (*imagen siguiente, formato HTML*).

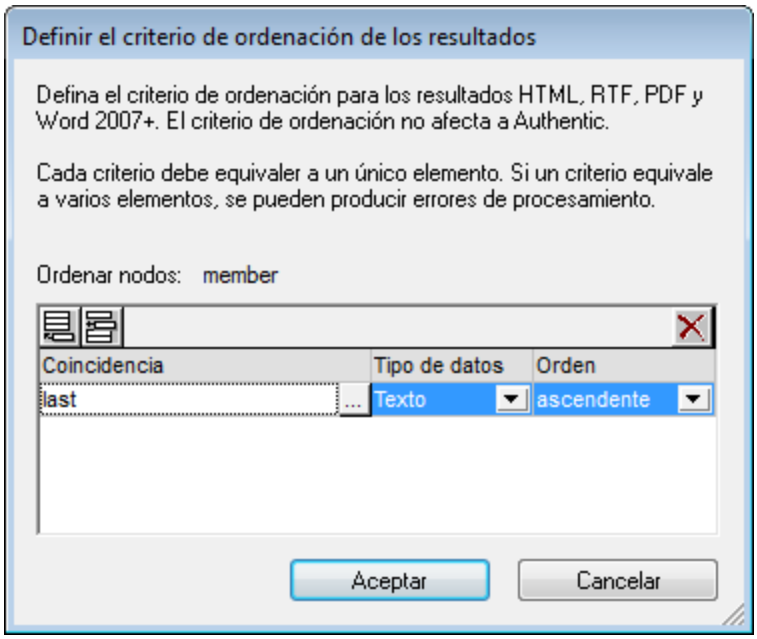
First	Last	Email
Andrew	Bentinck	a.bentinck@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com
John	Edwards	j.edwards@nanonull.com
Janet	Ashe	j.ashe@nanonull.com

En la vista **Diseño**, haga clic con el botón derecho en el elemento `member` (*resaltado en azul en la imagen siguiente*) y seleccione **Ordenar por** en el menú contextual.



Se abre el cuadro de diálogo "Definir el criterio de ordenación de los resultados" (*imagen siguiente*). Observe que el elemento seleccionado para la ordenación (`member`) aparece en el cuadro de diálogo como *Ordenar nodos:* `member`. Este nodo también es el nodo de contexto para las expresiones XPath que se usarán para seleccionar la clave de ordenación. Haga clic en el botón **Agregar fila** (esquina superior izquierda del panel) para añadir la primera instrucción de ordenación. En la columna *Coincidencia* introduzca una expresión XPath para seleccionar el nodo `last`. También puede hacer clic en el botón ... para abrir el cuadro de diálogo "Editar expresión XPath". En la columna *Tipo de datos* puede seleccionar cómo debe evaluarse el contenido de la clave de ordenación: como texto o como número. Y la columna *Orden* ofrece dos opciones: ascendente y descendiente. Seleccione `Texto` y `Ascendente`. Para terminar haga clic en **Aceptar**.

© 2016-2022 Altova GmbH

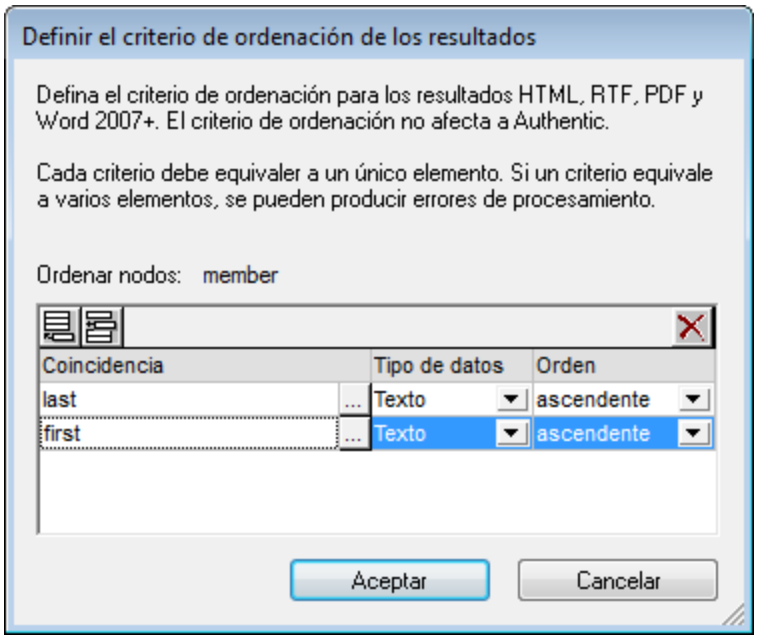


En la vista **Diseño** la etiqueta del elemento `member` ahora incluye un icono para indicar que contiene un filtro de ordenación (

). A continuación puede ver la misma lista en formato HTML con los miembros del equipo ordenados por apellido. Observe que hay dos personas con el apellido Edwards pero no están ordenadas alfabéticamente (Nadia aparece antes que John, siguiendo el orden de los elementos en el documento XML). Por tanto necesitamos aplicar otra clave de ordenación, esta vez en el nombre (`first`).

First	Last	Email
Janet	Ashe	j.ashe@nanonull.com
Andrew	Bentinck	a.bentinck@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com
John	Edwards	j.edwards@nanonull.com

En la vista **Diseño** haga clic con el botón derecho en la etiqueta del elemento `member` y seleccione **Ordenar por** en el menú contextual. En el cuadro de diálogo "Definir el criterio de ordenación de los resultados" anexe una fila nueva al panel y defina el elemento `first` como clave de ordenación (*imagen siguiente*). Para terminar haga clic en **Aceptar**.



A continuación puede ver cómo queda la lista en formato HTML, con todos los miembros del equipo ordenados por apellido y después por nombre.

First	Last	Email
Janet	Ashe	j.ashe@nanonull.com
Andrew	Bentinck	a.bentinck@nanonull.com
John	Edwards	j.edwards@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com

© 2016-2022 Altova GmbH

7.6 Parámetros y variables

En el archivo SPS puede declarar parámetros y variables y hacerles referencia. La diferencia entre estos dos mecanismos es que, mientras que el valor de una variable se define en el momento de declaración, a los parámetros les puede pasar un valor (en tiempo de ejecución desde StyleVision Server) que reemplace el valor predeterminado opcional asignado en el momento de declaración.

En esta sección describimos las funciones relacionadas con parámetros y variables:

- Parámetros declarados por el usuario 327: cómo usar parámetros definidos por el usuario en un diseño SPS.
- Parámetros para fragmentos de diseño 329: cómo usar parámetros con fragmentos de diseño.
- Parámetros del SPS para fuentes de datos 331: se trata de un tipo de parámetro especial. StyleVision los define automáticamente para los esquemas fuente (concretamente para los archivos XML de trabajo de los esquemas). Como el usuario conoce nombre y el valor de estos parámetros, puede hacerles referencia dentro del SPS y pasarles un valor en tiempo de ejecución desde StyleVision Server.
- Variables 332: puede (i) declarar una variable en cierto ámbito y definir su valor y (ii) hacer referencia al valor de variables declaradas y crear una plantilla en un nodo o en nodos seleccionados por la variable.

7.6.1 Parámetros declarados por el usuario

En los diseños SPS los parámetros declarados por el usuario se declaran globalmente con un nombre y un valor de cadena predeterminado. Una vez declarados, los parámetros se pueden usar en expresiones XPath en cualquier posición del diseño. El valor predeterminado del parámetro se puede reemplazar en las diferentes transformaciones XSLT pasando un valor global nuevo a la hoja de estilos XSLT desde StyleVision Server.

Uso de los parámetros

Los parámetros declarados por el usuario son muy prácticos en estas situaciones:

- Si quiere usar el mismo valor en varias partes o como entrada de varios cálculos. En este caso puede guardar el valor necesario como valor de parámetro y usar el parámetro en las posiciones y en los cálculos que quiera.
- Si quiere pasar un valor a la hoja de estilos en tiempo de procesamiento. En el SPS (y en la hoja de estilos) utilice un parámetro con un valor predeterminado. Después puede pasar el valor deseado al parámetro desde StyleVision Server en tiempo de procesamiento.

Utilización

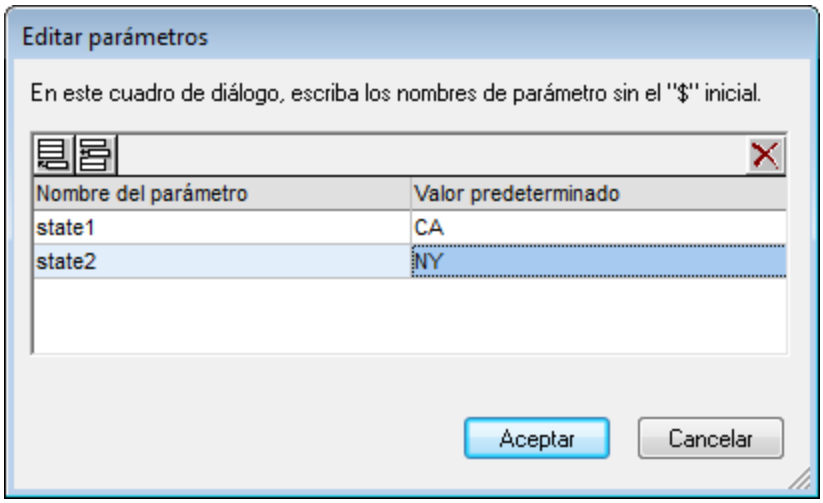
El trabajo con parámetros declarados por el usuario tiene dos fases:

1. Declarar los parámetros necesarios 327.
2. Hacer referencia a los parámetros declarados 328.

Declarar parámetros

Todos los parámetros definidos por el usuario se declaran y se editan en el cuadro de diálogo "Editar parámetros" (*imagen siguiente*). Para abrir este cuadro de diálogo haga clic en el comando **Edición |**

Parámetros de la hoja de estilos 799 o en el botón **Parámetros...** del cuadro de diálogo "Editar los filtros de la base de datos" (**Base de datos | Editar filtro de la base de datos** 861).



Para declarar un parámetro hay que darle un nombre y un valor de cadena, es decir, su valor predeterminado. Si se deja vacío, el valor predeterminado es una cadena vacía.

Siga estas instrucciones para declarar un parámetro:

1. En el cuadro de diálogo "Editar parámetros" anexe o inserte un parámetro nuevo haciendo clic en los botones **Anexar** o **Insertar**.
2. En la fila que aparece escriba el nombre del parámetro. Los nombres de parámetro deben comenzar con una letra y pueden incluir los caracteres `A` - `Z`, `a` - `z`, `0` - `9` y el carácter de subrayado.
3. Introduzca un valor predeterminado para el parámetro. El valor se acepta como cadena de texto.

En este cuadro de diálogo puede insertar tantos parámetros como quiera y modificar los parámetros existentes en cualquier momento mientras edita el diseño SPS o la vista Authentic.

Nota:

- El cuadro de diálogo "Editar parámetros" contiene todos los parámetros definidos por el usuario del diseño SPS.
- Los parámetros también se pueden declarar en la ventana Vista general del diseño 45.

Hacer referencia a los parámetros declarados

Para hacer referencia a un parámetro desde una expresión XPath añada el prefijo `$` al nombre del parámetro. Por ejemplo, puede hacer referencia a un parámetro en la expresión XPath de un cálculo automático (p. ej. `concat('www.', $company, '.com')`). Si el diseño SPS está basado en una BD, también puede usar parámetros como valor del criterio de filtrado de la BD. Sin embargo, los parámetros se declaran y editan 327 en el cuadro de diálogo Editar parámetros 799.

Nota: hacer referencia a un parámetro no declarado es un error pero no lo es declarar un parámetro y no hacerle referencia.

© 2016-2022 Altova GmbH

7.6.2 Parámetros para fragmentos de diseño

En StyleVision puede definir un parámetro en un fragmento de diseño creado previamente y dar al parámetro un valor predeterminado. Después puede modificar el valor del parámetro en cada instancia del fragmento de diseño, lo cual permite personalizar los resultados del fragmento de diseño en cada posición.

Por ejemplo, podemos crear un fragmento de diseño llamado **DireccionesCorreo** con un parámetro llamado `Dominio` que tenga el valor predeterminado `altova.com`. Imagine que este parámetro se usa en un cálculo automático del fragmento de diseño para generar las direcciones de correo electrónico de los empleados de la compañía. Para las direcciones de la EU, podemos usar el fragmento de diseño **DireccionesCorreo** y cambiar el valor del parámetro `Domain` por `altova.eu`. Igualmente, en la plantilla que se usará para los empleados japoneses, podemos cambiar el valor del parámetro `Domain` por `altova.jp`. Y para los empleados estadounidenses, podemos dejar el valor del parámetro `Domain` como está (con el valor predeterminado `altova.com`).

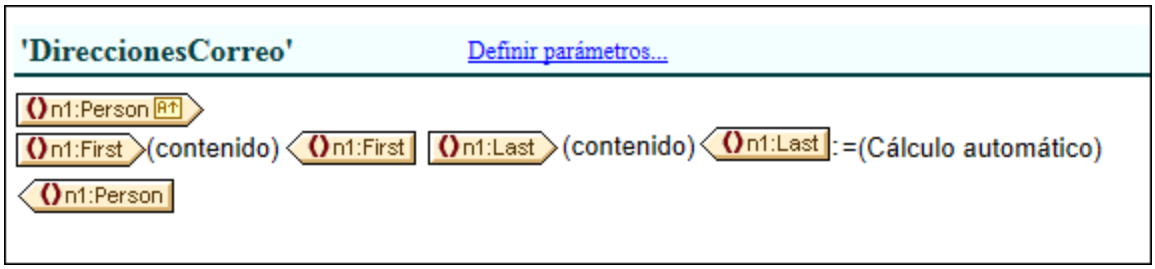
El trabajo con parámetros para fragmentos de diseño tiene dos fases:

1. Definir el parámetro 329 con un valor predeterminado en el fragmento de diseño.
2. Editar el valor del parámetro 330 allá donde se utilice el fragmento de diseño.

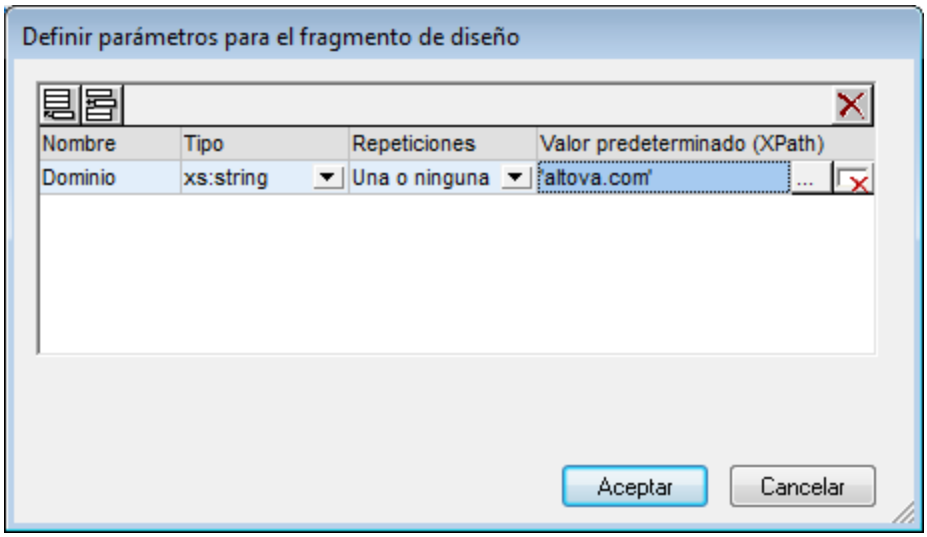
Nota: los parámetros para fragmentos de diseño son compatibles con la vista Authentic solamente en las ediciones Enterprise Edition de los productos de Altova

Definir el parámetro

En cada fragmento de diseño puede definir un número ilimitado de parámetros. Para ello haga clic en el vínculo `Definir parámetros...` que aparece en la barra de título del fragmento de diseño (*imagen siguiente*).



Se abre el cuadro de diálogo "Definir parámetros para el fragmento de diseño" (*imagen siguiente*). Haga clic en el icono **Anexar** o **Insertar** (esquina superior izquierda) para añadir un parámetro a la lista. Escriba o seleccione su nombre, su tipo de datos, el número de veces que debe aparecer y su valor predeterminado. El atributo *Repeticiones* del parámetro especifica cuántos elementos se devuelven al evaluar la expresión XPath indicada como valor predeterminado del parámetro. El atributo *Repeticiones* es opcional y su valor predeterminado es **Una o ninguna**. En este diálogo puede crear todos los parámetros que quiera.

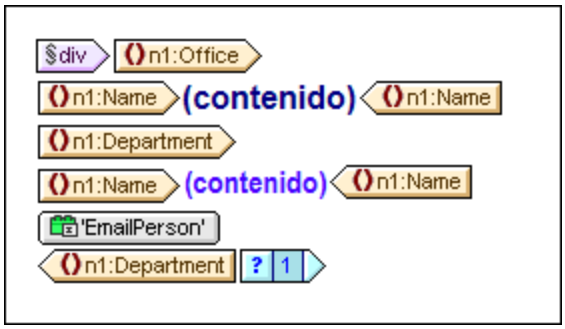


Observe que el cuadro de diálogo ofrece dos iconos **Eliminar**. El icono **Eliminar** situado al final de la fila de cada parámetro elimina el valor predeterminado del parámetro. El icono **Eliminar** situado en la esquina superior derecha del panel elimina el parámetro seleccionado en la tabla.

Nota: si el SPS utiliza XSLT 1.0, la expresión XPath que introduzca debe devolver un conjunto de nodos. De lo contrario se emite un error.

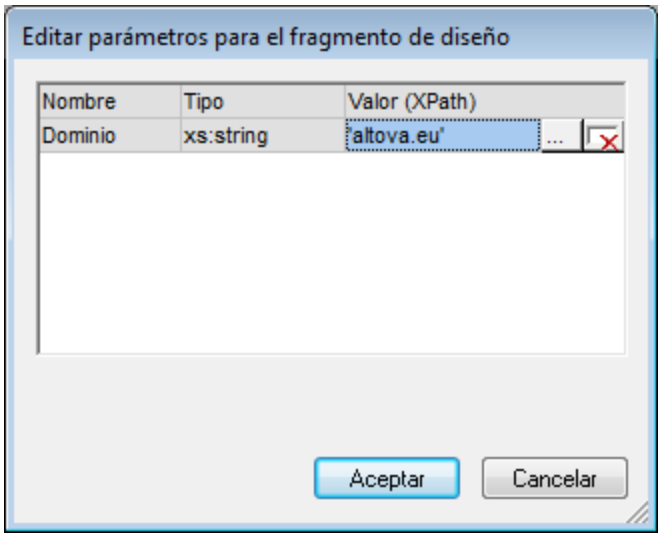
Usar el parámetro

Una vez creado, el fragmento de diseño se puede insertar en varias posiciones del diseño (arrastrándolo desde la Estructura del diseño o desde la Estructura del esquema). La imagen siguiente, por ejemplo, muestra cómo se insertó el fragmento de diseño **`EmailPerson`** después del elemento `n1:Name`.



Si definió un parámetro para este fragmento de diseño, puede editar su valor en esta instancia concreta del fragmento. Para ello haga clic con el botón derecho en el fragmento de diseño y seleccione el comando **Editar parámetros**. Se abre el cuadro de diálogo "Editar parámetros para el fragmento de diseño" (*imagen siguiente*).

 © 2016-2022 Altova GmbH



En este cuadro de diálogo puede editar el valor del parámetro. Cuando termine haga clic en **Aceptar** y el nuevo valor de parámetro se usará en esta instancia del fragmento de diseño. Si el valor de parámetro no se cambia, se usará el valor original (predeterminado) del parámetro.

Nota: si el SPS utiliza XSLT 1.0, la expresión XPath que introduzca debe devolver un conjunto de nodos. De lo contrario se emite un error.

7.6.3 Parámetros del SPS para fuentes de datos

Un diseño SPS puede tener varios esquemas fuente, como una DTD o un esquema XML en el que se basa el documento XML o un esquema XML generado a partir de una BD y en el que se basa la BD.

Todos los diseños SPS tienen un esquema principal y, opcionalmente, varios esquemas adicionales. Cuando se añade un esquema fuente nuevo al diseño SPS, StyleVision declara automáticamente un parámetro para el esquema y le asigna un valor que es el URI del archivo XML de trabajo asignado a dicho esquema. En los diseños basados en una BD, StyleVision genera un archivo XML temporal a partir de la BD y configura el parámetro para apuntar al nodo de documento de este archivo XML temporal.

Hacer referencia a parámetros para esquemas fuente

Cada parámetro para un esquema fuente apunta al nodo de documento del archivo XML que corresponde al esquema. En StyleVision el archivo XML de cada esquema es el archivo XML de trabajo o el archivo XML generado a partir de una BD. Por tanto, puede usar los parámetros del SPS para fuentes de datos de dos maneras:

1. En expresiones XPath del SPS para encontrar nodos en diferentes documentos. El parámetro se utiliza para identificar el documento y los pasos de la expresión XPath encuentran el nodo necesario dentro de dicho documento. Por ejemplo, la expresión: `count($XML2//Departamento/Empleado)` devuelve el número de elementos `Empleado` de todos los elementos `Departamento` del archivo XML de trabajo asignado al esquema fuente designado con `$XML2`.
2. En StyleVision Server puede pasar el URI de otro archivo XML como valor de un parámetro del SPS para fuentes de datos. Ahora bien, el archivo XML nuevo debería estar basado en el esquema que representa el parámetro. Por ejemplo, si `ArchivoA.xml` y `ArchivoB.xml` son archivos válidos según el

mismo esquema y `ArchivoA.xml` es el archivo XML de trabajo asignado a un esquema `$XML3` utilizado en un diseño SPS, cuando se invoque una transformación XSLT para dicho SPS desde StyleVision Server, `ArchivoB.xml` puede reemplazar a `ArchivoA.xml` usando el parámetro `$XML3="FileB.xml"`. Recuerde también que en StyleVision Server puede introducir valores para todos los parámetros del SPS para fuentes de datos excepto para el esquema principal. El archivo XML correspondiente al esquema principal será el punto de entrada de la hoja de estilos XSLT y será por tanto el archivo XML en el que se ejecuta la transformación.

7.6.4 Variables

El trabajo con variables tiene dos fases: (i) declarar la variable 332 y (ii) usarla 334.

Nota: las variables son compatibles con la vista Authentic solo en las ediciones Enterprise Edition de los productos de Altova.

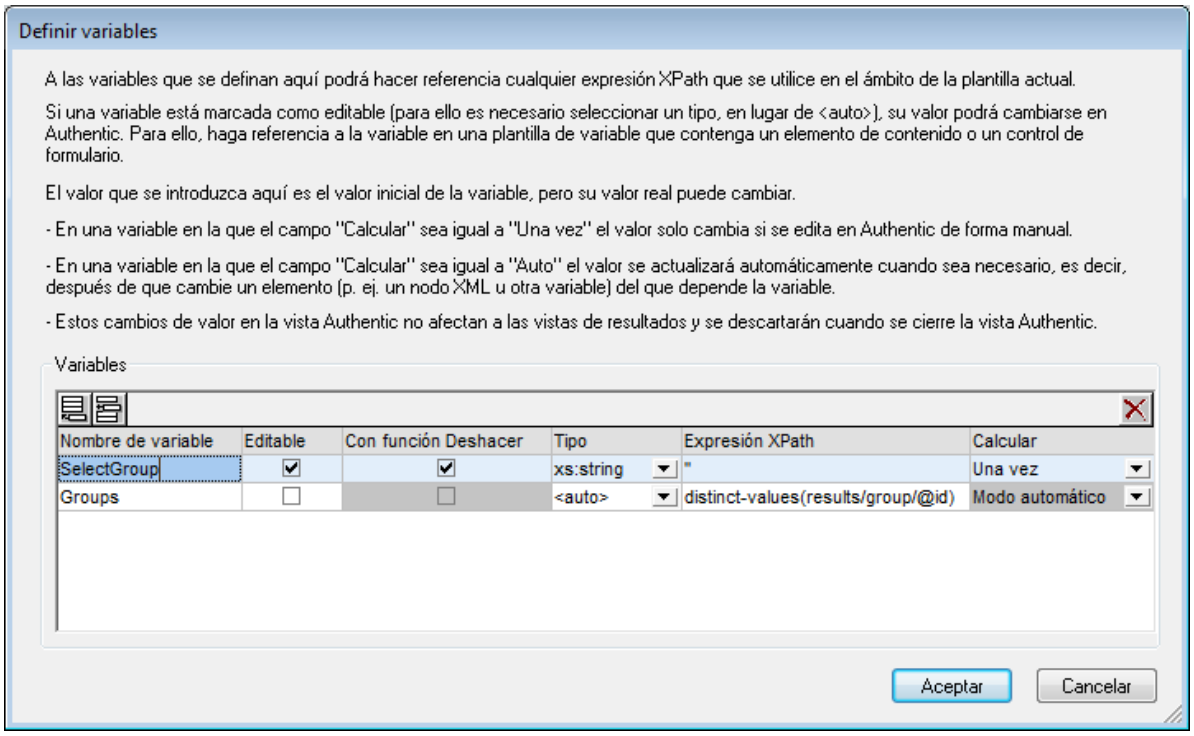
Declarar la variable

Puede declarar variables en cualquier plantilla del diseño. A cada variable se le asigna un nombre, un tipo de datos y un valor. Además puede especificar si la variable se puede editar o no en la vista Authentic (solo en las ediciones Enterprise Edition). Una vez definida, la variable estará en ámbito en esta plantilla y se puede usar en ella. Si quiere que la variable esté en el ámbito de todo el documento, declare la variable en la plantilla raíz. La ventaja de declarar una variable en la parte de la plantilla donde se necesita es que las expresiones XPath pueden ser más sencillas.

Siga estos pasos para declarar una variable:

1. Haga clic con el botón derecho en la plantilla de nodo donde desea crear la variable y seleccione el comando **Definir variables**.
2. Aparece el cuadro de diálogo "Definir variables" (*imagen siguiente*). Haga clic en el icono **Anexar variable** (esquina superior izquierda del panel) e introduzca un nombre de variable. El valor de la variable se define con una expresión XPath. Si prefiere darle un valor de cadena (como en la primera variable de la imagen siguiente), ponga la cadena de texto entre comillas. En la imagen siguiente, por ejemplo, el valor de la variable `SelectGroup` es una cadena de texto vacía. Si no usa las comillas el texto se leerá como un nombre de nodo o llamada a función.

 © 2016-2022 Altova GmbH



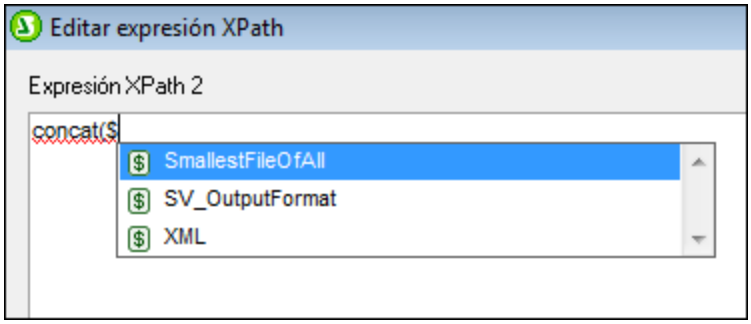
3. Si marca la variable como editable (segunda columna), la variable se podrá editar en la vista Authentic [334]. Si marca la variable como editable, también debe darle el tipo de datos correcto, como `xs:string`, por ejemplo. Si la variable es editable, el usuario de la vista Authentic podrá cambiar el valor de la variable que definió el diseñador del documento SPS. Estos cambios pueden ser como resultado de editar explícitamente la variable (p. ej. si el usuario edita el contenido de un cuadro de texto editable) o cuando el usuario de la vista Authentic modifica un nodo o valor que forma parte de la expresión XPath de la variable.
4. Si marca la variable como editable, se habilitan dos opciones más en el cuadro de diálogo, relacionadas con la vista Authentic: la opción *Con función Deshacer* y la opción *Calcular*. Si marca la opción *Con función Deshacer*, se generará un paso Deshacer por cada cambio realizado en la variable. De este modo el usuario de la vista Authentic podrá retroceder hasta recuperar valores anteriores de la variable. El valor de la opción *Calcular* puede ser `Una vez` or `Modo automático`. Si elige la opción `Una vez`, el valor de la variable se calcula una sola vez: cuando se evalúa la plantilla que contiene la variable. El valor de la variable cambia solamente cuando el usuario la edita explícitamente. Por el contrario, si elige la opción `Modo automático`, la variable se vuelve a calcular cada vez que se modifique un nodo o valor utilizado en la expresión XPath de la variable.
5. Puede definir todas las variables que quiera, pero en el mismo ámbito no puede haber dos variables con el mismo nombre. Para eliminar una variable, selecciónela y haga clic en el icono **Eliminar** (esquina superior derecha del panel).
6. Cuando termine haga clic en **Aceptar**. La etiqueta de la plantilla tiene ahora el icono $ para indicar que tiene variables.

Siguiendo estas instrucciones podrá crear variables para todas las plantillas de nodo del diseño. Cada variable tendrá un nombre y un valor y estará en el ámbito de la plantilla donde se declaró. Para editar una variable más adelante, haga clic con el botón derecho en la plantilla donde creó la variable y seleccione el comando **Definir variables**.

Usar una variable

Para poder usar una variable en cualquier posición, la variable debe estar en el ámbito de la posición. Esto significa que la variable solamente se puede usar dentro de la plantilla donde se definió. Las variables también se pueden editar en la vista Authentic para que el usuario pueda controlar su presentación. Cuando se cierra el SPS se desecha el valor editado.

Puede usar las variables definidas en cualquier expresión XPath y para hacerle referencia basta con añadir el prefijo `$` al nombre de la variable. Por ejemplo, la expresión XPath `$VarName/Name` selecciona el elemento secundario `Name` del nodo seleccionado por la variable llamada `VarName`.



Cuando empiece a teclear la expresión XPath en el cuadro de diálogo Editar expresión XPath 729 las variables que están en el ámbito aparecen en una lista emergente (*imagen anterior*). Seleccione una variable y pulse **Entrar** para insertar la referencia a la variable en la expresión.

7.6.5 Variables editables en Authentic

El usuario de la vista Authentic puede editar las variables que están en ámbito para controlar el aspecto de la vista Authentic. Por ejemplo, si se utiliza un documento XML de gran tamaño que contiene los datos del personal de varias oficinas de una compañía, podemos dar al usuario de la vista Authentic la opción de seleccionar una oficina concreta. La vista Authentic del documento mostrará solamente la oficina seleccionada por el usuario de la vista Authentic.

Funcionamiento

Para configurar una variable editable en la vista Authentic son necesarios tres pasos (*imagen siguiente*):

1. La variable se define en la plantilla donde se utilizará. Esta plantilla delimita el ámbito de la variable. La variable solamente se puede usar dentro de la plantilla en la que está en ámbito.
2. Se crea una plantilla definida por el usuario con el nombre de la variable. El contenido dinámico de esta plantilla definida por el usuario puede incluir el valor de la variable. Si se inserta el marcador de posición `(contenido)` o un campo de entrada dentro de la plantilla definida por el usuario, el usuario de la vista Authentic podrá insertar cualquier contenido como valor de la variable. Sin embargo, puede limitar las opciones que tiene el usuario de la vista Authentic si inserta un control de formularios (p. ej. un cuadro combinado) dentro de la plantilla definida por el usuario.
3. La variable se puede usar en una expresión XPath para controlar el aspecto de la vista Authentic. Por ejemplo, puede usar la variable en una condición. Dependiendo del valor de cada rama condicional, la

 © 2016-2022 Altova GmbH

vista tendrá un aspecto u otro. O puede usar la variable en una expresión de un modelo de coincidencia (`template match`) o en un filtro de plantilla.

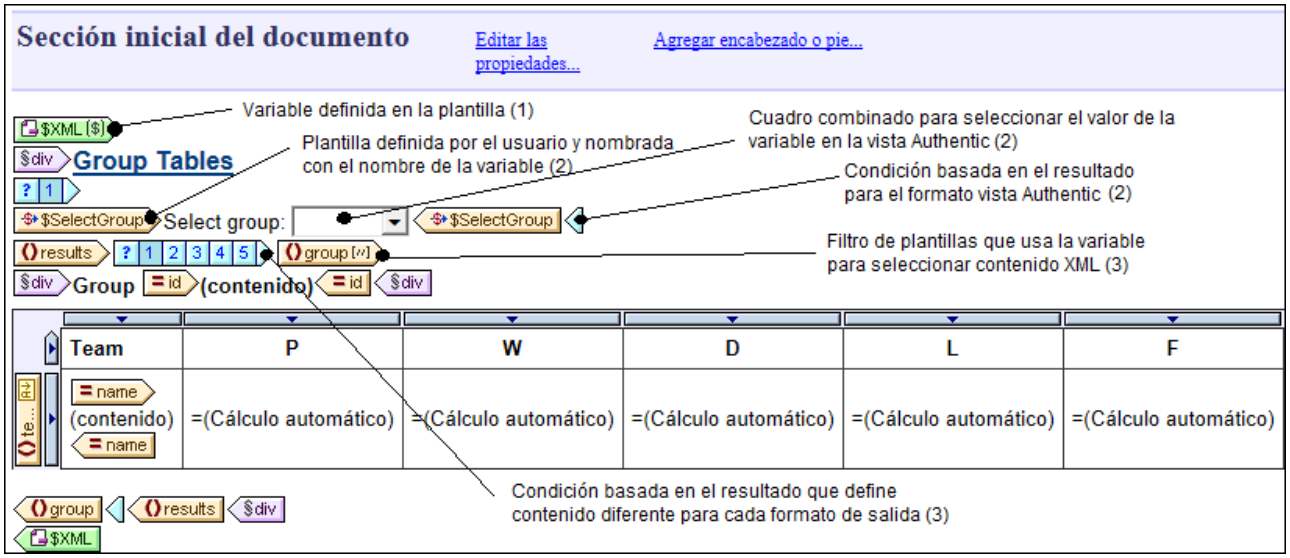
Notas:

- Como la vista Authentic no es compatible con los mecanismos de agrupación ni ordenación, las variables editables de la vista Authentic no se pueden usar en diseños SPS que contengan estos mecanismos.
- Como las variables editables solamente afectan a la vista Authentic, el diseño para esta vista tendrá que ser distinto al de los demás formatos de salida. Esto se consigue fácilmente gracias a las condiciones basadas en el resultado 307.

Archivo de ejemplo

El archivo `AuthenticVariables.sps` de la carpeta (Mis) Documentos 34, `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\Grouping\Scores\`, muestra cómo se pueden usar las variables editables en la vista Authentic. El archivo XML contiene los resultados de todos los partidos de un torneo. Los equipos que participan en el torneo se dividen en dos grupos. La variable editable permite al usuario de la vista Authentic seleccionar un grupo y ver los resultados de ese grupo solamente.

La imagen siguiente muestra la vista **Diseño** del SPS. Se creó una variable editable y los pasos necesarios para crearla se señalan en la imagen. Siga leyendo para ver cómo se creó este diseño paso a paso.



En este diseño SPS los pasos clave fueron:

- En la plantilla `$XML` (la plantilla raíz) se definió una variable editable llamada `SelectGroup` que es de tipo `xs:string`. Esta variable estará en ámbito en toda la plantilla.
- En la plantilla `$XML` también se declaró una variable no editable 332 llamada `Groups` que es de tipo `<auto>`. Su objetivo es recopilar dinámicamente los valores de todos los atributos `results/group/@id`. La idea es mostrar los diferentes valores de este atributo como opciones en una lista desplegable de un cuadro combinado 194 donde el usuario de la vista Authentic podrá elegir el grupo que desea ver.

- Después se creó una plantilla definida por el usuario 270 llamada `$SelectGroup` (el nombre de la variable editable). No importa qué posición tiene la plantilla definida por el usuario siempre y cuando esté en el ámbito de la variable editable.
- Dentro de esta plantilla definida por el usuario 270 se insertó un cuadro combinado. El cuadro combinado utiliza la expresión XPath `$Groups, 'All'` para seleccionar los valores de entrada de la lista desplegable. Esta expresión XPath devuelve la secuencia de elementos comprendida por la variable `$Groups` (que recopila todos los grupos disponibles dinámicamente) y añade un elemento `All` (*Todos*) a la secuencia que devuelve la variable `$Groups`. La entrada `All` del cuadro combinado servirá para mostrar todos los grupos.
- La plantilla definida por el usuario se incluye dentro de una plantilla condicional basada en el resultado 307 que apunta a la vista Authentic. Esto se debe a que esta variable editable solamente se puede usar en la vista Authentic.
- El paso siguiente es usar el valor de la variable editable seleccionado por el usuario de la vista Authentic. Este valor se usará para filtrar el grupo de equipos que aparece en la vista Authentic. Para ello se definió un filtro 274 en la plantilla `results/group`. La expresión XPath de este filtro es:

  ```
  if ($SelectGroup != 'All') then @id=$SelectGroup else @id
  ```

  Esta expresión crea un paso de predicado en el elemento `group`. Si el valor de la variable `$SelectGroup` no es igual a `All`, el paso de predicado será `[@id=X]`, siendo `X` el valor de la variable `$SelectGroup` (seleccionado por el usuario de la vista Authentic en el cuadro combinado). Este filtro consigue seleccionar el grupo cuyo atributo `id` coincida con el valor seleccionado por el usuario de la vista Authentic. Si el valor de la variable `$SelectGroup` es `All`, entonces la expresión de predicado seleccionará todos los grupos que tengan un atributo `id` (es decir, todos los grupos).
- La plantilla `group` se incluye dentro de una condición basada en el resultado 307, cuyas ramas seleccionan un formato de salida diferente cada una. La única rama donde se aplica un filtro a la plantilla es la rama de la vista Authentic.

Este sería el resultado de la transformación en la vista Authentic:

Observe que al seleccionar la opción B del cuadro combinado la tabla solamente muestra los resultados del grupo B.

 © 2016-2022 Altova GmbH

7.7 Tablas de contenido, referencias y marcadores

La tabla de contenido (TDC) y los demás mecanismos de referencia de StyleVision crean anclajes en los puntos necesarios del documento de diseño y después hacen referencia a estos anclajes desde tablas de contenido, referencias de texto, secuencias de numeración automática e hipervínculos.

Esta sección describe el mecanismo de anclaje (o marcadores) y después el funcionamiento de las tablas de contenido. Para comprender cómo funcionan las tablas de contenido es importante saber cómo funcionan los marcadores.

Funcionamiento de los marcadores

En StyleVision se usan dos tipos de marcadores: simples y complejos. Los marcadores complejos son los que se utilizan para crear tablas de contenido.

- Un marcador simple se puede crear en cualquier posición del documento de diseño. Al marcador se le da un nombre único que se utiliza como destino de los vínculos que apuntan al marcador. Este mecanismo simple se utiliza en marcadores e hipervínculos 364. (Recuerde que los hipervínculos también pueden apuntar a direcciones URL situadas fuera del documento.)
- Para las características de referencia más complejas, como las tablas de contenido o la numeración automática de las secciones de documento, la creación de marcadores tiene dos fases:

  1. Al documento de diseño se le da una estructura jerárquica dividida en niveles. Estos niveles se denominan *niveles TDC*. La estructura se consigue asignando niveles TDC a distintos puntos del documento y anidando niveles TDC dentro de otros niveles. Por ejemplo, puede asignar un nivel TDC a un capítulo de un libro y dentro de este nivel puede crear otro nivel TDC para las secciones del capítulo.
  2. Dentro de los diferentes niveles TDC se crean marcadores TDC. Estos marcadores TDC identifican las secciones de documento de los diferentes niveles que se incluirán en la tabla de contenido. Además cada marcador TDC define el texto que aparecerá en el componente referente.

Una vez definidos todos los niveles TDC y el texto de referencia de los marcadores TDC, puede diseñar la plantilla de la tabla de contenido (plantilla TDC) que contiene los componentes referentes.

A continuación describimos el funcionamiento general de las tablas de contenido. Las diferentes características de referencia se explican en los apartados de esta sección.

Funcionamiento de las tablas de contenido

Si seleccionó la versión **XSLT 2.0** 115 (en lugar de XSLT 1.0) para el diseño SPS, puede crear una tabla de contenido (TDC) en cualquier posición del diseño. Esta tabla es básicamente una plantilla para la tabla de contenido.

- Se recomienda crear marcadores 340 en los elementos del diseño que desea incluir en la TDC. Estos elementos pueden ser de contenido estático o dinámico. En la parte inferior de la imagen siguiente, las etiquetas de marcador TDC de color amarillo 'MyTOC' 'MyTOC' situadas dentro del elemento `header` indican que este elemento incluye un marcador (para poder insertarlo en la plantilla de tabla de contenido `TOC`).
- Es necesario crear una plantilla para la TDC 347 (*resaltada en la imagen siguiente*). La plantilla `TOC` contiene el diseño de la tabla de contenido y se puede colocar en cualquier parte del diseño. En el ejemplo de la imagen siguiente la plantilla `TOC` se colocó al principio del documento.

Nota: no importa el orden en que se creen estos dos componentes. Incluso se pueden crear al mismo tiempo. Sin embargo, recomendamos crear los marcadores TDC antes que la plantilla TDC.

Las tablas de contenido son compatibles con la vista Authentic y con los formatos de salida HTML y RTF. No olvide que (i) puede crear tablas de contenido con una estructura plana o jerárquica (con la correspondiente numeración) y (ii) puede crear varias tablas de contenido en el mismo diseño. Por ejemplo, el diseñador de la hoja de estilos puede crear un documento con una TDC (jerárquica) que incluya los capítulos del libro y otras TDC (también jerárquicas) que incluyan las secciones de cada capítulo, así como listas (planas) de tablas e imágenes.

Procedimiento general para crear tablas de contenido

A continuación aparecen los pasos necesarios para crear una TDC. Primero se crean marcadores en los elementos que se incluirán en la TDC. La plantilla TDC se crea justo después. Otra opción es crear primero la plantilla TDC y después los marcadores. O puede crear la plantilla y seleccionar sus elementos al mismo tiempo.

1. Compruebe que la versión XSLT del diseño es **XSLT 2.0** 115.
2. *Estructure el documento por niveles TDC* 341. Si quiere que la TDC tenga varios niveles, estructure el diseño de forma jerárquica con varios niveles TDC anidados. Si prefiere que su TDC tenga una estructura plana (con un solo nivel), cree un nivel TDC (en el diseño) que incluirá los marcadores TDC.
3. *Cree los marcadores TDC* 344 *dentro de cada nivel del diseño.* Los marcadores TDC identifican los componentes de cada nivel TDC que deben aparecer en la TDC.

4. *Crear una plantilla TDC con referencias a los niveles TDC* 347. La plantilla TDC debe tener el número necesario de referencias a niveles TDC. Si la tabla de contenido tiene varios niveles, las referencias de nivel deberían estar anidadas en la plantilla (*imagen anterior*).
5. *Crear referencias de nivel en la plantilla TDC* 350. En la plantilla TDC cree una referencia TDC por cada nivel. Cada referencia TDC hará referencia por nombre a los marcadores TDC dentro del nivel TDC correspondiente. Otra opción es que la referencia TDC haga referencia a marcadores TDC de otros niveles.
6. *Dar formato a los elementos de la TDC* 351. En los documentos de salida, cada elemento de texto de la TDC es generado por una referencia TDC de la plantilla TDC. Las definiciones de referencia TDC pueden especificar la numeración (también jerárquica), el texto, las líneas de guía y, en medios impresos, el número de página del elemento. En StyleVision puede dar un formato distinto a cada referencia TDC y sus componentes. Recuerde que también puede definir numeración automática dentro de un marcador TDC del cuerpo principal del documento. Para más información consulte el apartado Numeración automática 359.)

Actualizar el número de las páginas en la tabla de contenido de documentos DOCX y RTF

Cuando un usuario edita un documento de salida DOCX o RTF en MS Word que afecte al recuento de páginas, puede que la tabla de contenido no se actualice con las nuevas referencias de página. Se trata de un problema conocido en MS Word. Para actualizar las referencias de página en la tabla de contenido, pulse **Ctrl+A** para seleccionar todo y después pulse **F9**. Consulte esta página para obtener más información.

Terminología

En la tabla que aparece más abajo puede ver todos los términos relacionados con las tablas de contenido. Los componentes de una tabla de contenido pueden aparecer en dos partes del documento de diseño: en el **cuerpo del documento** o en la **plantilla TDC** (la plantilla que especifica el diseño de la tabla de contenido propiamente dicha, que suele aparecer al principio del documento).

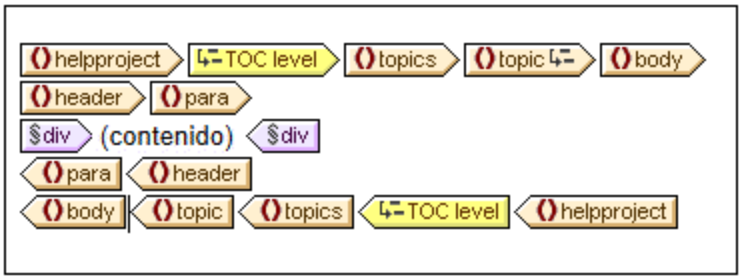
- Los componentes de TDC que aparecen en el ***cuerpo del documento*** marcan qué elemento se usarán en la plantilla TDC.
- Los componentes de TDC que aparecen en la ***plantilla TDC*** hacen referencia a los elementos marcados en el cuerpo del documento. El nombre de los componentes de la plantilla TDC llevan la palabra `referencia`.

CUERPO DEL DOCUMENTO	plantilla TDC
Nivel TDC: los niveles TDC dan estructura al documento, organizándolo en una jerarquía anidada.	**Referencias de nivel:** se corresponden con la estructura de niveles TDC definidas en el cuerpo del documento. Permiten apuntar a los marcadores TDC del nivel correspondiente.
Marcador TDC: tiene un nombre con el que se identifica un nodo del documento como elemento de la TDC.	**Referencias TDC:** hacen referencia a un marcador TDC por su nombre.

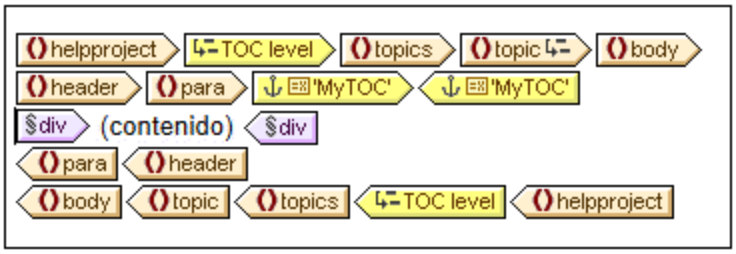
7.7.1 Crear marcadores para incluir elementos en la TDC

Dos pasos son necesarios para crear un marcador en un elemento y así poder incluirlo en la tabla de contenido. El orden de los pasos no tiene importancia.

1. Estructurar el diseño, organizándolo en una jerarquía de niveles TDC anidados 341. Puede crear un nivel TDC en una plantilla o alrededor de un componente de diseño. En la imagen siguiente puede ver que hay un nivel TDC en la plantilla `topic` topic.



Cuando se crea un nivel TDC en una plantilla, esto se indica en la etiqueta inicial de la plantilla con un icono como este: topic. Cuando el nivel TDC se crea alrededor de un componente, esto se indica con las etiquetas TOC level TOC level. En la imagen anterior, la plantilla `topics` está incluida en un nivel TDC. La diferencia entre crear el nivel TDC en una plantilla o alrededor de ella se explica detalladamente en el apartado Estructurar el diseño por niveles TDC 341. Cuando cree la plantilla TDC 347, deberá organizarla jerárquicamente por niveles y cada nivel debe corresponder a un nivel TDC creado en el cuerpo del documento. Incluso si la TDC tiene una estructura plana (un solo nivel), en el diseño debe haber el nivel TDC correspondiente.
2. Crear un marcador TDC 344 en el diseño con un nombre y texto para el elemento de la TDC. El marcador TDC puede incluir o no un componente de diseño. Es decir, puede estar vacío o no. En la imagen siguiente, por ejemplo, el marcador TDC no incluye ningún componente de diseño.

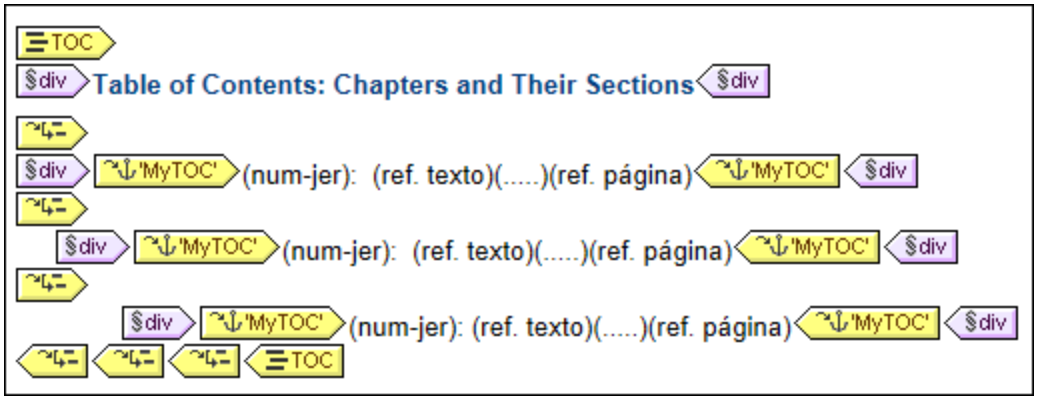


El marcador TDC sirve de anclaje. En la imagen anterior, por ejemplo, el marcador TDC (con icono en forma de ancla) está al principio de las instancias de los elementos `para`. El marcador TDC tiene dos atributos: (i) un nombre que se usará para hacerle referencia cuando se cree el elemento en la plantilla TDC y (ii) una cadena de texto que se usará como texto del elemento correspondiente de la TDC. Para ver cómo asignar estos atributos a los marcadores TDC consulte el apartado Crear marcadores TDC 344.

 © 2016-2022 Altova GmbH

Cómo hacer referencia en la plantilla TDC a los elementos que tienen marcadores

La plantilla TDC 347 está dividida en niveles anidados llamados *referencias de nivel* (y no *niveles TDC*, que son los que se crean en el cuerpo principal del diseño). Dentro de cada referencia de nivel , se inserta una referencia TDC 'MyTOC' (*imagen siguiente*). La referencia TDC que está dentro de una referencia de nivel remite a los marcadores TDC usando el nombre del marcador. Cada marcador TDC que tenga ese nombre en ese nivel del documento XML se insertará como elemento de la TDC en ese nivel. Por ejemplo, en la imagen siguiente, la referencia TDC 'chapters' remite a todos los marcadores TDC llamados `chapters` del nivel correspondiente del documento XML (si el ámbito de la referencia TDC es `nivel actual`). El atributo de texto de los respectivos marcadores TDC se usará para generar el texto del elemento de la TDC.



En la plantilla TDC de la imagen anterior hay tres referencias de nivel anidadas y dentro de cada una de ellas hay una referencia TDC que contiene la plantilla para el elemento de TDC de dicho nivel. Por ejemplo, en la primera referencia de nivel, hay una referencia TDC que apunta a los marcadores TDC llamados `MyTOC` 'MyTOC'. Como resultado todos los marcadores TDC del primer nivel (del diseño) llamados `MyTOC` se usarán para generar el contenido de este nivel de la TDC. La referencia TDC del segundo nivel también remite a los marcadores llamados `MyTOC`. Como resultado todos los marcadores del segundo nivel (del diseño) llamados `MyTOC` se usarán para los elementos de segundo nivel de la TDC. La referencia de tercer nivel funciona igual: los marcadores TDC llamados `MyTOC` que aparezcan en el tercer nivel del diseño se usan para crear los elementos de tercer nivel de la TDC.

En los siguientes subapartados describimos (i) cómo estructurar el diseño por niveles 341 y (ii) cómo crear marcadores 344. Para aprender a crear una plantilla de tabla de contenido consulte la sección Crear la plantilla TDC 347.

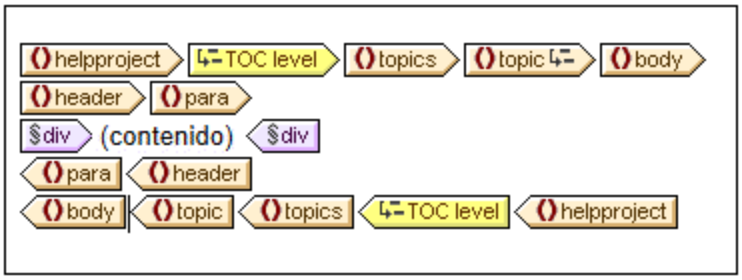
7.7.1.1 Estructurar el diseño por niveles TDC

La estructura jerárquica que define el diseño de la tabla de contenido se especifica por medio de **niveles anidados**. Se trata de una estructura jerárquica que, a pesar de estar relacionada con la estructura del documento XML, es independiente de esta. Esta estructura se especifica en el diseño SPS. La plantilla TDC usará una estructura equivalente a esta estructura jerárquica. En el caso de las tablas de contenido de estructura plana (con un solo nivel), el diseño debe tener como mínimo un nivel. Si en el documento hay más de un nivel, puede crear una tabla de contenido plana para cualquiera de ellos o para varios (agregados como un solo nivel).

Los niveles se pueden crear en la plantilla principal, en plantillas globales o en ambas. Lo importante es que todos los niveles juntos describan una estructura jerárquica clara.

Crear niveles

Los niveles del diseño se crean uno por uno, bien en una plantilla, bien alrededor de un componente. En la imagen siguiente, por ejemplo, se creó un nivel en la plantilla `topic` (observe el icono de la etiqueta inicial topic) y otro alrededor del elemento `topics` (tal y como indican las etiquetas TOC level TOC level).



Siga estos pasos para crear un nivel:

1. Seleccione el componente (una plantilla o cualquier otro componente).
2. Haga clic con el botón derecho y, en el menú contextual, seleccione el comando **La plantilla funciona como nivel** (si seleccionó una plantilla) o **Incluir en | Nivel TDC**. Ambos comandos también están disponibles en el menú **Insertar | Insertar tabla de contenido**: **Nivel TDC** o **La plantilla funciona como nivel**.

Niveles en las plantillas globales

También puede crear niveles en las plantillas globales. En este caso debe tener cuidado y comprobar que los niveles creados en diferentes plantillas globales, así como los de la plantilla principal, definen **juntos** una estructura jerárquica cuando se ejecute el diseño SPS. En la imagen siguiente, por ejemplo, hay dos niveles: uno en la plantilla principal (en la plantilla `topic`) y uno en la plantilla global para `topic` (en la plantilla `topic`).

 © 2016-2022 Altova GmbH

En el modelo de contenido que representa la imagen anterior, `topic` es un elemento recursivo, es decir, un elemento `topic` que puede contener un elemento descendiente `topic`. En la plantilla principal (cuyo final se señala con la etiqueta $XML) se creó un nivel en el primer nivel de `topic` topic. La instrucción `(resto de contenido)` de la plantilla principal especifica que se aplicarán plantillas a todos los elementos secundarios de `topic/body` excepto a `header`. Esto significa que se procesará la plantilla global para los secundarios `topic` de `topic/body`.

En la plantilla global de `topic` se creó un nivel en la plantilla `topic` (indicado por la etiqueta topic). Este segundo nivel de la jerarquía de la TDC, que aparece en el segundo nivel de elementos `topic`, está anidado dentro del primer nivel de la jerarquía. Como esta plantilla global también tiene una instrucción (`resto de contenido`), la plantilla global para `topic` se aplicará a todos los elementos `topic` recursivos, lo cual creará niveles anidados adicionales en la jerarquía: un tercer nivel, un cuarto nivel y así sucesivamente.

El diseñador del SPS debería saber cuántos niveles se crearon en el diseño porque cuando construya la plantilla TDC deberá especificar explícitamente cómo se seleccionan los elementos de la TDC para cada nivel y qué formato se les da.

Niveles en plantillas de TDC

En una tabla de contenido plana, los elementos de la TDC se crean todos al mismo nivel (p. ej. una lista de imágenes del documento).

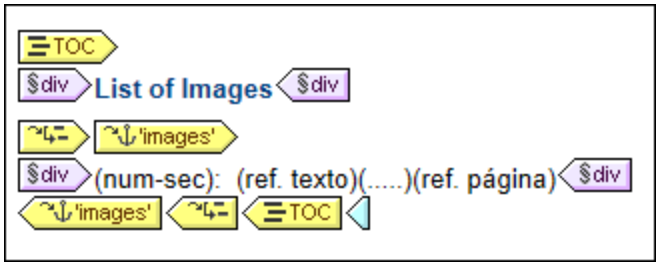
Hay varias formas de obtener una jerarquía plana:

- Puede estructurar el documento de diseño con un solo nivel TDC. La plantilla TDC tendrá una sola referencia de nivel con una sola referencia TDC.
- Si el documento de diseño tiene más de un nivel TDC, entonces la plantilla TDC puede tener un

número de referencias de nivel igual a la posición secuencial del nivel TDC al que se hace referencia. La referencia de nivel correspondiente al nivel TDC al que se apunta contendrá la única referencia TDC de la tabla de contenido.
- Si el documento de diseño tiene más de un nivel TDC, la única referencia TDC de la tabla de contenido debe tener un ámbito que abarque todos los niveles de documento a los que se apunta.

Imagine que quiere reunir todas las imágenes del documento en una tabla de contenido plana. Para ello el diseño debe tener al menos un nivel y este nivel debe contener todos los marcadores TDC necesarios. En la plantilla de TDC a las imágenes se les debe hacer referencia tal y como hemos explicado: (i) creando el número correspondiente de referencias de nivel y (ii) creando una referencia TDC dentro de la referencia de nivel que corresponde al nivel TDC al que se apunta. La referencia TDC tendrá el nombre de los marcadores TDC del nivel TDC al que se apunta.

La plantilla TDC que aparece a continuación tiene una referencia de nivel con una referencia TDC que remite a los marcadores TDC llamados `images`. El ámbito de la referencia TDC es el `nivel actual e inferiores`. Como resultado se hará referencia a todos los marcadores TDC llamados `images` del primer nivel y de los niveles inferiores (es decir, de todo el documento).



Si el diseño contiene más de un nivel y quiere crear una tabla de contenido plana para los elementos de segundo nivel, por ejemplo, la plantilla TDC podría tener dos referencias de nivel, con una referencia TDC dentro del segundo nivel (y ninguna referencia en el primero). También puede usar la propiedad `ámbito` de las referencias TDC para especificar en qué niveles del diseño se deben buscar los marcadores con un nombre determinado.

7.7.1.2 Crear marcadores TDC

Los marcadores TDC se deben crear dentro de un nivel TDC 341 del diseño. Se pueden crear en la plantilla principal y en plantillas globales. Un marcador TDC cumple dos funciones:

- Marca un componente (estático o dinámico) del diseño con un nombre estático definido por el usuario. Puede incluir un componente de diseño o estar vacío. En los documentos de salida, se genera una instancia del marcador TDC como anclaje con nombre. A este anclaje con nombre se le puede hacer referencia desde los elementos de la tabla de contenido.
- Define la cadena de texto que será el texto del elemento de la tabla de contenido. Esta cadena de texto puede ser el contenido de elementos secundarios del nodo donde está el marcador o el resultado de una expresión XPath.

Hay dos formas de crear marcadores TDC:

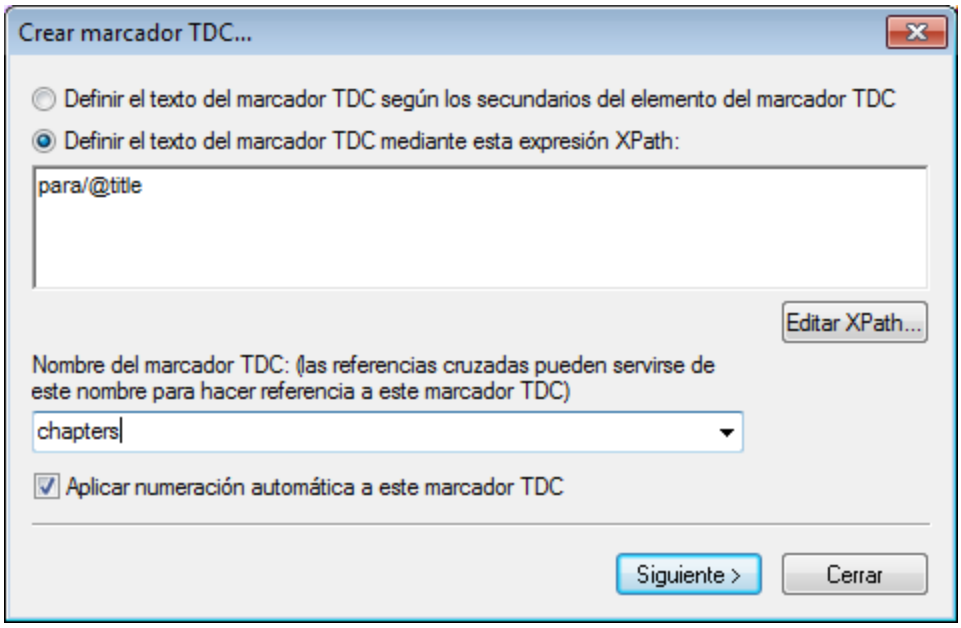
 © 2016-2022 Altova GmbH

- Con el asistente para la creación de marcadores TDC 345, que permite especificar el nombre del marcador TDC, su texto, si se numera automáticamente y el nivel donde debe aparecer.
- Insertando un marcador TDC vacío 346, cuyas propiedades se definen más tarde.

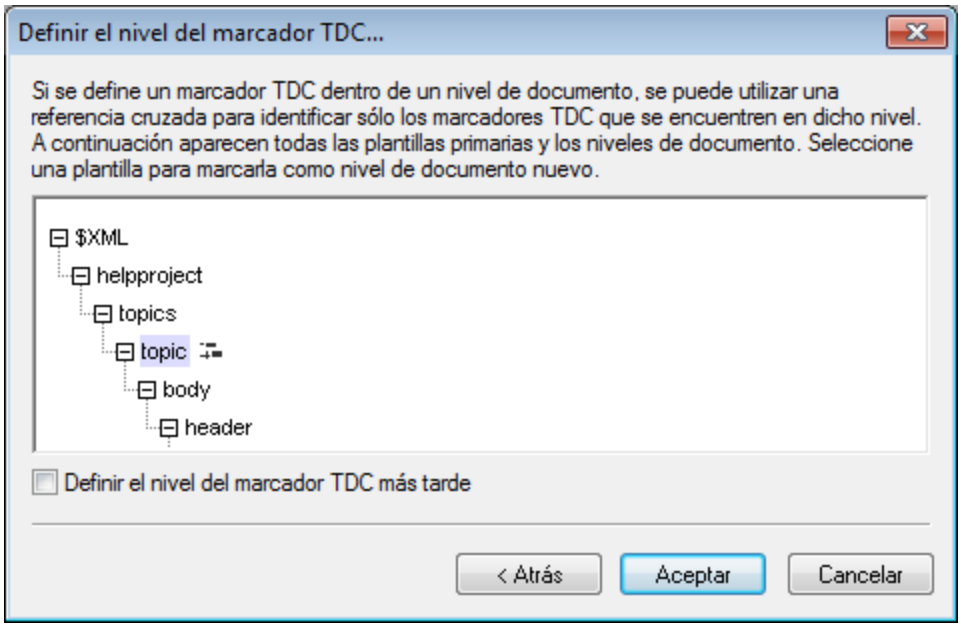
Crear el marcador TDC con el asistente

Si quiere usar el asistente para crear un marcador TDC:

1. Ponga el cursor en el punto del diseño donde desea insertar el marcador TDC. Si lo prefiere, puede seleccionar el componente de diseño que desea insertar dentro del marcador TDC.
2. Haga clic con el botón derecho y seleccione **Insertar tabla de contenido | Marcador TDC (Asistente)**. Este comando también está en el menú **Insertar**. Si quiere incluir un nodo dentro de un marcador TDC, entonces seleccione el comando **Incluir en | Marcador TDC (Asistente)**. A continuación aparece el cuadro de diálogo "Crear marcador TDC" (*imagen siguiente*).



3. En la primera pantalla del asistente (*imagen anterior*) puede definir: (i) el texto del elemento de la TDC, (ii) el nombre del marcador TDC y (iii) si el marcador debe aparecer en los documentos de salida. Para definir el texto del marcador tiene dos opciones: usar el texto de los secundarios del nodo o una expresión XPath. Para definir el nombre del marcador también tiene dos opciones: escribir el nombre en el campo de entrada o seleccionar un nombre de la lista desplegable (que contiene los nombres de otros marcadores TDC). Cuando termine haga clic en **Siguiente**.
4. En la segunda pantalla del asistente (*imagen siguiente*) puede crear un nivel TDC en una plantilla. En esta pantalla aparece un árbol con todas las plantillas antecesoras del punto de inserción. Si ya se creó alguna plantilla como nivel TDC, se indica con un icono especial. En la imagen, por ejemplo, el icono de la plantilla `topic` indica que funciona como nivel. Si quiere crear un nivel más en alguna plantilla antecesora, seleccione la plantilla en este árbol. También puede definir los niveles más tarde (haga clic en la casilla *Definir el nivel del marcador TDC más tarde*). Cuando termine haga clic en **Finalizar**. Tenga en cuenta que si ya existe un nivel TDC en una plantilla y selecciona dicha plantilla, al hacer clic en **Finalizar** no se creará un nivel TDC nuevo.



Al hacer clic en **Finalizar** se crea un marcador TDC en el punto de inserción y, si así se indicó en la segunda pantalla del asistente, se crea un nivel TDC en una plantilla. El marcador TDC recién creado estará en el nivel TDC. Por ejemplo, si el nivel TDC es el tercer nivel TDC de la jerarquía, entonces el marcador TDC insertado estará en el tercer nivel TDC.

Crear un marcador TDC

Para crear un marcador TDC sin atributos (sin nombre, sin texto, etc.):

1. Ponga el cursor en el punto de inserción o seleccione el componente de diseño que desea incluir dentro del marcador TDC.
2. Haga clic con el botón derecho y seleccione **Insertar tabla de contenido | Marcador TDC** (este comando también está disponible en el menú **Insertar**). Se inserta el marcador TDC, que no tiene nombre ni texto. Para definir estos atributos del marcador use los comandos de edición 347 (*ver más abajo*).

Insertar numeración jerárquica o secuencial para un componente

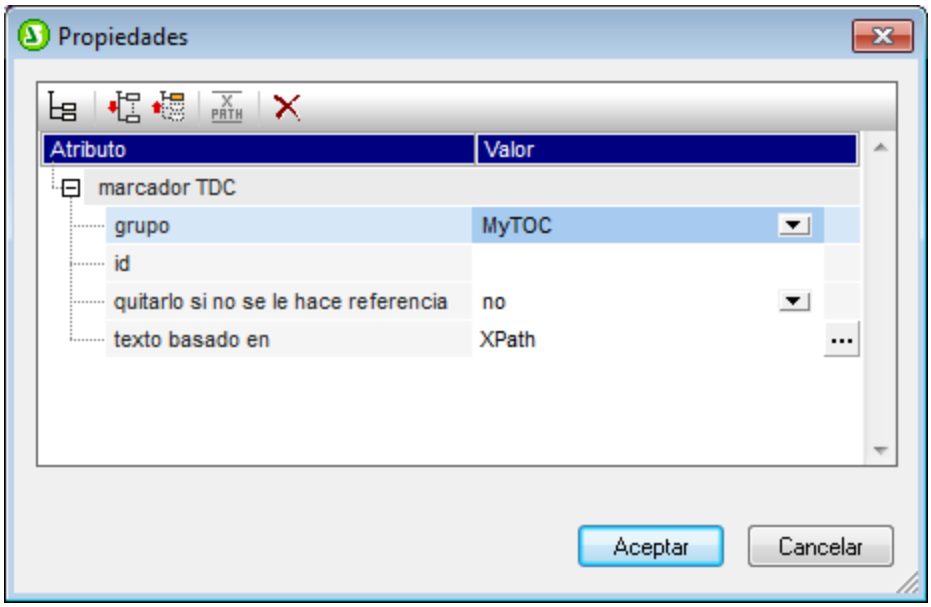
La numeración jerárquica o secuencial del cuerpo principal del documento de salida (no de la tabla de contenido) se puede insertar dentro (o fuera) de las etiquetas del marcador TDC. Haga clic con el botón derecho donde desea insertar la numeración, seleccione **Insertar tabla de contenido | Numeración jerárquica / Numeración secuencial**. Por ejemplo, si colocamos un marcador de numeración automática alrededor de la plantilla del título del capítulo, se numerarán todos los títulos de capítulo generados por esta plantilla.

Tenga en cuenta que la numeración está basada en la estructura de los niveles TDC. Por ejemplo, si un elemento de título de capítulo está en el primer nivel TDC, el cuarto título de capítulo tendrá el número `4` porque es la cuarta instancia de título de capítulo dentro del primer nivel TDC. Si las secciones de un capítulo aparecen dentro del segundo nivel TDC, entonces la tercera sección del cuarto capítulo tendrá el número `4.3`. Esto se debe a que, dentro del primer nivel TDC (capítulo), es la cuarta instancia de un capítulo y, dentro del segundo nivel TDC (sección) del cuarto capítulo, es la tercera instancia de una sección.

 © 2016-2022 Altova GmbH

Editar el nombre y el texto de un marcador TDC

El nombre y el texto del marcador TDC se puede editar en la ventana Propiedades (*imagen siguiente*). Para editar estas propiedades, seleccione el marcador TDC y después edite la propiedad en la ventana Propiedades 59. O haga clic con el botón derecho en el marcador y seleccione la propiedad que desea editar en el cuadro de diálogo que aparece.



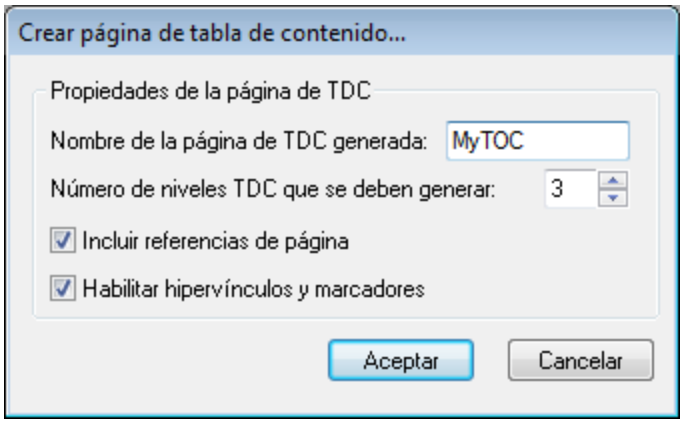
Las propiedades de un marcador TDC son: (i) el nombre del grupo de marcadores TDC (`grupo`); (ii) un identificador único (`id`); (iii) una opción para quitar el marcador si no se le hace referencia y (iv) una opción para especificar el texto, que puede venir del contenido del marcador o de una expresión XPath (`texto basado en`).

7.7.2 Crear la plantilla TDC

La plantilla TDC es la plantilla que genera la tabla de contenido en los documentos de salida. Se puede crear en cualquier posición del diseño SPS e incluso puede crear varias plantillas TDC dentro del mismo SPS.

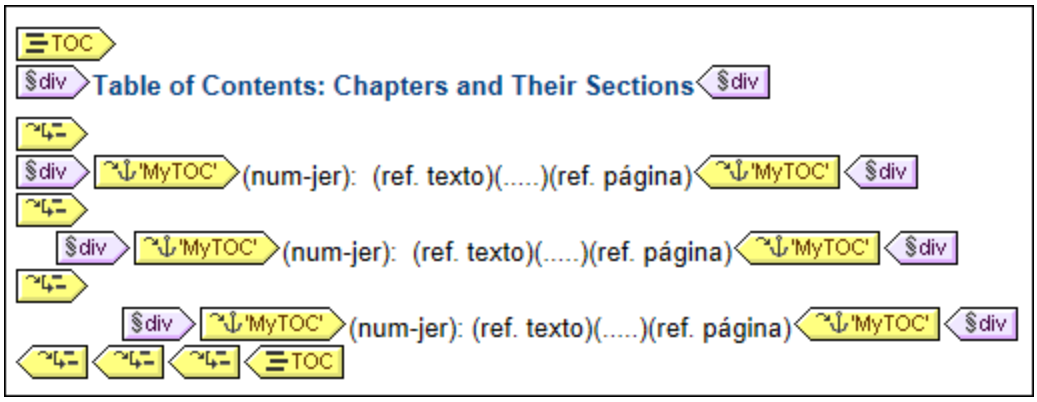
Siga estas instrucciones para crear una plantilla TDC:

1. Ponga el cursor en la posición donde desea insertar la plantilla TDC.
2. Haga clic en el comando de menú **Insertar | Insertar tabla de contenido | Tabla de contenido**. Este comando también está disponible en el menú contextual que parece si hace clic con el botón derecho en el punto de inserción. Al hacer clic en este comando se abre el cuadro de diálogo "Crear página de tabla de contenido" (*imagen siguiente*).



3. Rellene la información del cuadro de diálogo:
   - el nombre de la página de TDC generada es el nombre (referencia TDC) que se usará para apuntar a los marcadores TDC 344 del documento de diseño. Si selecciona varios niveles para la tabla de contenido (segunda opción) se usará el mismo nombre de referencia TDC en todas las referencias de nivel (aunque los nombres de referencia TDC se pueden editar más tarde 350).
   - el número de referencias de nivel TDC 341 indica cuántas referencia de nivel debe tener la TDC.
   - para los medios impresos también puede indicar si se generan referencias de página (es decir, si se incluye el número de la página).
   - las entradas de texto de la TDC se pueden usar como vínculos a los marcadores TDC.

4. Cuando termine haga clic en **Aceptar**. La plantilla TDC se crea con el número de referencias de nivel indicados.



En la imagen anterior puede ver que dentro de cada referencia de nivel hay una referencia TDC con un nombre que identifica los marcadores TDC que deben ser los elementos de dicha referencia de nivel. Dentro de cada referencia TDC hay una plantilla predeterminada para el elemento de la TDC. Esta plantilla se puede editar en todo momento 351.

Editar la plantilla TDC

Estas son las opciones de edición disponibles:

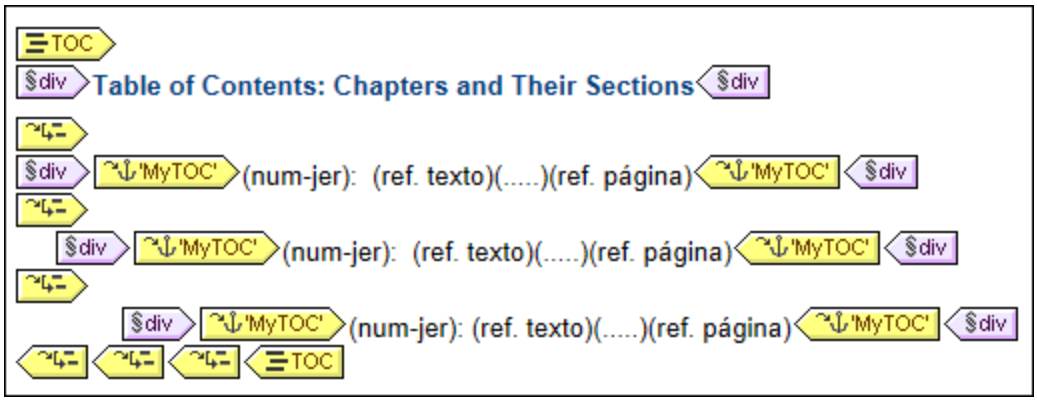
- La plantilla TDC se puede arrastrar a otra posición del diseño SPS. Sin embargo, recuerde que al arrastrarla puede cambiar el nodo de contexto, lo cual afectará a las expresiones XPath de la plantilla TDC.

 © 2016-2022 Altova GmbH

- Puede añadir o eliminar referencias de nivel 349 en la estructura de la plantilla TDC.
- Puede editar las propiedades de cada referencia TDC 350. El nombre y el ámbito de una referencia TDC se pueden cambiar y también puede especificar si el elemento de la TDC que corresponde a la referencia se crea como hipervínculo o no.
- Puede añadir o eliminar referencias TDC 350 de las referencias de nivel de la plantilla TDC.
- Puede dar formato al elemento de tabla de contenido 351 de una referencia TDC con ayuda de estilos CSS 390.
- Las características básicas del diseño SPS (como imágenes, cálculos automáticos y componentes de formato por bloques) se pueden insertar en cualquier parte de la plantilla TDC.

7.7.2.1 Referencias de nivel en la plantilla TDC

La plantilla TDC 347 está estructurada mediante referencias de nivel (*imagen siguiente*). Estos niveles se crean cuando se crea la plantilla TDC y el número de referencias de nivel es el número indicado en el cuadro de diálogo Crear página de tabla de contenido 347.



Observe que las referencias de nivel están anidadas. Existe una equivalencia entre las referencias de nivel de la plantilla TDC y los niveles del diseño SPS. Por consiguiente, la primera referencia de nivel de la plantilla TDC equivale al primer nivel del diseño SPS, la segunda equivale al segundo nivel del diseño SPS y así sucesivamente. Las referencias TDC de cada referencia de nivel identifican los marcadores TDC 344 del ámbito indicado 350 del diseño SPS. Por ejemplo, una referencia TDC puede apuntar a los marcadores TDC del respectivo nivel de documento o a los marcadores TDC de todos los niveles del documento o a los del nivel actual y de los niveles inferiores.

Insertar y eliminar referencias de nivel

Una vez creada la plantilla TDC puede insertar o eliminar referencias de nivel. Hay dos maneras de hacerlo.

Puede seleccionar el contenido de la plantilla donde desea crear la referencia de nivel y seleccionar el comando de menú **Incluir en | Referencia de nivel TDC** (este comando también está en el menú contextual que aparece al hacer clic con el botón derecho en el contenido seleccionado). La otra opción es insertar una referencia de nivel vacía en el punto de inserción (clic en **Insertar | Insertar tabla de contenido | Referencia de nivel TDC**. También disponible en el menú contextual).

Para eliminar una referencia de nivel de la plantilla TDC, seleccione la referencia y pulse la tecla **Suprimir** o el comando **Eliminar** del menú contextual. Tenga en cuenta que solamente se eliminará la referencia de nivel y no su contenido.

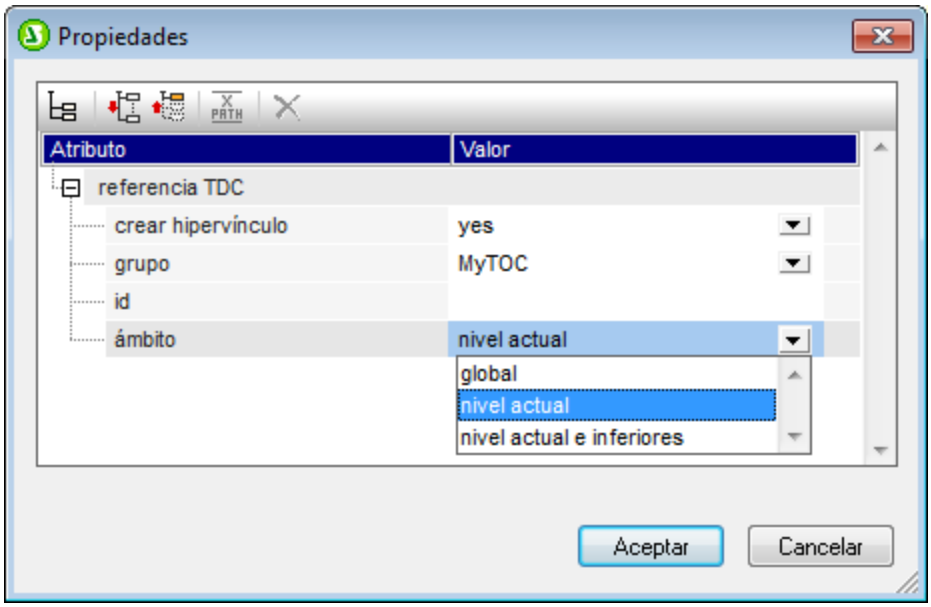
7.7.2.2 Referencias TDC: nombre, ámbito, hipervínculo

Las referencias TDC aparecen dentro de las referencias de nivel de una plantilla TDC y tienen cuatro propiedades (*imagen siguiente*):

- **`crear hipervínculo`**: esta propiedad puede activarse/desactivarse con `yes/no` respectivamente para indicar si los respectivos elementos TDC se crean como hipervínculos o no.
- **`grupo`**: esta propiedad indica el nombre de la referencia TDC e identifica a los marcadores TDC del mismo nombre que están dentro del ámbito especificado. Los marcadores TDC identificados aportan los elementos que se deben incluir en dicha referencia de nivel de la tabla de contenido.
- **`id`**: esta propiedad es el identificador único que identifica a la referencia TDC.
- **`ámbito`**: esta propiedad especifica a qué niveles del diseño SPS se aplica la referencia TDC. Las opciones disponibles son: (i) **global**, (ii) **nivel actual**, (iii) **nivel actual e inferiores** (*imagen siguiente*).

Para insertar una referencia TDC ponga el cursor dentro de una referencia de nivel y seleccione el comando de menú (o del menú contextual) **Insertar | Insertar tabla de contenido | Referencia TDC**.

Para editar las propiedades de una referencia TDC, haga clic con el botón derecho en su etiqueta y seleccione la propiedad que desea editar (**Crear hipervínculo**, **Editar Id.**, **Editar grupo**, **Editar ámbito**). Esto abre la ventana Propiedades con la propiedad correspondiente seleccionada (*imagen siguiente*).



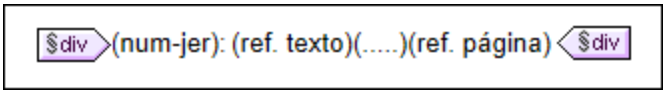
También puede editar las propiedades de la referencia TDC en la ventana Propiedades (grupo de propiedades *referencia TDC*).

© 2016-2022 Altova GmbH

7.7.2.3 Aplicar formato a los elementos de la TDC

Cada elemento de la tabla de contenido está formado por hasta cuatro componentes más el contenido opcional especificado por el usuario. Los cuatro componentes estándar son (*imagen siguiente*):

- el **texto del elemento**, que en la plantilla TDC se indica con el marcador de posición `(ref. texto)`
- la **línea de guía** entre el texto y el número de página (solo en formatos de salida para medios impresos), que en la plantilla TDC se indica con el marcador de posición `(.....)`
- la **referencia de página** (solo en formatos de salida para medios impresos), que en la plantilla TDC se indica con el marcador de posición `(ref. página)`
- la **numeración jerárquica o secuencial**, que en la plantilla TDC se indica con el marcador de posición `(num-lvl)` o `(num-seq)` respectivamente



Nada más crear la plantilla TDC, la entrada de texto se inserta automáticamente dentro de referencias TDC. Si marcó la opción *Incluir referencias de página* cuando creó la plantilla, la línea de guía y la referencia de página se incluye también. Estos componentes puede editarse y eliminarse según corresponda. Para insertar un componente haga clic con el botón derecho en el punto de inserción del elemento TDC y seleccione **Insertar tabla de contenido | Referencia TDC | Entrada de texto** o **Línea de guía** o **Referencia de página** o seleccione **Insertar tabla de contenido | Numeración jerárquica** o **Numeración secuencial** según corresponda. Utilice numeración jerárquica cuando el diseño esté estructurado por niveles anidados y numeración secuencial cuando solamente haya un nivel TDC, es decir, cuando la jerarquía sea plana (ver más abajo). Para eliminar un componente selecciónelo y pulse la tecla **Suprimir**.

Además puede insertar contenido estático (p. ej. texto) y dinámico (p. ej. cálculos automáticos) en el elemento de la tabla de contenido.

Niveles de las tablas de contenido planas

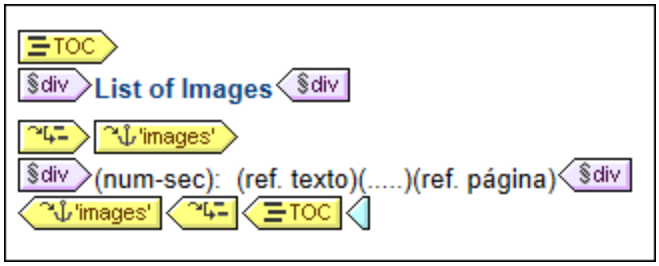
En una tabla de contenido plana, los elementos de la TDC se crean todos al mismo nivel (p. ej. una lista de imágenes del documento).

Hay varias formas de obtener una jerarquía plana:

- Puede estructurar el documento de diseño con un solo nivel TDC. La plantilla TDC tendrá una sola referencia de nivel con una sola referencia TDC.
- Si el documento de diseño tiene más de un nivel TDC, entonces la plantilla TDC puede tener un número de referencias de nivel igual a la posición secuencial del nivel TDC al que se hace referencia. La referencia de nivel correspondiente al nivel TDC al que se apunta contendrá la única referencia TDC de la tabla de contenido.
- Si el documento de diseño tiene más de un nivel TDC, la única referencia TDC de la tabla de contenido debe tener un ámbito que abarque todos los niveles de documento a los que se apunta.

Imagine que quiere reunir todas las imágenes del documento en una tabla de contenido plana. Para ello el diseño debe tener al menos un nivel y este nivel debe contener todos los marcadores TDC necesarios. En la plantilla de TDC a las imágenes se les debe hacer referencia tal y como hemos explicado: (i) creando el número correspondiente de referencias de nivel y (ii) creando una referencia TDC dentro de la referencia de nivel que corresponde al nivel TDC al que se apunta. La referencia TDC tendrá el nombre de los marcadores TDC del nivel TDC al que se apunta.

La plantilla TDC que aparece a continuación tiene una referencia de nivel con una referencia TDC que remite a los marcadores TDC llamados `images`. El ámbito de la referencia TDC es el `nivel actual e inferiores`. Como resultado se hará referencia a todos los marcadores TDC llamados `images` del primer nivel y de los niveles inferiores (es decir, de todo el documento).



Si el diseño contiene más de un nivel y quiere crear una tabla de contenido plana para los elementos de segundo nivel, por ejemplo, la plantilla TDC podría tener dos referencias de nivel, con una referencia TDC dentro del segundo nivel (y ninguna referencia en el primero). También puede usar la propiedad `ámbito` de las referencias TDC para especificar en qué niveles del diseño se deben buscar los marcadores con un nombre determinado.

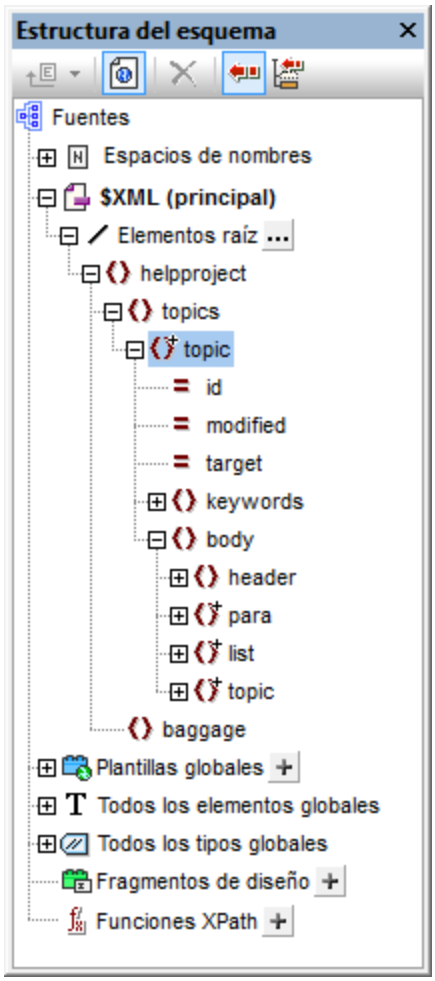
Dar formato a los elementos de la tabla de contenido

Puede aplicar formato a los elementos de la TDC con ayuda de estilos CSS 390 desde la ventana Estilos 398. A cada componente del elemento se le puede dar un formato distinto. Para ello seleccione el componente en la vista **Diseño** y después defina sus propiedades de estilo 398 en la ventana Estilos.

7.7.3 Ejemplo: tabla de contenido básica

El diseño SPS `ChaptersSimple.sps` está en la carpeta (Mis) Documentos 34, `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\TOC`, y muestra el uso básico de las tablas de contenido. Este diseño está basado en un esquema que define el modelo de contenido de un documento de gran tamaño dividido en capítulos. La estructura del esquema aparece en la imagen siguiente y se puede ver en la ventana Estructura del esquema de StyleVision al abrir el archivo `ChaptersSimple.sps`. (Para pasar directamente a un ejemplo de tabla de contenido más complejo consulte el apartado Ejemplo: TDC jerárquica y secuencial 356.)

 © 2016-2022 Altova GmbH



El elemento de documento del esquema es `helpproject`, que contiene un elemento secundario llamado `topics`. El elemento `topics` puede contener un número ilimitado de elementos `topic` y cada uno de estos puede tener elementos secundarios llamados `topic`. El primer nivel de elementos `topic` serían los capítulos del documento, mientras que sus elementos `topic` descendientes serían las secciones del documento, las subsecciones y así sucesivamente.

Este SPS crea una tabla de contenido al principio del documento de salida que enumera los nombres de los capítulos (los `topics` de primer nivel). Para crear esta tabla de contenido son necesarios estros tres pasos:

1. Estructurar el diseño por niveles TDC 341: en el documento de diseño se insertan **niveles** para dar una estructura jerárquica a los documentos de salida. Esta estructura jerárquica será la de la tabla de contenido. En el ejemplo que nos ocupa se creó un solo nivel TDC en la plantilla `Topic`. Como solo hay un nivel en el diseño, la plantilla TDC solamente puede tener un nivel (es decir, una referencia de nivel).
2. Crear marcadores TDC 344: dentro del nivel TDC creado en el primer paso se crea un **marcador TDC**. Esto permite a las referencias TDC de la plantilla TDC (que se creará en el próximo paso) apuntar a este marcador. El marcador TDC también especifica qué texto aparecerá en el elemento de la TDC al que apunta.
3. Crear la plantilla TDC 347: se trata de la plantilla que genera la tabla de contenido en el documento. Se divide en referencias de nivel, que deben corresponderse con la estructura de niveles TDC del documento de diseño. Por ejemplo, si hay tres **referencias de nivel** anidadas en la plantilla TDC,

entonces el diseño debe tener como mínimo tres niveles anidados. En nuestro ejemplo tenemos una sola referencia de nivel que equivale al nivel TDC del diseño. Dentro de la referencia de nivel se coloca la **referencia TDC**, que se ocupa de generar los elementos TDC de este nivel de la tabla de contenido.

Estructura y niveles del diseño SPS

Observe la estructura del diseño SPS y verá que la plantilla principal (delimitada por las etiquetas `$XML` verdes) contiene la tabla de contenido. El resto de la plantilla principal especifica (mediante la instrucción `(resto de contenidos)`) que debe aplicarse la plantilla global y las plantillas predeterminadas. El resto del diseño SPS (es decir, todo lo que esté fuera de la plantilla principal) son plantillas globales.

Las definiciones de tabla de contenido (niveles TDC y marcadores TDC del diseño) están en la plantilla global de `topic` (*imagen siguiente*). En esta plantilla global se insertó una condición para separar los elementos `topic` según su número de elementos `topic` antecesores, lo cual permite procesar diferentemente los capítulos, las secciones y las subsecciones (con ayuda de ramas condicionales).

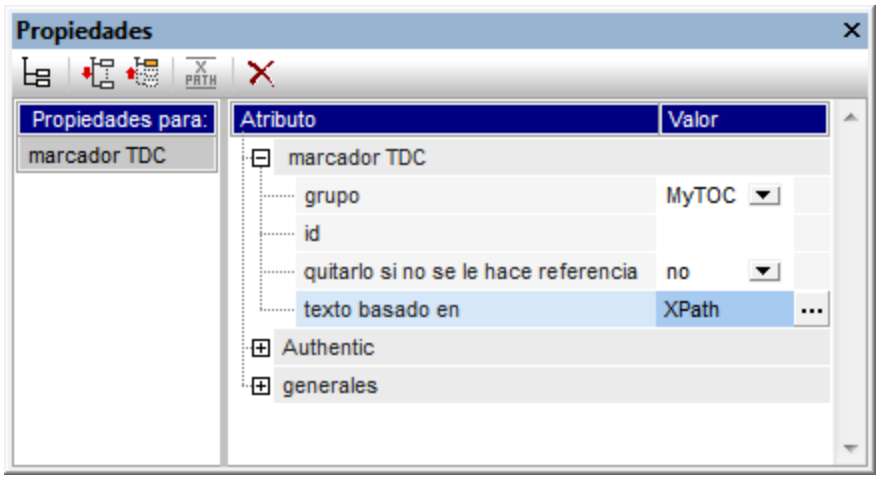
En la imagen anterior puede ver el contenido de la primera rama condicional, para los elementos de primer nivel y de tipo capítulo `topic`. Observe que se creó un nivel TDC en la etiqueta inicial de plantilla de este elemento `topic`. En las otras dos ramas condicionales no se creó ningún nivel TDC para la plantilla de `topic`. Por tanto, al documento se le asignó un solo nivel TDC y este se encuentra en el elemento de primer nivel `topic` (de tipo capítulo).

Crear los marcadores TDC

Después se creó un marcador TDC (*etiquetas amarillas de la imagen siguiente*) dentro del elemento `header` descendiente de `topic` (pero fuera del elemento `para`). Este marcador TDC sirve de anclaje para cada elemento `topic` de nivel superior (de tipo capítulo).

Las propiedades del marcador TDC se pueden editar en la ventana Propiedades (*imagen siguiente*).

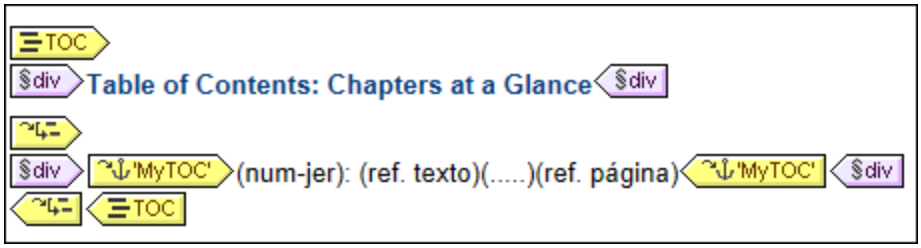
 © 2016-2022 Altova GmbH



La propiedad `grupo` especifica el grupo de marcadores TDC (y es el nombre del marcador TDC). En nuestro ejemplo hemos usado el valor `MyTOC` para esta propiedad. Cuando se cree la TDC se hará referencia a este grupo de marcadores y se pueden especificar varios grupos en el mismo nivel. La propiedad `id` asigna un identificador único a las instancias creadas del marcador. La propiedad `quitarlo si no se le hace referencia` es una opción para quitar el marcador si no se le hace referencia. La propiedad `texto basado en` especifica la entrada de texto que se usará como texto del elemento TDC de la tabla de contenido. El texto puede estar basado en el contenido del marcador (el contenido situado entre las etiquetas del marcador en el diseño) o en una expresión XPath. En nuestro ejemplo usamos una expresión XPath que devuelve el texto del título de cada elemento `topic` de primer nivel.

La plantilla TDC

Dentro de la plantilla TDC (*imagen siguiente*), se insertó una sola referencia de nivel . Esta referencia de nivel equivale al nivel TDC asignado al elemento de primer nivel `topic`.

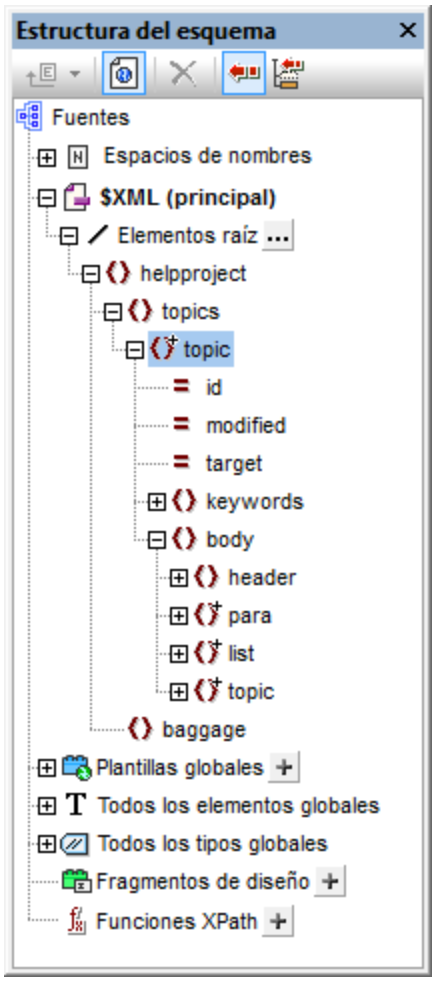


Dentro de esta referencia de nivel se insertó una referencia TDC . Esta referencia TDC se configuró para seleccionar los marcadores que (i) están en el grupo de marcadores llamado `MyTOC` y que (ii) están dentro del ámbito del nivel actual solamente. Esta configuración se puede definir en la ventana Propiedades cuando se selecciona la referencia TDC.

El aspecto del elemento TDC se especifica dentro de las etiquetas de la referencia TDC de la plantilla TDC. El formato de numeración, el texto, la línea de guía y la referencia de página se pueden insertar haciendo clic con el botón derecho dentro de las etiquetas de la referencia TDC y seleccionando el componente en el menú contextual. Estos componentes se pueden editar y modificar en la ventana Propiedades.

7.7.4 Ejemplo: TDC jerárquica y secuencial

En la carpeta (Mis) Documentos[34], `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\TOC`, encontrará un archivo SPS de ejemplo llamado `Chapters.sps` que ilustra el uso de las tablas de contenido. Este SPS está basado en un esquema que define el modelo de contenido de un documento de gran tamaño dividido en capítulos. La estructura del documento aparece en la ventana Estructura del esquema (*imagen siguiente*) nada más abrir el archivo `Chapters.sps`.



El elemento de documento del esquema es `helpproject`, que contiene un elemento secundario `topics`. El elemento `topics` puede incluir un número ilimitado de elementos `topic` y cada uno de ellos puede incluir a su vez elementos descendientes `topic`. El primer nivel de elementos `topic` son los capítulos del documento, mientras que sus elementos descendientes `topic` son las secciones, subsecciones y así sucesivamente.

El diseño SPS contiene tres tablas de contenido, situadas al principio del documento en este orden:

1. *Chapters at a glance*[357]*:* esta TDC enumera los nombres de los capítulos (los elementos `topic` de primer nivel).

2. *Chapters and their sections*[358]*:* esta TDC enumera todos los capítulos con sus secciones descendientes (los `topic` de primer nivel, más la jerarquía de `topics` de cada `topic` hasta llegar a los elementos `topics` de tercer nivel).
3. *List of images*[359]*:* esta TDC es una lista plana con todas las imágenes del documento (excepto la primera) en orden alfabético según el nombre del archivo.

Estructura del diseño SPS

Antes de centrarnos en las tablas de contenido observemos la estructura del diseño. Observe que la plantilla principal (delimitada por las etiquetas `$XML` verdes) contiene las TDC. El resto de la plantilla principal especifica que se apliquen las plantillas globales y predeterminadas (mediante la instrucción `(resto de contenidos)`).

Las definiciones de tabla de contenido están en las plantilla globales de `topic` y de `image`. En la plantilla global de `topic` (*imagen siguiente*) se creó un nivel TDC en el elemento `topic` y un marcador TDC dentro del elemento secundario `header` (pero fuera del elemento `para`).

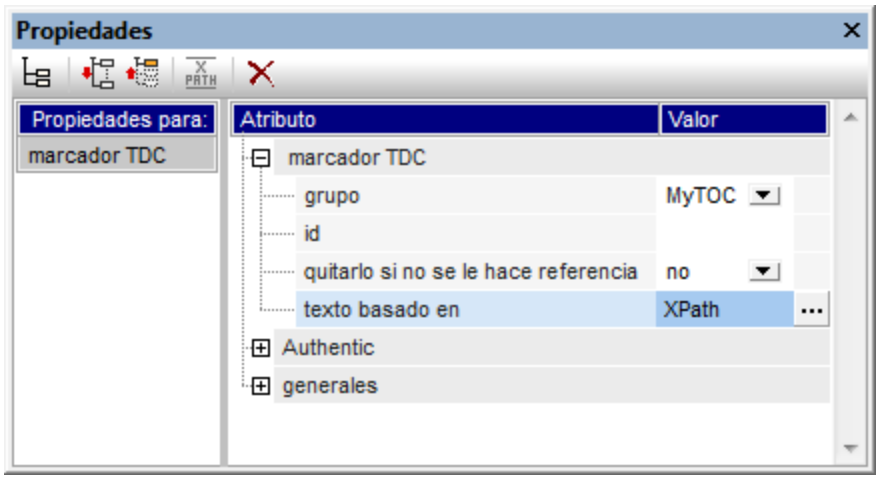
Como el elemento `topic` es recursivo, el nivel TDC y el marcador TDC también irán hacia atrás. Es decir, que en la primera recursión, se crea un nuevo nivel TDC y un nuevo marcador TDC subordinados. Este proceso se repite en cada `topic` descendiente, lo cual da lugar a una jerarquía de niveles TDC descendientes y cada uno de ellos tiene un marcador TDC. Como el formato del título de cada nivel TDC debe ser distinto, se incluyó cada nivel dentro de una rama distinta de una condición con tres ramas. Cada rama mira en qué nivel TDC aparece un `topic`: el primero, el segundo o el tercero.

Observe que se insertó numeración jerárquica (`num-lvl`) dentro del nivel. Para insertarla haga clic con el botón derecho en la posición deseada y seleccione **Insertar tabla de contenido | Numeración jerárquica**. Con esto se consigue insertar el número jerárquico adecuado antes del título de cada capítulo/sección (como `3.1` o `4.2.3`. por ejemplo).

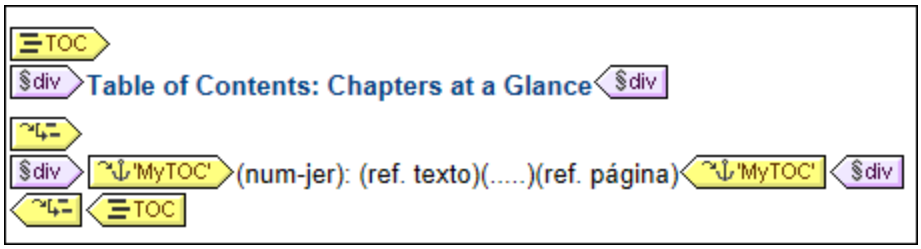
Descripción de las tablas de contenido

A continuación describimos cada tabla de contenido.

Chapters at a glance: seleccione el marcador TDC de la plantilla global para `topic`. En la ventana Propiedades (*imagen siguiente*) observe que el texto de entrada se construye con una expresión XPath. Al hacer clic en el botón de edición del valor de la propiedad `texto basado en`, verá que la expresión XPath definida es `para`. Esto significa que el contenido del elemento `para` secundario de `header` se usará como texto del elemento de la tabla de contenido (porque el marcador TDC se insertó dentro del elemento `header`).

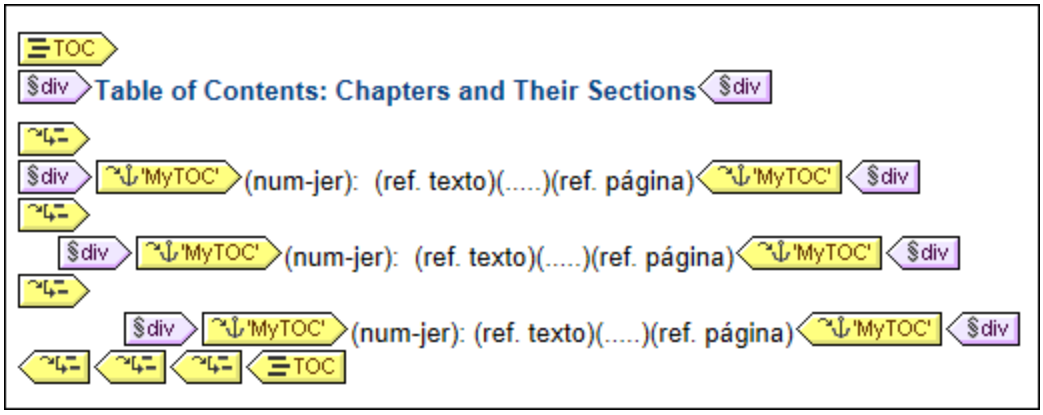


La plantilla TDC propiamente dicha (*imagen siguiente*) contiene una referencia de nivel y la referencia TDC de esa referencia de nivel 'MyTOC' selecciona los marcadores TDC llamados `MyTOC` que están dentro del ámbito del nivel actual solamente (es decir, el primer nivel). Como resultado se crean elementos de TDC solamente para los `topic` de primer nivel.



Observe que se definió la numeración de tipo jerárquica.

Chapters and their sections: en esta tabla de contenido (*imagen siguiente*) hay tres referencias de nivel anidadas y cada una de ellas contiene una referencia TDC cuyo ámbito es el nivel actual.

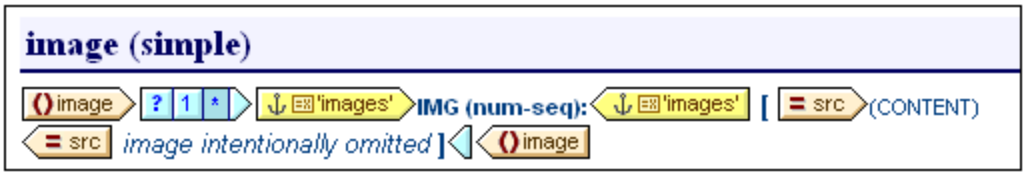


Como cada elemento de la TDC está dentro de un bloque `div`, pueden darse propiedades de formato distintas para cada bloque.

© 2016-2022 Altova GmbH

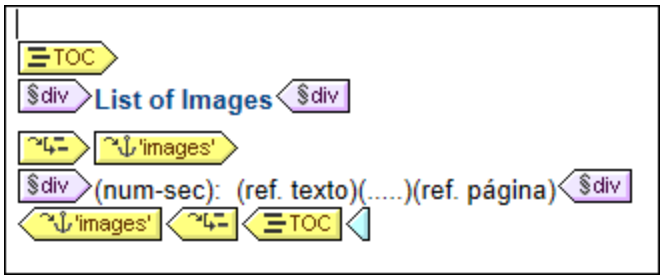
List of images: la lista de imágenes es una lista plana. Primero, debe tener en cuenta en qué niveles aparecen las imágenes en el documento de instancia. El elemento `image` es un secundario del elemento `para`. Como los niveles se crearon en los elementos `topic`, los elementos `image` aparecerán en el primer, segundo y tercer nivel del documento. Por tanto no es necesario crear ningún nivel para el elemento `image`.

La condición de la plantilla global de `image` (*imagen siguiente*) permite procesar diferentemente (i) la primera imagen y (ii) las demás imágenes.



Observe que el marcador TDC se colocó dentro de la segunda rama condicional. Esto significa que las imágenes seleccionadas en la primera rama no tienen marcadores. Además la numeración secuencial (`num-sec`) de las imágenes (insertada con **Insertar tabla de contenido | Numeración secuencial**) empezará con la segunda imagen (porque la primera imagen se selecciona en la primera rama condicional). También debe prestar atención al hecho de que la numeración tiene asignado un formato. Para ver el formato haga clic con el botón derecho en `(num-sec)` y seleccione el comando **Editar formato**. En el cuadro de diálogo que aparece verá que el formato establecido es `01`. Esto indica que se insertará un `0` delante de los números de una sola cifra.

En la plantilla TDC de `images` (*imagen siguiente*) verá que hay una referencia TDC que identifica los marcadores llamados `images` y que esta referencia TDC está dentro de una sola referencia de nivel. El ámbito de la referencia TDC (que se puede editar en la ventana Propiedades) se definió como: `nivel actual e inferiores`. El nivel actual, que viene dado por la referencia de nivel, es el primer nivel. Los niveles inferiores son el segundo y tercer nivel y así sucesivamente. De este modo se seleccionan todas las imágenes a partir del primer nivel.



Como la numeración elegida es secuencial, las imágenes se enumeran consecutivamente en la lista plana.

7.7.5 Numeración automática en el cuerpo del diseño

Las instancias repetidas de un nodo se pueden numerar automáticamente en el cuerpo principal del documento usando la característica de numeración automática. Por ejemplo, en un elemento `Libro` que contiene varios elementos `Capítulo`, cada elemento `Capítulo` se puede numerar automáticamente. Esto permite insertar fácilmente una numeración basada en la estructura del documento XML.

Nota: la característica de numeración automática se refiere a la numeración de componentes del cuerpo principal del documento. **No se refiere** a la numeración dentro de las tablas de contenido, donde la numeración es una propiedad específica del elemento de la TDC.

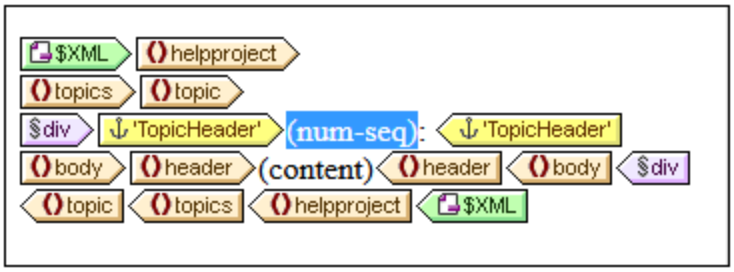
La numeración automática puede ser secuencial (plana) o jerárquica. La numeración secuencial se establece en un solo nivel. La numeración jerárquica está basada en la jerarquía de niveles TDC creada en el documento y según la posición del elemento dentro de la jerarquía de niveles TDC.

Si se trata de una numeración jerárquica con números, puede dar un formato distinto a cada componente del número. Por ejemplo, un número formado por tres componentes podría tener este formato: `A.1.i.`. El formato de los números no se asigna igual en la numeración jerárquica y en la numeración secuencial (ver más abajo).

Numeración secuencial `(num-seq)`

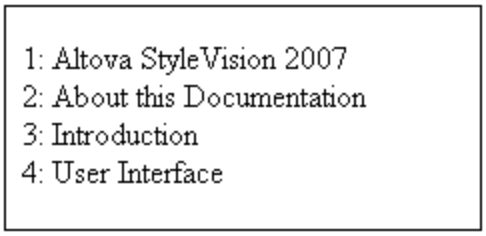
Puede insertar numeración secuencial dentro de un marcador TDC 344 del documento de diseño (*imagen siguiente*). Siga estos pasos para hacerlo:

1. Ponga el cursor dentro del nodo que desea numerar y cree el marcador TDC (clic con el botón derecho y clic en **Insertar tabla de contenido | Marcador TDC**). El marcador TDC se crea en el nodo. Como queremos numerar el elemento `topic`, el marcador TDC se creó dentro del elemento `topic`. La posición exacta dentro del elemento `topic` depende de en qué posición desea usar la numeración. (En la imagen siguiente, la numeración se colocó justo a la izquierda del título del capítulo).
2. Haga clic con el botón derecho dentro de las etiquetas del marcador TDC y seleccione **Insertar tabla de contenido | Numeración secuencial**. Se inserta el marcador de posición de numeración secuencial `(num-seq)` (*imagen siguiente*).



3. Si tiene pensado hacer referencia al marcador TDC desde una plantilla TDC, puede editar las propiedades del marcador TDC. Pero si el marcador TDC solo se va a utilizar para la numeración automática, no hace falta darle un nombre ni editar sus propiedades. Si quiere darle un nombre, haga clic con el botón derecho en el marcador y seleccione **Editar grupo**.

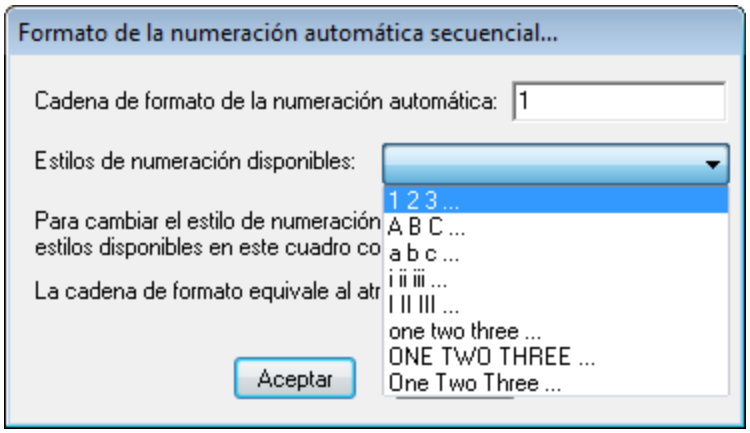
En el ejemplo de la imagen anterior se estableció numeración secuencial para el nodo `topic`. El resultado es que cada elemento `topic` recibe un número (*imagen siguiente*). El número es la posición de cada elemento `topic` dentro de la secuencia de elementos `topic` de ese nivel de la jerarquía XML del documento.



 © 2016-2022 Altova GmbH

Nota: si desea que la numeración secuencial continúe en otro conjunto de nodos, utilice un marcador TDC con el mismo nombre en ambos conjuntos de nodos.

Para dar formato a la numeración secuencial haga clic con el botón derecho en el marcador de posición `(num-seq)` y seleccione **Editar formato**. Se abre el cuadro de diálogo "Formato de la numeración automática secuencial..." (*imagen siguiente*).

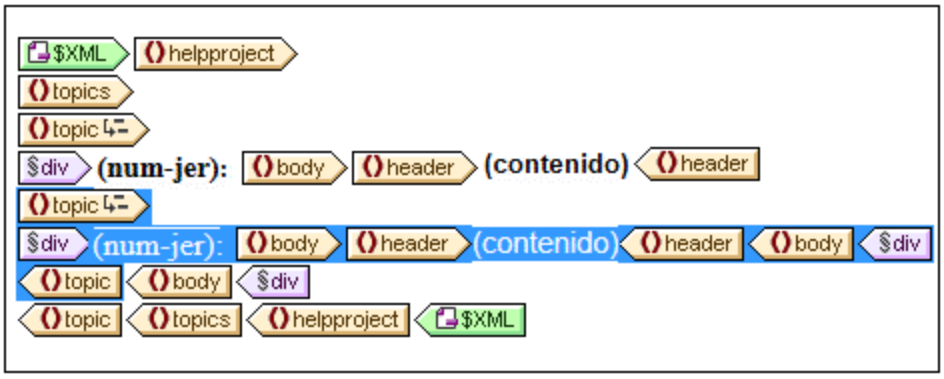


Seleccione el formato en la lista desplegable del cuadro combinado *Estilos de numeración disponibles* (*imagen siguiente*) y haga clic en **Aceptar** para aplicar el formato seleccionado.

Numeración jerárquico `(num-lvl)`

La numeración jerárquica se puede insertar dentro de un nivel TDC del diseño 341. Por tanto, para crear numeración jerárquica en un documento, primero debe crear la estructura de niveles TDC (siguiendo las instrucciones del apartado Estructurar el diseño por niveles TDC) 341. Debe tener en cuenta estos puntos:

- Los niveles deben crearse en el nodo que quiere numerar o dentro del nodo.
- Los niveles deben anidarse en función de la jerarquía de la numeración.
- El marcador de posición de la numeración jerárquica debe insertarse dentro del nivel correspondiente del diseño (*imagen siguiente*).



En el ejemplo de la imagen anterior hay dos niveles. El elemento `topic` es recursivo y se creó un nivel en dos elementos `topic` (clic con el botón derecho en la etiqueta del nodo y después clic en **La plantilla funciona como nivel**). Un elemento `topic` (resaltado en la imagen) se anida dentro de otro. Como resultado los niveles

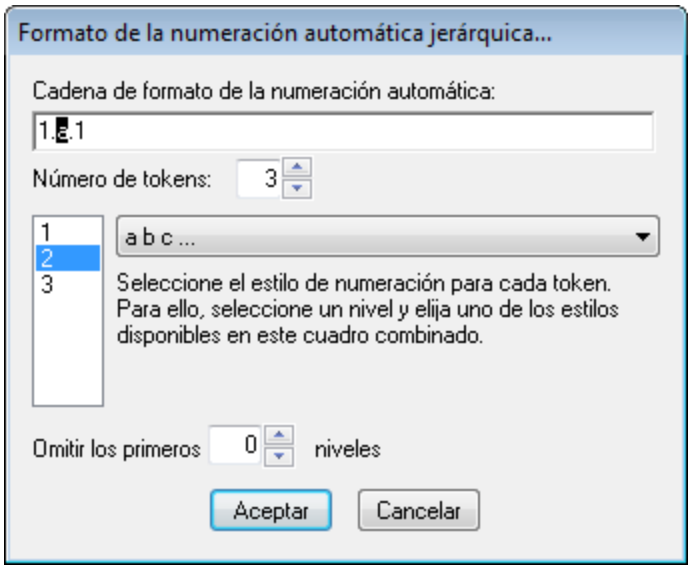
también están anidados. Dentro de cada nivel se insertó un marcador de posición `(num-lvl)` (clic con el botón derecho dentro del nivel y después clic en **Insertar tabla de contenido | Numeración jerárquica**).

El resultado del diseño aparece en esta imagen.

1: Altova StyleVision 2007
2: About this Documentation
3: Introduction
3.1: What Is an SPS?
3.2: Product Features
3.3: Setting up StyleVision
4: User Interface
4.1: Main Window
4.2: Design Entry Helpers

El primer nivel aparece en negrita y el segundo nivel normal.

Para dar formato a la numeración jerárquica, haga clic con el botón derecho en el marcador de posición `(num-lvl)` y seleccione el comando **Editar formato**. Se abre el cuadro de diálogo "Formato de la numeración automática jerárquica" (*imagen siguiente*).



Primero seleccione un número en el cuadro combinado *Número de tokens.* Este número debe ser igual al número de niveles TDC del documento. Cada token puede tener un formato diferente. En el cuadro vertical seleccione el token al que desea dar formato (por ejemplo, en la imagen anterior está seleccionado el segundo token). Después seleccione el estilo de formato en el cuadro combinado situado a su derecha. En la imagen anterior, por ejemplo, se seleccionaron letras minúsculas para el segundo token (el resultado aparece en el primer cuadro). Además puede puede omitir niveles con la opción *Omitir los primeros X niveles*.

No olvide que debe dar formato a cada numeración jerárquica de cada nivel.

 © 2016-2022 Altova GmbH

Cuando termine haga clic en **Aceptar**.

7.7.6 Referencias cruzadas

Una referencia cruzada es una referencia a otra parte del documento. Las referencias cruzadas se crean en dos pasos en el diseño SPS: primero se establece el destino de la referencia cruzada y después se define el vínculo al destino.

Para establecer el destino es necesario crear un marcador TDC dentro de un nivel TDC. El vínculo al destino es una referencia de texto dentro de la referencia TDC. La referencia de texto genera el texto de salida y funciona como vínculo.

La creación de referencias cruzadas tiene tres pasos:

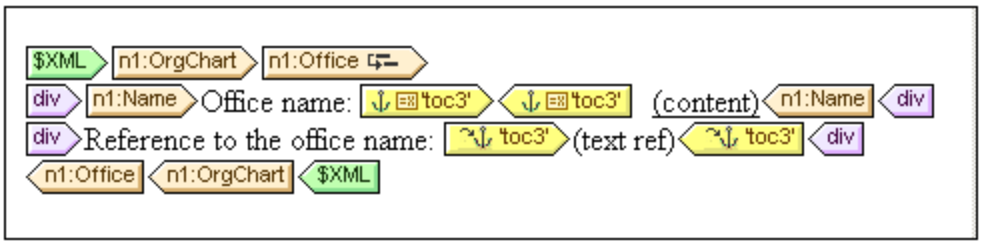
Paso 1: niveles

El documento se estructura por niveles TDC tal y como se describe en el apartado Estructurar el diseño por niveles TDC 341. Los niveles TDC se usarán para especificar el ámbito de la referencia. Solamente se apuntará a los marcadores TDC que tengan determinado nombre y que estén en determinado ámbito. En la imagen siguiente, se creó un nivel en el elemento `n1:Office`.

Paso 2: crear marcadores TDC

Para crear el marcador TDC dentro de un nivel haga clic con el botón derecho en la posición deseada y seleccione **Insertar tabla de contenido | Marcador TDC**. Al marcador se le da un nombre y una expresión XPath que genera el texto de salida. La expresión XPath suele identificar un nodo del documento, cuyo contenido se usa como texto de salida.

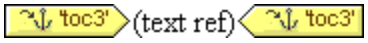
En la imagen siguiente, el marcador TDC del elemento `n1:Name` 'toc3' 'toc3' se llama `toc3` y tiene una expresión XPath que encuentra el nodo actual. Esto significa que el texto de salida será el contenido del nodo `n1:Name`.



Cuando se procesa el documento XML, se crea un anclaje por cada elemento `n1:Name`. Este anclaje tendrá una referencia de texto (el texto de la referencia cruzada) que es el valor del elemento `n1:Name`.

Paso 3: crear referencias TDC

La referencia TDC se inserta (menú contextual, **Insertar tabla de contenido | Referencia TDC**) para crear un vínculo a los anclajes generados por un marcador TDC.



En la imagen anterior la referencia TDC `toc3` está dentro del mismo nivel TDC que el marcador TDC al que hace referencia (el nivel `Office`). También debe especificar el ámbito de la referencia TDC. El ámbito indica en qué niveles TDC se deben buscar los marcadores TDC que se llamen igual que la referencia TDC. En el ejemplo anterior el ámbito es el nivel actual. Esto significa que esta referencia apunta a los marcadores TDC del nivel actual que se llaman `toc3`.

La imagen anterior también muestra la plantilla `n1:Office`. Cuando se procesa un nodo `n1:Office`, se crea un anclaje con el texto de salida que es el contenido del nodo `n1:Name`. Esto se debe a que el marcador TDC especifica por medio de una expresión XPath (propiedad `texto basado en` del marcador TDC) que el contenido de este nodo será el texto de salida. La referencia TDC de la siguiente línea identifica el anclaje llamado `toc3` y el componente `ref. texto` genera el texto de salida del vínculo al anclaje (texto de color violeta). El resultado sería algo así:

Office name: Nanonull, Inc.
Reference to the office name: Nanonull, Inc.
Office name: Nanonull Europe, AG
Reference to the office name: Nanonull Europe, AG

En el ejemplo anterior el ámbito es el nivel actual. Se pueden usar otros dos ámbitos: (i) un ámbito global o (ii) el nivel actual y los niveles inferiores. Gracias a estas opciones podrá apuntar a marcadores de otros niveles del diseño.

7.7.7 Marcadores e hipervínculos

Puede insertar marcadores en cualquier parte del diseño SPS. Estos marcadores se transforman en anclajes en los documentos de salida, a los cuales se puede apuntar desde hipervínculos. Los hipervínculos pueden enlazar con marcadores y con recursos externos como páginas web, por ejemplo. StyleVision ofrece gran flexibilidad a la hora de generar URI de destino para los hipervínculos.

Esta sección describe:

- Cómo insertar marcadores 364 en el diseño SPS.
- Cómo insertar hipervínculos 366 en el diseño SPS y cómo enlazan con las páginas de destino.

Nota: la especificación FO es compatible con el uso de vínculos a documentos externos, pero puede que su procesador FO no lo sea. Si tiene pensado usar vínculos a documentos externos en sus documentos PDF de salida, consulte primero si su procesador FO es compatible con esta característica.

7.7.7.1 Insertar marcadores

Los marcadores (anclajes) se pueden insertar en cualquier posición del diseño SPS, bien en el punto de inserción del cursor o bien alrededor de un componente SPS.

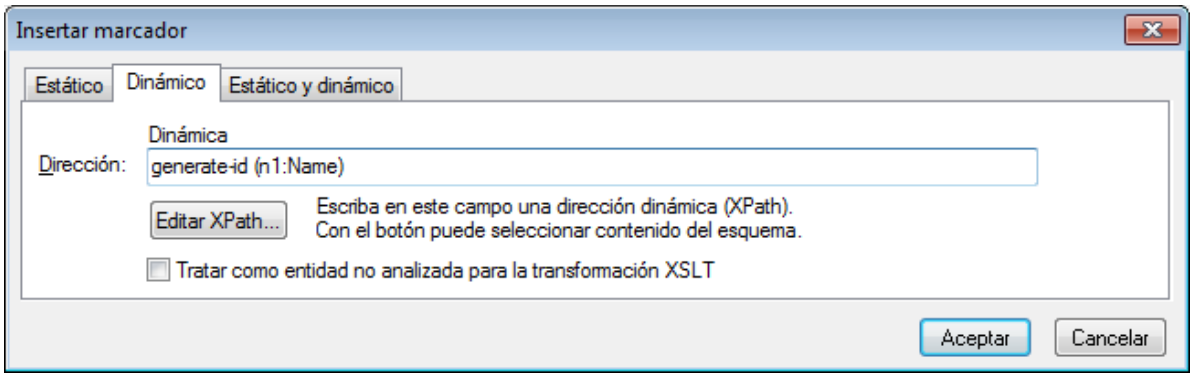
 © 2016-2022 Altova GmbH

En el SPS los marcadores se crean en el cuadro de diálogo "Insertar marcador" (*imagen siguiente*). En este cuadro de diálogo puede definir el nombre del marcador, que puede ser estático o dinámico: (i) dinámico derivado del contenido del documento XML o (ii) generado aleatoriamente con una expresión XPath.

Crear un marcador

Siga estos pasos para insertar un marcador:

1. Ponga el cursor en la posición donde desea crear el marcador.
2. Seleccione el comando de menú **Insertar | Insertar marcador** 825 o haga clic con el botón derecho y seleccione **Insertar | Marcador**.
3. Se abre el cuadro de diálogo "Insertar" (*imagen siguiente*), seleccione una pestaña dependiendo del tipo de nombre que desea darle al marcador (estático, dinámico o estático y dinámico). En el ejemplo de la imagen siguiente se creó un marcador dinámico, cuyo nombre es un Id. único por cada elemento `Name` secundario del nodo de contexto.



4. Haga clic en **Aceptar** para terminar de definir el marcador.

Una vez creado el marcador puede enlazar con un hipervínculo 366.

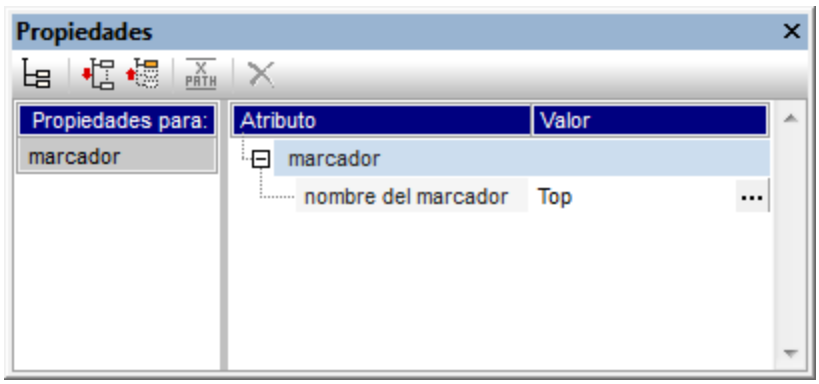
Nota: los marcadores se crean en la posición indicada del diseño. Si la posición elegida está dentro de un elemento que se repite, entonces se crea un marcador dentro de cada instancia del elemento. Si se le da un nombre estático, cada marcador tendrá el mismo nombre. Por tanto, en el caso de marcadores situados en elementos que se repiten, es mejor darles un nombre dinámico (p. ej. el nombre de un elemento secundario del nodo de contexto). Si el nodo seleccionado para el nombre dinámico tiene el mismo contenido en varias instancias, puede utilizar la función `generate-id()` para generar un nombre único para el marcador. Para hacer referencia a este tipo de marcadores, se puede generar un Id. igual al valor `href` de un hipervínculo 369. En este caso asegúrese de usar el identificador de fragmentos `#` delante de la función `generate-id()`. La expresión XPath sería: `concat('#', generate-id(nodoXXX))`.

Modificar un marcador

Una vez creado el marcador, su nombre se puede modificar en el cuadro de diálogo "Editar marcador". Para abrir este cuadro de diálogo:

1. Seleccione el marcador en el diseño.

2. En la ventana Propiedades haga clic en el botón de edición de la propiedad `nombre del marcador` (*imagen siguiente*). En ese momento se abre el cuadro de diálogo "Editar marcador" que es idéntico al cuadro de diálogo "Insertar marcador" (*imagen anterior*).



3. Ahora edite el nombre del marcador en una de las tres pestañas: *Estático*, *Dinámico* o *Estático y dinámico*.

Eliminar un marcador

Para eliminar un marcador selecciónelo en el diseño y pulse la tecla **Suprimir**.

7.7.7.2 Definir hipervínculos

Los hipervínculos se pueden crear alrededor de componentes SPS como texto o imágenes. El destino de los hipervínculos pueden ser (i) marcadores del diseño SPS o (ii) recursos externos como páginas web o mensajes de correo electrónico. Esta sección describe el contenido de los hipervínculos y su destino.

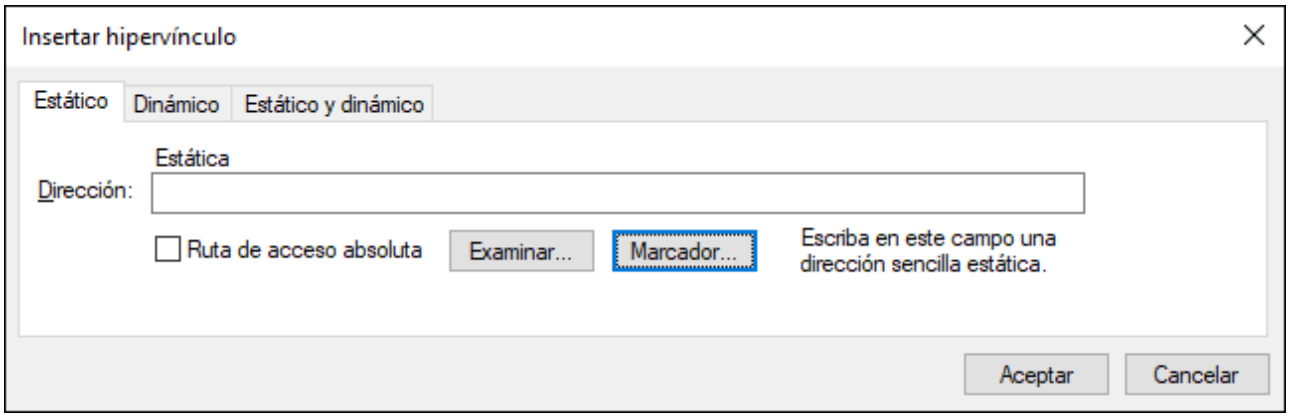
Crear hipervínculos

Los hipervínculos se pueden crear de varias formas:

- Alrededor de texto (estático o dinámico), nodos, imágenes, plantillas condicionales, cálculos automáticos y bloques de contenido o nodos. No se puede crear alrededor de dispositivos de entrada de datos como campos de entrada y cuadros combinados (aunque se pueden crear alrededor de un nodo o plantilla condicional que incluya un dispositivo de entrada de datos). Al hacer clic en este contenido del hipervínculo se abre el destino. Para crear un hipervínculo alrededor de un componente, seleccione el componente y use el comando de menú **Incluir en | Hipervínculo**.
- Puede insertar un hipervínculo vacío con el comando de menú **Insertar | Hipervínculo**. El contenido deberá introducirse más tarde entre las etiquetas del hipervínculo recién creado.

Definir el destino del hipervínculo

El destino del hipervínculo se define en el cuadro de diálogo "Insertar hipervínculo" (*imagen siguiente*). Para abrir este cuadro de diálogo haga clic en **Incluir en | Hipervínculo** 840 o en **Insertar | Hipervínculo** 826.



El destino de un vínculo puede ser:

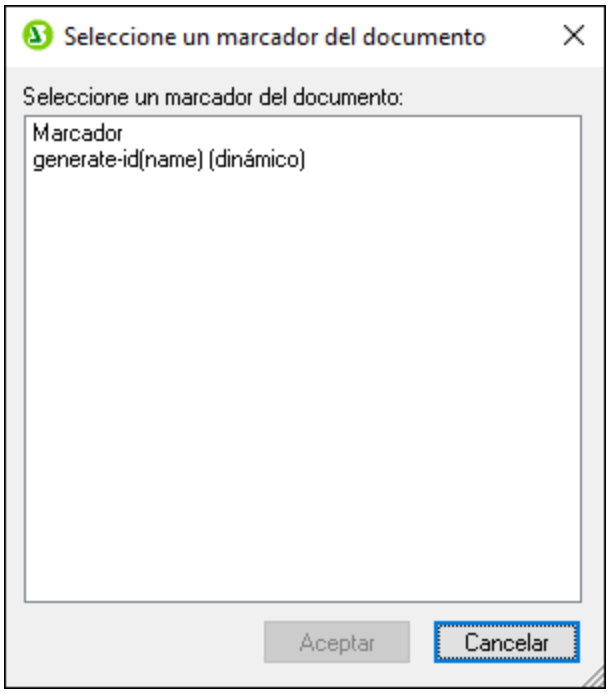
- Un marcador 367 del mismo diseño SPS (en cuyo caso el URI de destino debe ser un identificador de fragmento)
- Un destino generado dinámicamente 369 que se corresponda con anclajes de marcadores (estos URI también son identificadores de fragmento),
- Un recurso externo 369. El URI puede ser estático, dinámico (tomado de un nodo del documento XML), estático y dinámico o el valor de una entidad sin analizar.

A continuación explicamos cómo definir estos destinos de un hipervínculo. Una vez definido el URI en el cuadro de diálogo "Insertar/Editar hipervínculo" haga clic en **Aceptar** para terminar.

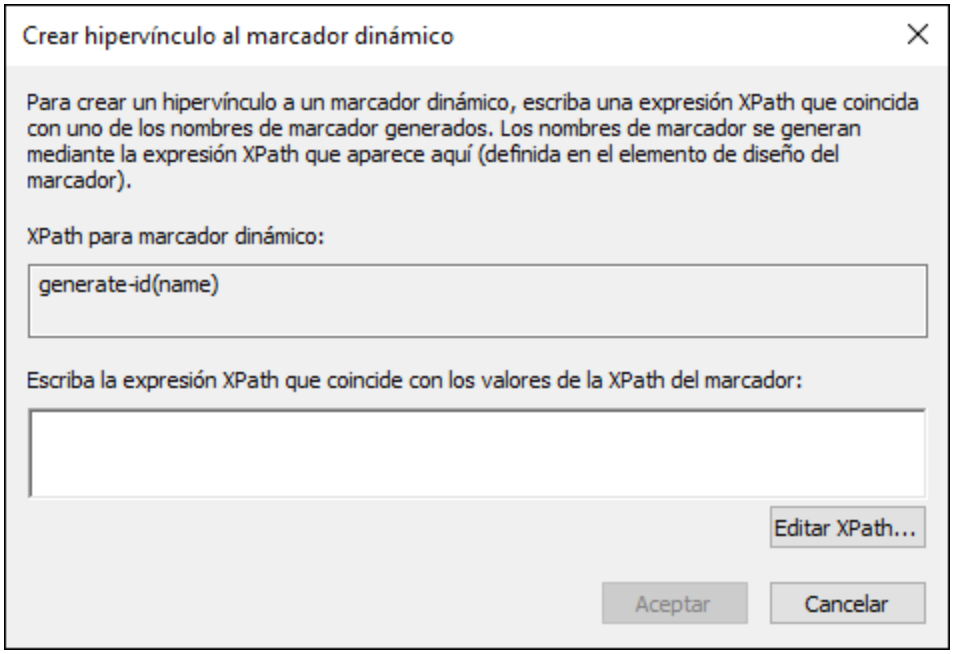
Crear vínculos a marcadores

Siga estos pasos para crear vínculos a marcadores:

1. En la pestaña *Estático* del cuadro de diálogo "Insertar hipervínculo" haga clic en el botón **Marcador**. Aparece otro cuadro de diálogo llamado "Seleccione un marcador del documento" (*imagen siguiente*). En nuestro ejemplo hay dos marcadores disponibles: uno estático y otro dinámico.



2. Para seleccionar un marcador estático como URI de destino, haga doble clic en el marcador estático y después clic en **Aceptar**. Si hace doble clic en un marcador dinámico, aparece otro cuadro de diálogo donde puede escribir la expresión XPath del marcador dinámico seleccionado (*imagen siguiente*).



El marcador dinámico 364 es de hecho una expresión XPath que genera el nombre del marcador y no el nombre del marcador propiamente dicho. El cuadro de diálogo "Crear hipervínculo al marcador dinámico" (*imagen anterior*) muestra la expresión XPath del marcador dinámico y permite construir una expresión XPath que generará un nombre que coincidirá con el del marcador al que se apunta. Cuando termine haga clic en **Aceptar**.

 © 2016-2022 Altova GmbH

Crear vínculos a marcadores con Id. generado dinámicamente

Los marcadores pueden tener anclajes con Id. generados dinámicamente 365. Si quiere crear un vínculo a dicho marcador, hay algo que debe tener en cuenta. Como los nombres de los anclajes generados dinámicamente se generan en tiempo de ejecución y, por tanto, son desconocidos en tiempo de diseño, ¿cómo podemos establecer el valor de `href` de un hipervínculo 826 que apunte a dicho anclaje? La solución es usar la función `generate-id()`, esta vez dentro del valor de `href` del hipervínculo 826. La clave está en una propiedad de la función `generate-id()`. En cada transformación, cada vez que se evalúa la función `generate-id()` para un nodo determinado, siempre se genera el mismo Id. Por este motivo los Id. generados del marcador del marcador y del hipervínculo son los mismos.

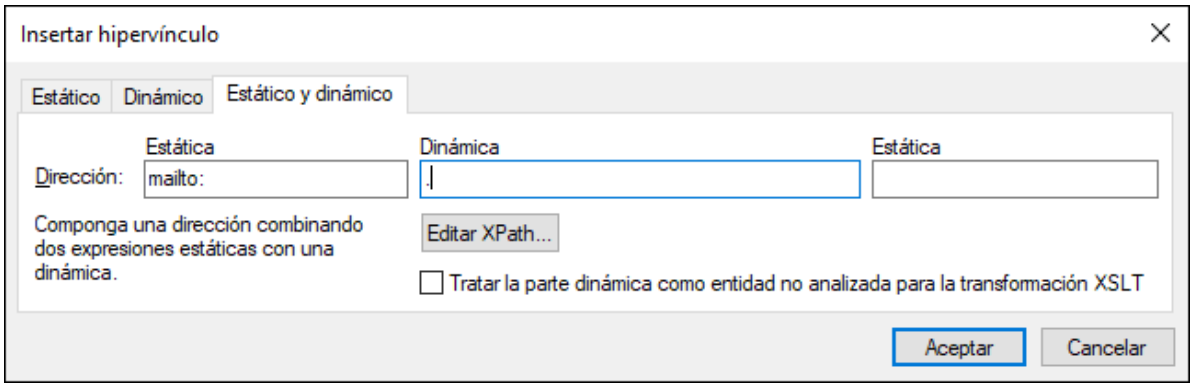
Es necesario tener dos cosas en cuenta:

- Como la función `generate-id()` debe evaluarse como expresión XPath, utilice la pestaña *Dinámico* del cuadro de diálogo "Insertar hipervínculo" (*imagen siguiente*) para establecer el destino del hipervínculo.
- El valor evaluado del atributo `href` debe empezar por `#` (el identificador de fragmento). Como resultado la expresión XPath será: `concat('#', generate-id(nodoXXX))`. Otra opción es introducir, en la pestaña *Estático y dinámico* el identificador de fragmento `#` en la parte estática de la dirección y la función `generate-id(nodoXXX)` en la parte dinámica.

Crear vínculos a recursos externos

Puede construir identificadores URI para encontrar recursos externos:

- Escribiendo el URI directamente en la pestaña *Estático* del cuadro de diálogo "Insertar hipervínculo". Por ejemplo, puede escribir el enlace a una página web (`http://www.altova.com`) directamente en el campo *Dirección* de la pestaña *Estático*.
- Seleccionando un nodo en el documento XML en la pestaña *Dinámico* del cuadro de diálogo "Insertar hipervínculo". El nodo del XML puede aportar una cadena de texto que sea (i) el URI al que se debe apuntar o (ii) el nombre de una entidad sin analizar 469 que tenga el URI necesario como valor. Por ejemplo, la dirección del sitio web de Altova puede ser el contenido de un nodo.
- Creando un URI con partes dinámicas y estáticas en la pestaña *Estático y dinámico* del cuadro de diálogo "Insertar hipervínculo". Esto puede ser útil si quiere añadir prefijos estáticos (p. ej. un protocolo) o un sufijo (p. ej. un nombre de dominio). Por ejemplo, puede usar direcciones de correo electrónico con la parte estática `mailto:` y una parte dinámica que tome el contenido del nodo `//Contact/@email`. El ejemplo de la imagen crea un vínculo en el marcador de posición `(contenido)` del nodo `//Contact/@email` y por eso se usó el selector abreviado `self::node()`. El botón **Editar XPath** abre el cuadro de diálogo Editar expresión XPath 729 para ayudarle a construir la parte dinámica del hipervínculo.



Consulte el apartado URI de entidades sin analizar 469 de este manual para obtener más información sobre cómo usar entidades sin analizar.

Nota: la especificación FO es compatible con el uso de vínculos a documentos externos, pero puede que su procesador FO no lo sea. Si tiene pensado usar vínculos a documentos externos en sus documentos PDF de salida, consulte primero si su procesador FO es compatible con esta característica.

Modificar el contenido del nodo primario en la vista Authentic

El usuario de la vista Authentic puede seleccionar el valor del nodo primario de un hipervínculo 464. Es decir, puede diseñar el documento SPS para que cambie la presentación dependiendo de lo que seleccione el usuario de la vista Authentic.

Editar las propiedades de los hipervínculos

Para editar un hipervínculo haga clic con el botón derecho en la etiqueta de apertura o cierre del hipervínculo y seleccione **Editar URL** en el menú contextual. Esto abre el cuadro de diálogo "Editar hipervínculo" (*imagen anterior*). Este cuadro de diálogo también se puede abrir desde la propiedad `URL` de la ventana Propiedades (grupo *Hipervínculo*).

Quitar y eliminar hipervínculos

Para eliminar un hipervínculo, selecciónelo (haciendo clic en una de sus etiquetas) y pulse la tecla **Suprimir**. El hipervínculo y su contenido se elimina.

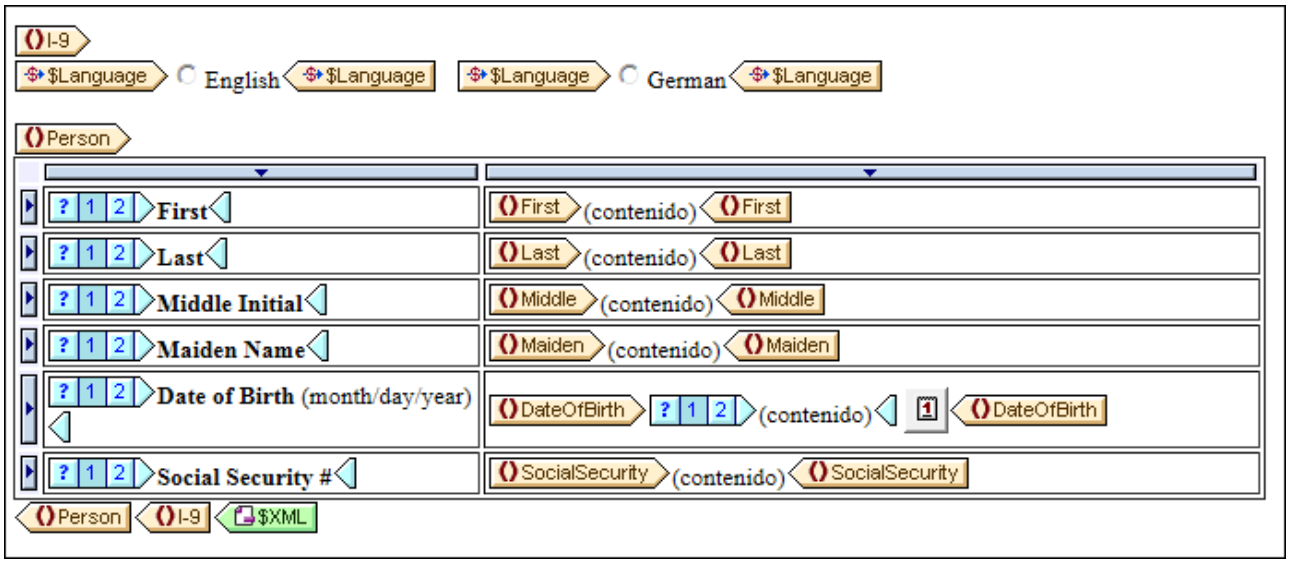
© 2016-2022 Altova GmbH

7.8 Ejemplo: documentos en varios idiomas

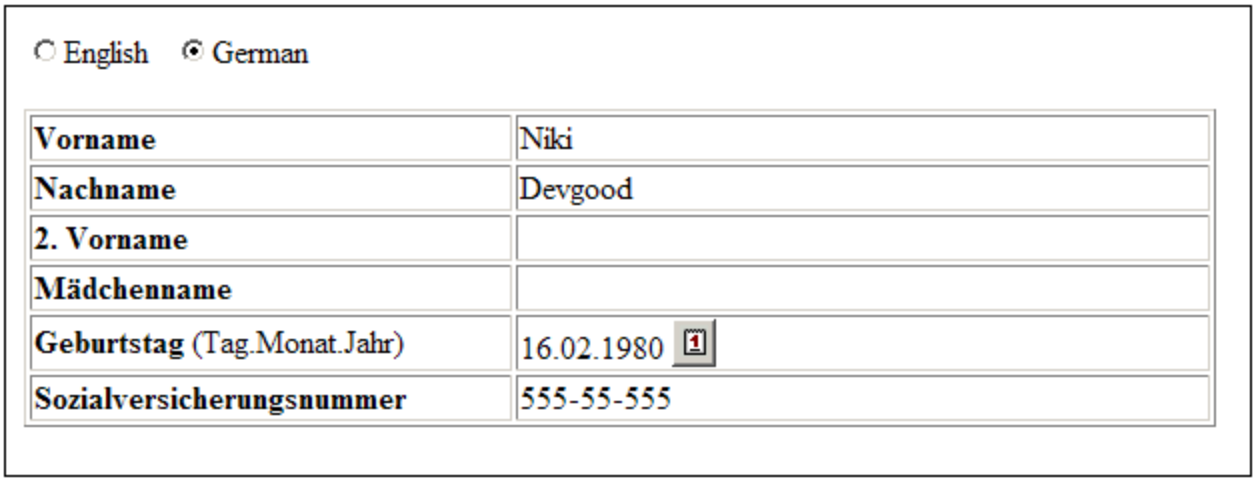
A menudo es necesario incluir contenido en varios idiomas en nuestros documentos y formularios de Authentic y en otras ocasiones se le pide al usuario de los documentos que elija su idioma preferido. StyleVision ofrece varias características enfocadas al trabajo con varios idiomas. A continuación describimos algunas, todas ellas ilustradas en los archivos de ejemplo de la carpeta *Multiple Language* del proyecto `Examples` que viene con StyleVision. Este proyecto debería cargarse automáticamente cuando se carga StyleVision por primera vez pero también se puede cargar a mano con el comando de menú **Proyecto | Abrir**. El proyecto es el archivo `Examples.svp` de la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples`.

Usar variables y condiciones

El idioma elegido por el usuario se puede introducir en unavariable editable 334. Con una condición 303 formada por varias ramas podemos asignar cada idioma al contenido correspondiente. El idioma elegido por el usuario se usa para seleccionar la rama condicional correspondiente.



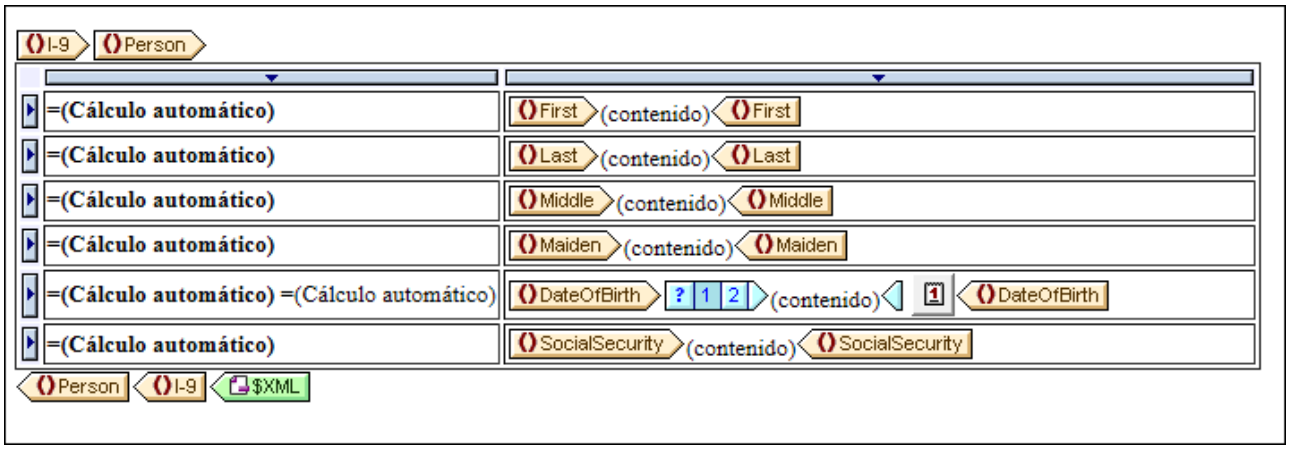
En la imagen anterior puede ver que la elección del usuario se introduce como valor de la variable editable. Las condiciones de la tabla tienen dos ramas para sendos idiomas y comprueban el valor de la variable editable. El resultado en la vista Authentic sería este.



Esta estrategia es ideal para crear formularios en los que el usuario debe seleccionar un idioma. Para más información consulte el archivo `MultiLangByCondition.sps` del proyecto `Examples`.

Usar parámetros y cálculos automáticos

Imagine que necesita generar los mismos datos en varios idiomas diferentes. Una opción sería usar un parámetro 327 cuyo valor desencadene el idioma de salida pertinente. El idioma se puede determinar con ayuda de un cálculo automático 293, por ejemplo. El cálculo automático puede generar el contenido adecuado dependiendo del valor del parámetro.



En el ejemplo de la imagen anterior los cálculos automáticos tienen expresiones XPath de este tipo:

```
if ( $Language = 'E' ) then 'First' else
if ( $Language = 'G' ) then 'Vorname' else ''
```

El valor del parámetro global `$Language` se puede cambiar en el diseño SPS o se puede suministrar desde StyleVision Server en tiempo de ejecución y la transformación se puede ejecutar varias veces para generar los mismos datos en varios idiomas.

Para más información consulte el archivo `MultiLangByAutoCalcs.sps` del proyecto `Examples`.

 © 2016-2022 Altova GmbH

Archivos de ejemplo

Para ver más ejemplos con varios idiomas abra el archivo de proyecto `Examples.svp`, que está en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples`.

8 Archivos SPS: presentación

StyleVision ofrece un conjunto de características para aplicar estilos a los componentes del diseño SPS. Estos estilos se convierten en el marcado de estilo correspondiente en los documentos de salida (*vista Authentic*, *HTML, texto, RTF, PDF y Word 2007+ en la edición Enterprise Edition; vista Authentic, HTML, RTF y texto en la edición Professional Edition; HTML en la edición Basic Edition*). Algunos efectos de presentación, como los efectos de presentación web interactivos (p. ej. cuadros combinados y controladores de eventos JavaScript), no pueden aparecer en documentos de salida para formatos impresos (RTF). En estos casos los medios impresos representarán de forma gráfica este tipo de componentes. Sin embargo, StyleVision ofrece algunas opciones de definición de página 419 esenciales para los medios impresos. Estas opciones para medios impresos 419, como tamaño de la página, diseño de la página, encabezado y pie de página se definen por separado y se usan solamente para los documentos de salida RTF.

Nota: como el texto plano no lleva formato, las propiedades de estilo del texto y el diseño de página se ignoran al generar documentos de salida de texto.

Aplicación de estilos a los componentes del SPS

Todos los estilos de los componentes del SPS se aplican usando la sintaxis CSS2. Los estilos se pueden definir en hojas de estilos externas, de forma global para todo el SPS o de forma local para un componente concreto. El orden en cascada de CSS2 afecta al diseño SPS y ofrece una gran flexibilidad a la hora de diseñar estilos. Para aprender a trabajar con estilos CSS consulte el apartado Trabajar con estilos CSS 390.

Los valores de las propiedades de estilo se pueden editar directamente en las ventanas Estilos y Propiedades o se pueden definir mediante expresiones XPath 402. Las principales ventajas de usar expresiones XPath son: (i) que el valor de una propiedad puede tomarse de un archivo XML y (ii) que el valor de una propiedad se puede asignar de forma condicional según una prueba incluida en la expresión XPath.

Además en el diseño SPS hay algunos elementos HTML que están disponibles como marcado para los componentes del SPS. Estos formatos predefinidos 375 se envían a los documentos de salida HTML. El formato inherente en este marcado también se usa para aplicar estilos a los componentes del SPS. Cuando se aplican estilos CSS a formatos predefinidos, los estilos CSS tienen prioridad sobre el estilo inherente del formato predefinido. Los formatos predefinidos se describen en el apartado Formatos predefinidos 375 de esta sección. No olvide que los estilos inherentes de los formatos predefinidos se convierten en el marcado equivalente para formatos de salida RTF.

Nota: cuando defina estilos CSS para un componente del SPS debe tener en cuenta que ciertos estilos no se pueden aplicar en los formatos de salida de medios impresos (RTF). Además, si se usan selectores HTML (en hojas de estilos externas y en reglas de estilo globales), estos no afectarán a los formatos de salida de medios impresos (RTF). Cuando se usan estos selectores, aparece un comentario junto a cada selector explicando que el estilo no se aplicará a los formatos RTF.

Diseñar el documento SPS para medios impresos

StyleVision ofrece opciones de diseño y definición de páginas 419 para diseños para medios impresos (formato de salida RTF y hojas de estilos XSLT para RTF). Estas opciones se usan junto con los demás estilos y se describen en el apartado Diseñar medios impresos 419 de esta sección.

 © 2016-2022 Altova GmbH

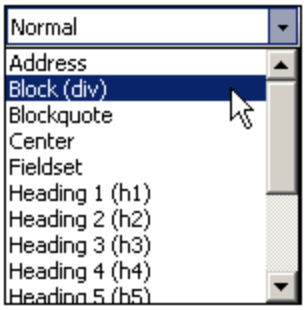
8.1 Formatos predefinidos

StyleVision ofrece varios formatos predefinidos, correspondientes a elementos HTML (*imagen siguiente*). Cuando aplique un formato predefinido a un componente en la vista **Diseño**, el componente se marca con la semántica HTML correspondiente. Esto tiene dos consecuencias:

- Se aplica el formato inherente al formato predefinido seleccionado.
- El componente se incluye dentro del tipo de componente `párrafo` lo cual permite aplicarle estilos locales 398.

Asignar formatos predefinidos

Para asignar un formato predefinido a un componente del diseño SPS seleccione el componente, haga clic en el comando de menú **Insertar | Párrafo especial** y seleccione el formato deseado. También puede seleccionar el componente y después elegir el formato en la lista desplegable de formatos predefinidos de la barra de herramientas "Formato" (*imagen siguiente*).



Estilos inherentes

Los formatos predefinidos utilizados en StyleVision tienen uno de estos dos componentes de estilo o ambos:

- el componente de estilo de texto
- el componente de espaciado

Por ejemplo: el formato predefinido `para (párrafo)` tiene solo el componente de espaciado: inserta espacio vertical antes y después del componente al que se le aplica el formato, pero no le aplica estilo de texto. Por otro lado, el formato predefinido `título de primer nivel (h1)` tiene tanto el componente de estilo de texto como el de espaciado.

Es importante tener en cuenta estos aspectos de los formatos predefinidos:

- El componente de espaciado de un formato predefinido se aplica a todos los componentes SPS, pero el estilo de texto no se puede aplicar a todos los componentes. Por ejemplo, si selecciona una imagen y le aplica el formato predefinido `título de primer nivel (h1)`, solamente se aplica el componente de espaciado y no el de estilo de texto.
- El componente de estilo de texto de un formato predefinido no se aplica a los dispositivos de entrada de datos.
- A un componente del diseño SPS no se le puede aplicar más de un formato predefinido a la vez.
- El formato predefinido `texto con formato predefinido (pre)` aplica el formato que aplica la tabulación `pre` de HTML: los saltos de línea y el espaciado del texto se mantienen y se usa una fuente monoespaciada (p. ej. Courier). Si quiere usar líneas continuas sin saltos de línea, entonces

elija el formato predefinido `texto con formato predefinido (pre)` para mostrar las líneas de texto sin ajuste de texto. Si quiere ajustar el texto, use el formato predefinido `texto con formato predefinido y ajuste de texto (pre-wrap)`.

Definir estilos adicionales para un formato predefinido

Puede añadir estilos al estilo inherente de un formato predefinido. Para ello seleccione el formato predefinido y aplíquele un estilo local desde la ventana Estilos 398.

La tecla Entrar y los formatos predefinidos

Cuando se pulsa la tecla **Entrar** en la vista Authentic dentro del contenido de un elemento que tiene aplicado un formato predefinido, se completa la instancia del elemento actual y su bloque y se inserta una instancia nueva del elemento y un bloque nuevo. Esta propiedad es ideal si quiere que el usuario de la vista Authentic sea capaz de crear un elemento nuevo (p. ej. un elemento de tipo párrafo) con solo pulsar la tecla **Entrar**.

© 2016-2022 Altova GmbH

8.2 Caracteres de escape en documentos de salida

Un carácter de escape es un carácter escrito como referencia de caracteres o referencia de entidad. Ambos tipos de referencias (de caracteres y de entidad) se delimitan con una Y comercial al principio y un punto y coma al final. Por ejemplo:

- la referencia de carácter hexadecimal (o Unicode) del carácter `A` es `A`
- la referencia de carácter decimal del carácter `A` es `A`
- la referencia de entidad HTML (y XML) del carácter `&` es `&`
- la referencia de carácter hexadecimal (o Unicode) del carácter `&` es `&`
- la referencia de carácter decimal del carácter `&` es `&`
- la referencia de entidad HTML (y XML) del carácter `<` es `<`

Caracteres de escape de salida

Con *caracteres de escape de salida* nos referimos al modo en que se representan en el formato de salida los caracteres con escape de los datos de entrada. Por tanto, decimos que un carácter tiene escape de salida cuando en el documento de salida se representa como referencia de carácter o de entidad. Tenga en cuenta que un carácter solamente puede tener escape de salida si tiene escape en los datos de entrada (*ver tabla más abajo*). En los diseños SPS puede habilitar/deshabilitar el escape de salida de:

- fragmentos de texto estático
- el marcador de posición `(contenido)` y
- cálculos automáticos

Esto se hace con la propiedad `disable-output-escaping` del grupo de propiedades *Texto*. El valor predeterminado de esta propiedad es `no`, lo cual significa que el escape de salida no se deshabilita. De modo que los caracteres con escape de los datos de entrada **tendrán escape** en el documento de salida (*ver tabla más abajo*).

Para deshabilitar el escape de salida:

1. Seleccione (i) el texto estático, (ii) el fragmento de texto estático, (iii) el marcador de posición `(contenido)` o (iv) el cálculo automático cuyo escape de salida desea deshabilitar.
2. En la ventana Propiedades expanda el grupo de propiedades *Texto* y elija un valor para la propiedad `deshabilitar-escape-de-salida`:

   - `Para HTML` (deshabilita el escape de salida en documentos de salida HTML).
   - `Para Authentic` (deshabilita el escape de salida en la vista Authentic). Esta opción solamente es compatible con las ediciones Enterprise Edition de StyleVision, Authentic Desktop, Authentic Browser y XMLSpy).
   - `Para RTF` (deshabilita el escape de salida en documentos de salida RTF).
   - `Para todos` (deshabilita el escape de salida en todos los formatos de salida excepto texto).

Nota: el escape de salida no afecta al texto. (No está habilitado ni deshabilitado.)

Cuando el escape de salida se deshabilita en un formato de salida concreto (por ejemplo, en HTML), el texto seleccionado no aparecerá con escape en ese formato de salida, pero sí en los demás.

A continuación aparecen algunos ejemplos con el escape de salida habilitado y deshabilitado.

Texto estático	deshabilitar-escape-de-salida	Texto de salida
&	no	&
&	sí	&
&	no	&
&	sí	&
<	no	<
<	sí	<
A	no	A
A	sí	A
<	no	<
<	sí	<
<	sí	<
&<	sí	&<

Nota: la opción **deshabilitar-escape-de-salida** es compatible con la vista Authentic de las ediciones Enterprise Edition de los productos de Altova.

Nota sobre el uso de la opción **deshabilitar-escape-de-salida**

Si deshabilita el escape de salida, la cadena de texto puede tener transcendencia en un formato de salida, pero ser irrelevante en otro. Por ejemplo, el texto de entrada que aparece a continuación incluye caracteres con escape:

```
&lt;b&gt;This text is bold.&lt;/b&gt;
```

Si deshabilitamos el escape de salida, el texto aparecerá así en el documento de salida:

```
<b>This text is bold.</b>
```

Si el escape de salida se deshabilita en el formato de salida HTML y su resultado se ve en un navegador (en lugar de en un editor de texto), el marcado tendrá un significado para el explorador HTML y el texto aparecerá en negrita:

This text is bold.

Sin embargo, si el resultado se ve en otro formato de salida, como PDF, el marcado que tenía un significado en HTML no tiene relevancia alguna en este otro formato de salida. El texto tendría este aspecto en el formato de salida PDF:

```
<b>This text is bold.</b>
```


© 2016-2022 Altova GmbH

En otras palabras, el texto de salida obtenido al deshabilitar el escape de salida puede interpretarse como código en algunos formatos de salida pero no en otros. Esto debe tenerse en cuenta cuando se use la propiedad `disable-output-escaping`.

8.3 Formato de valores (formato de tipos de datos numéricos)

Gracias a la característica *formato de valores* de StyleVision el contenido de nodos de tipo de datos XML Schema numérico (*ver lista* 380) puede aparecer con una representación distinta a la representación léxica de dicho tipo de datos. Por ejemplo, la representación léxica de un nodo de tipo de datos `xs:date` es `YYYY-MM-DD`, con un componente opcional de zona horaria, como `+02:00`.) El formato de valor se presenta en la vista Authentic y, dependiendo de la definición de formato, también en los documentos de salida HTML, texto y RTF. La característica *formato de valores* también se puede usar para dar formato al resultado de un cálculo automático si el resultado del cálculo está en el formato léxico de uno de los tipos de datos numéricos (*ver lista* 380) a los que se puede aplicar el formato de valor.

Los apartados de esta sección describen:

- el funcionamiento del formato de valores 380 y
- la sintaxis 383 necesaria para definir el formato de valor.

Nota: el formato de valores no cambia el formato en el que se guardan los datos en el documento XML. En el documento XML válido los datos se almacenan siempre en el formato léxico adecuado para el tipo de datos del nodo. El formato de valor afecta a la presentación en la vista Authentic y, si corresponde, a la presentación en los diferentes formatos de salida.

Tipos de datos numéricos para los que hay formato de valores

La característica *formato de valores* está disponible para estos tipos de datos:

- `xs:decimal`; `xs:integer`; los 12 tipos integrados derivados de `xs:integer`
- `xs:double` y `xs:float` cuando los valores están comprendidos entre 0.000001 y 1,000,000. Los valores que no estén comprendidos en este intervalo aparecen en forma de notación científica (p. ej. `1.0E7`) y no se les puede aplicar un formato de valor.
- `xs:date`; `xs:dateTime`: `xs:duration`
- `xs:gYear`; `xs:gYearMonth`; `xs:gMonth`; `xs:gMonthDay`; `xs:gDay`

8.3.1 Funcionamiento

Puede aplicar formato de valor a:

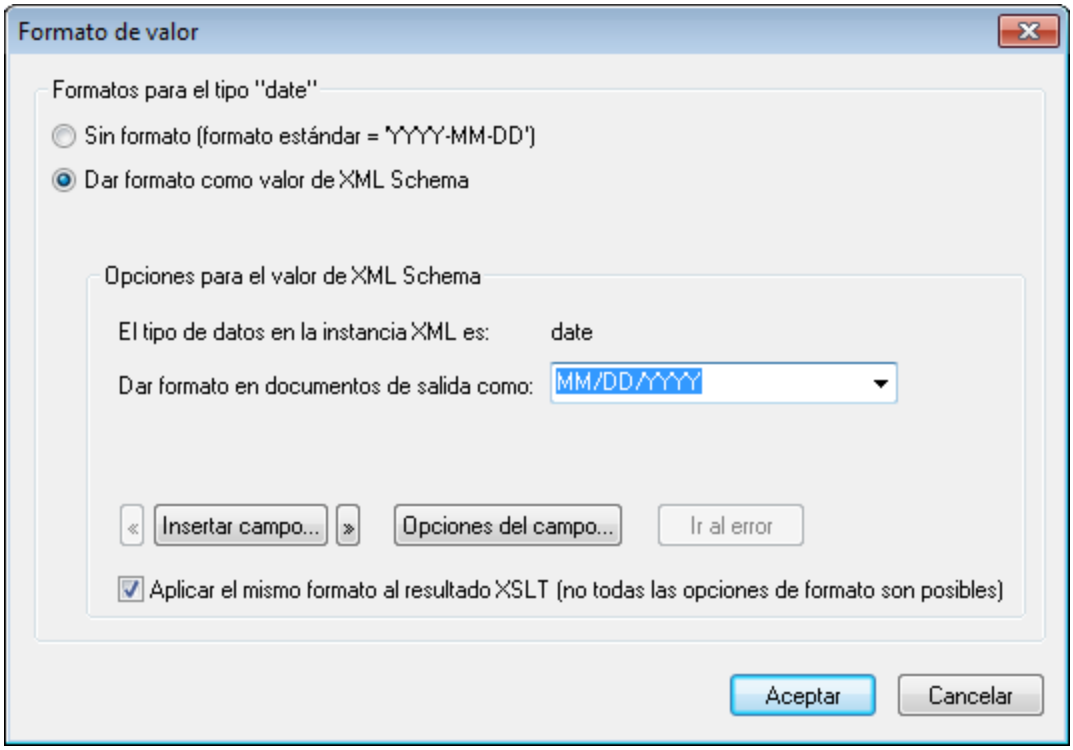
- Un nodo de tipo de datos numérico 380, como `xs:decimal` o `xs:date`, que esté presente en el diseño SPS como `(contenido)` o como campo de entrada.
- Un cálculo automático cuyo resultado tenga el formato léxico de un tipo de datos numérico 380.

Definir el formato de valor

Para definir el formato de valor de un nodo o de un cálculo automático del SPS:

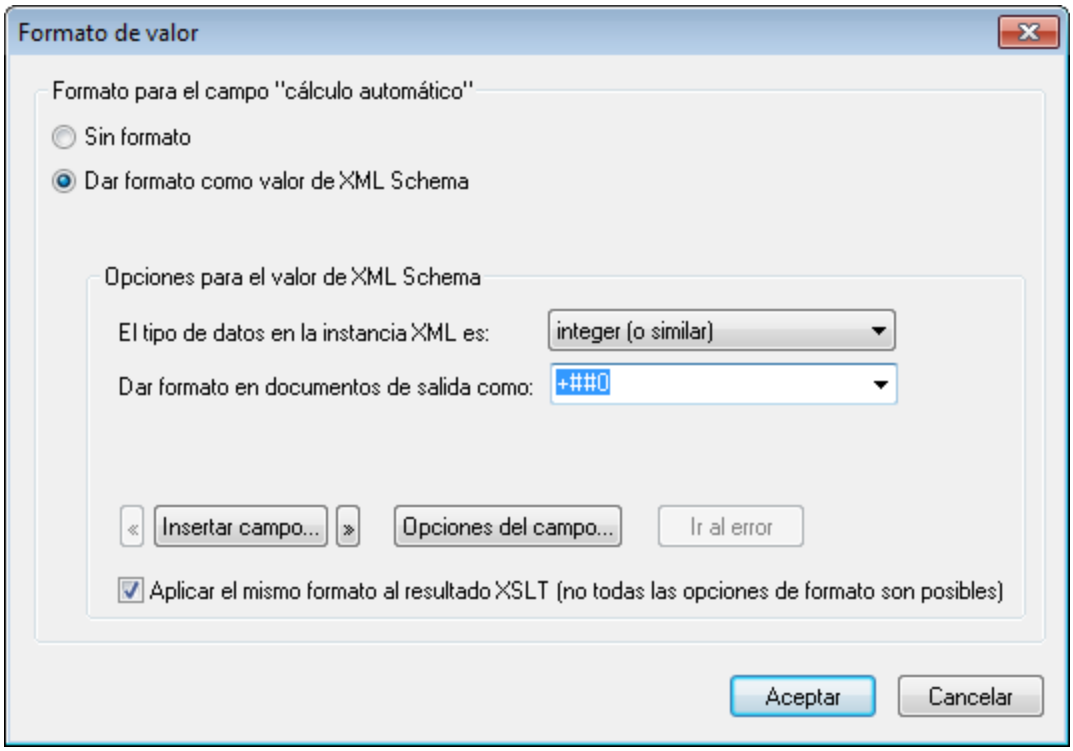
1. Seleccione el marcador de posición `(contenido)` o el campo de entrada del nodo o el cálculo automático.
2. En la ventana Propiedades, seleccione el componente y expanda el grupo *contenido* o *cálculo automático*. Haga clic en el botón de edición ... de la propiedad `formato de valor`. Otra manera de

hacer esto es hacer clic con el botón derecho en el componente y seleccionar **Editar formato de valor** en el menú contextual. Ahora se abre el cuadro de diálogo "Formato de valor" (*imagen siguiente*), cuyas opciones dependen del tipo de componente seleccionado. Si el componente seleccionado es un nodo, aparece un cuadro de diálogo como el de la imagen. El nodo utilizado para este ejemplo es un nodo de tipo `xs:date`.



Observe que el cuadro de diálogo dice *Formatos para el tipo "date"* y que el formato estándar para el tipo de datos `xs:date` aparece en el primer botón de opción *Sin formato*. Por tanto, esta información del cuadro de diálogo varía dependiendo del tipo de datos.

Si el componente seleccionado es un cálculo automático, el cuadro de diálogo tendría este aspecto.



3. Ahora debe especificar si la presentación del valor del componente tiene formato o no. Si quiere dejar el resultado sin formato, seleccione el primer botón de opción *Sin formato.* Si quiere que tenga formato, seleccione el botón de opción *Dar formato como valor de XML Schema*. Si el valor está sin formato, el resultado tiene el formato estándar del tipo de datos del nodo seleccionado o del tipo de datos del resultado del cálculo automático. Si eligió la opción *Dar formato como valor de esquema XML* para un cálculo automático, tendrá que seleccionar también (en una lista desplegable) el tipo de datos del resultado esperado del cálculo automático.
4. Escriba la definición de formato de valor. Esta definición puede introducirse de varias formas: (i) seleccionando las opciones disponibles para el tipo de datos en el campo *Dar formato en documentos de salida como*; (ii) introduciendo la definición en el campo de entrada directamente y (iii) usando los botones **Insertar campo** y **Opciones del campo** para generar la definición correctamente. Para más información sobre las opciones de formato disponible consulte el apartado Sintaxis de formato de valores 383.

Errores de sintaxis

Si hay un error en la sintaxis del formato de valor, ocurre esto:

- La definición aparece en color rojo.
- Aparece un mensaje de error de color rojo debajo del campo de entrada.
- Se deshabilita el botón **Aceptar** del cuadro de diálogo.
- Se habilita el botón **Ir al error** del cuadro de diálogo. Al hacer clic en este botón el cursor se coloca en el punto de la definición donde está el error.

 © 2016-2022 Altova GmbH

Desajuste entre el formato de los datos y el tipo de datos

Si los datos introducidos en un nodo XML no se ajustan al formato léxico del tipo de datos de ese nodo (o si el resultado de un cálculo automático no se ajusta al formato léxico del tipo de datos esperado), entonces el formato estará sin definir y no aparecerá bien en el documento de salida.

Aplicar el formato de valor a los documentos de salida

El formato de valor que defina en el cuadro de diálogo "Formato de valor" se aplica a la vista Authentic, disponible en las ediciones Enterprise y Professional Edition.

Algunas definiciones de formato de valor (no todas) también se aplican a los documentos de salida HTML, si marca la casilla *Aplicar el mismo formato al resultado XSLT* del cuadro de diálogo "Formato de valor". Si no marca esta opción o si la opción no está disponible, solamente la vista Authentic mostrará el formato de valor definido. Los demás documentos de salida mostrarán el valor en el formato predeterminado del tipo de datos del componente (su formato léxico).

8.3.2 Sintaxis

La sintaxis para definir un formato de valor es:

([carácter/caracteres prefijo]campo[carácter/caracteres sufijo][{campo-opción1,campo-opción2,...}])+

siendo **carácter/caracteres prefijo** y **carácter/caracteres sufijo** caracteres opcionales que controlan la alineación y la presentación de símbolos negativos/positivos; **campo** puede ser cualquier formato o texto propio del tipo de datos; y **{campo-opciones}** es un calificador opcional que habilita opciones de formato adicionales.

Explicación sobre la sintaxis de definición

Así se construye la definición de formato de valor:

- La definición está compuesta por un campo o varios. Por ejemplo, la definición `DD Month YYYY` tiene tres campos.
- Los campos pueden ir juntos o pueden estar separados por estos caracteres: espacio, guión, coma, dos puntos, punto o por una cadena de texto entre comillas simples o dobles. Por ejemplo, en la definición `DD-Month' en el año 'YYYY`, los campos `DD` y `Month` están separados por un guión y los campos `Month` y `YYYY` están separados por una cadena de texto entre comillas simples.
- Un campo puede tener caracteres prefijo / sufijo. Por ejemplo: `<+###,##0.00`.
- Un campo puede tener una opción de campo o varias. Las opciones de campo de cada campo deben tener un conjunto de llaves y deben seguir al campo sin estar separados por espacios. Si usa varias opciones para un campo, entonces sepárelas con una coma ",". Por ejemplo, en la definición `DD Month{uc,ro} YYYY`, las opciones de campo `uc` y `ro` que aparecen entre llaves son las opciones del campo `Month`.

Ejemplos

Ejemplo de formato de valor para un tipo de datos `xs:decimal`:

```
"$"(##0.00)
```

Ejemplos de resultados:

```
$ 25.00
$ 25.42
$267.56
```

Ejemplo de formato de valor para un tipo de datos `xs:date`:

```
DD Month{uc,ro} YYYY
```

siendo `uc` y `ro` opciones de campo para convertir el campo del mes `Month` en mayúsculas y solo lectura

Ejemplo de resultado:

```
24 SEPTEMBER 2003
```

Tipos de campo

Un tipo de campo equivale a un componente de los datos y la forma en que se debe dar formato al componente. El formato inherente al tipo de campo se puede modificar con prefijos y sufijos modificadores así como con opciones de campo. Las tablas que aparecen a continuación enumeran todos los tipos de campo disponibles. No olvide que puede modificar las opciones del menú desplegable del cuadro de diálogo "Editar campo" con ayuda de los prefijos y sufijos modificadores y de las opciones de campo.

Tipo de campo	Definición
#	espacio si no hay un dígito en esta posición
0	cero si no hay un dígito en esta posición
.	marca decimal
,	separador de grupo de dígitos
Y	Año
y	año (base = 1930); ver nota más abajo
MM	Mes, debe tener una longitud de 2
DD	Día, debe tener una longitud de 2
W	Número de la semana
d	Número del día de la semana (del 1 al 7)
i	Día del año (del 1 al 366)
hh	Hora (de 0 a 23), debe tener una longitud de 2
HH	Hora (de 0 a 12), debe tener una longitud de 2

© 2016-2022 Altova GmbH

mm	Minuto, debe tener una longitud de 2
ss	Segundo, debe tener una longitud de 2
AM	AM o PM
am	am o pm
AD	AC o DC
ad	ac o dc
CE	EC o AEC
ce	ec o aec

Tipo de campo	Definición
Weekday	día de la semana (Lunes, Martes...)
WEEKDAY	día de la semana (LUNES, MARTES...)
weekday	día de la semana (lunes, martes...)
Wkd	día de la semana (Lun, Mar...)
WKD	día de la semana (LUN, MAR...)
wkd	día de la semana (lun, mar...)
Month	mes (Enero, Febrero...)
MONTH	mes (ENERO, FEBRERO...)
month	mes (enero, febrero...)
Mon	mes (Ene, Feb...)
MON	mes (ENE, FEB...)
mon	mes (ene, feb...)

Notas sobre la longitud del campo y de la entrada

Debe tener en cuenta estos aspectos sobre la longitud de los componentes de datos:

Longitud de los campos de fecha: cuando utilice campos como `MM`, `DD`, `HH`, `hh`, `mm` y `ss`, compruebe que tienen una longitud de 2. Cuando utilice los campos `y` o `Y`, el número de caracteres `y` o `Y` de la definición determina la longitud del resultado. Por ejemplo, si especifica `YYY`, el resultado del valor `2006` sería `006`; pero si la definición fuera `YYYYYY`, el valor sería `002006`.

Aumentar la longitud del campo: el símbolo * se usa para aumentar la longitud de un campo numérico no semántico (enteros, decimales, etc.). En el caso de los decimales, puede usar el asterisco a ambos lados del punto decimal. Por ejemplo, el valor de formato `*0.00*` hace que en la posición de cada asterisco aparezca un cero.

Longitud de las entradas en la vista Authentic: la vista Authentic del contenido de un nodo se vasa en la definición de formato de valor del nodo. Por tanto, el usuario de la vista Authentic no podrá insertar más caracteres de los permitidos por la definición de formato de valor. Esto es ideal para restringir la entrada de datos en la vista Authentic. Sin embargo, recuerde que si la longitud de un valor ya existente del documento XML supera la longitud especificada por la definición de formato, entonces aparecerá todo el valor.

Nota: si un campo no genera texto en el documento de salida, puede ser debido a su configuración regional en Windows. Por ejemplo, Windows devuelve una cadena vacía para el campo AM/PM si la opción de idioma regional es Alemán.

Prefijos y sufijos modificadores

Los prefijos y sufijos modificadores se usan para modificar la alineación del texto y la representación positiva/negativa de los campos. Esta tabla muestra los prefijos y sufijos disponibles.

 © 2016-2022 Altova GmbH

Prefijo	Sufijo	Definición
<		Alineado a la izquierda. Formato predeterminado para texto. Para números, que se alinean a la derecha por defecto, debe tener en cuenta el número de espacios iniciales.
>		Alineado a la derecha. Formato predeterminado para números.
?		Símbolo menos junto al número si es negativo. Si no, nada. Formato predeterminado para los números.
<?		Símbolo menos a la izquierda si es negativo. Si no, nada. El número alineado a la izquierda y después el símbolo menos.
<?>		Símbolo menos a la izquierda si es negativo. Si no, nada. El número alineado a la derecha.
-	-	Símbolo menos junto al número si es negativo. Si no, espacio. Situado antes del número (prefijo), después del número (sufijo).
<-	>-	Símbolo menos si es negativo. Si no, espacio. El número junto al símbolo menos. Alineados a la izquierda (prefijo), alineados a la derecha (sufijo).
<->		Símbolo menos a la izquierda si es negativo. Si no, espacio. El número alineado a la derecha.
+	+	Símbolo menos o más siempre, junto al número. Antes del número (prefijo), después del número (sufijo)
<+	>+	Símbolo menos o más siempre, junto al número. Alineado a la izquierda (prefijo), alineado a la derecha (sufijo)
<+>		Símbolo menos o más siempre, alineado a la izquierda. El número alineado a la derecha.
(	)	Paréntesis si es negativo. Si no, espacio. Situado junto al número.
<(		Paréntesis si es negativo. Si no, espacio. Situado junto al número. Alineado a la izquierda.
<(>		Paréntesis si es negativo. Si no, espacio. Paréntesis izquierdo alineado a la izquierda. El número y el paréntesis derecho uno junto al otro y alineados a la derecha.
[	]	Paréntesis si es negativo. Si no, nada. Situado junto al número.
*	*	Número de dígitos ampliable a la izquierda (prefijo) o a la derecha (sufijo)
_	_	Espacio
^	^	Carácter de relleno (definido en las opciones)
	th	Escala ordinal del número en inglés (st, nd, rd o th)
	TH	Escala ordinal del número en inglés y mayúsculas (ST, ND, RD o TH)

Opciones de campo

Las opciones de campo sirven para realizar algunos cambios en los campos del valor de formato. Estas son las opciones disponibles:

Opción	Definición
uc	Mayúsculas
lc	Minúsculas
left	Alineado a la izquierda
right	Alineado a la derecha
ro	Solo lectura (XML), edición no permitida
edit	El campo es editable (opción activa por defecto)
dec=<char>	Especificar un carácter para el punto decimal (el punto es la opción predeterminada)
sep=<char>	Especificar un carácter para el separador de dígitos (la coma es la opción predeterminada)
fill=<char>	Especificar el carácter de relleno
base=<year>	Año base para los campos de año (*ver nota más abajo*)
pos	Solo mostrar los números positivos. Está permitido introducir números negativos

Debería usar las opciones de campo para generar formatos numéricos en idiomas europeos, que usan las comas y los puntos de forma diferente: p. ej `123.456,75`.

El formato de valor necesario para obtener el formato de este ejemplo sería: `###,###.##{dec=,,sep=.}`

Observe que el campo retiene el formato inglés, mientras que las opciones de campo `dec` y `sep` especifican el símbolo decimal y el separador de dígitos. Si el símbolo decimal y separador de dígitos no se especifican, se usará el símbolo decimal y el separador de dígitos predeterminado de la configuración regional del sistema operativo Windows.

Nota sobre el año base

Cuando use formatos de año cortos (como `yy` y `YY`), el año base especifica el corte/límite para un siglo. Por ejemplo, la opción de campo de año base podría usarse en la definición `DD-MM-YY{base=1940}`. Si el usuario introduce un valor igual o mayor que los dos últimos dígitos del año base (que se consideran un entero positivo de dos cifras), el siglo será el mismo que el del año base. Si el valor introducido por el usuario es menor al valor entero de los dos últimos dígitos, entonces el siglo será el siglo del año base más uno. Por ejemplo, si la definición dice `base=1940` y el usuario introduce `50`, el valor que se introduce en el documento XML será `1950`; pero si el usuario introduce `23`, el valor que se introduce en el documento XML será `2023`.

Recuerde que:

 © 2016-2022 Altova GmbH

- Aunque lo más común es usar dos dígitos como formato de año corto, también se pueden usar formatos de año con uno y tres dígitos y con un año base.
- Los tipos de datos que se pueden usar para los formatos de año corto son: `xs:date`, `xs:dateTime`, `xs:gYear` y `xs:gYearMonth`.
- Si el formato de valor se define para un cálculo automático, asegúrese de seleccionar el tipo de datos correcto en el cuadro de diálogo "Formato de valor". El tipo de datos seleccionado debería ser el del resultado que obtiene el cálculo automático.
- Si usa el tipo de campo `yy`, el año base predeterminado es 1930. Si establece un año base se invalida el año base predeterminado.
- Si usa el tipo de campo `YY` sin especificar un año base, el usuario de la vista Authentic solamente podrá modificar los dos últimos dígitos del valor de año de cuatro dígitos. Los dos primeros dígitos se quedan tal y como están en el documento XML.

8.4 Trabajar con estilos CSS

El documento SPS se diseña con ayuda de estilos y reglas CSS. Las reglas de estilo se pueden definir:

- En hojas de estilos CSS externas 391. Las hojas de estilos CSS externas se añaden desde las ventanas Vista general del diseño 45 y Repositorio de estilos 55.
- En estilos globales 395 para el documento SPS, que se definen al principio del propio SPS. En los documentos de salida HTML estos estilos globales se definen dentro del elemento secundario `style` del elemento `head`. Los estilos globales se definen en la ventana Repositorio de estilos 55.
- De forma local 398, en los diferentes componentes del documento. En los documentos de salida HTML estas reglas se definen en el atributo `style` de cada elemento HTML. Las reglas de estilo locales se definen en la ventana Estilos 58.

Los apartados de esta sección describen los diferentes métodos para crear estilos en StyleVision.

Nota: como los estilos CSS no se pueden aplicar al texto plano, no afectarán a la salida en formato texto.

Terminología

Una hoja de estilos CSS se compone de reglas de estilo. Este es el formato de una regla:

```
H1 { color: blue }
```

o

```
H1 { color: blue;
         margin-top: 16px; }
```

Una regla de estilo está formada por un selector (`H1` en el ejemplo anterior) y una declaración (`color: blue`). La declaración es una lista de propiedades (por ejemplo: `color`) con valores (`blue`). Cada pareja de propiedad-valor se denomina *definición de estilo*. En StyleVision los estilos CSS se definen en las ventanas Estilos 58 (estilos locales) y Repositorio de estilos 55 (estilos globales).

Orden en cascada

En el documento SPS se aplica el orden en cascada de CSS. Es decir, para determinar qué reglas tienen prioridad se tiene en cuenta:

1. ***El origen:*** las hojas de estilos externas tienen menor prioridad que los estilos globales y los estilos globales tienen menor prioridad que los estilos locales. Las hojas de estilos externas son básicamente hojas de estilos importadas y el orden de importación es importante (tiene prioridad la última hoja de estilos que se importó).
2. ***La precisión:*** si dos reglas se aplican al mismo elemento, tiene prioridad la regla cuyo selector sea más preciso.
3. ***El orden:*** si dos reglas tienen el mismo origen y la misma precisión, entonces tiene prioridad la regla que aparezca más tarde en la hoja de estilos. Además se entiende que las hojas de estilos importadas aparecen antes que el conjunto de reglas de la hoja de estilos donde se importó.

© 2016-2022 Altova GmbH

Estilos CSS en archivos SPS modulares

Cuando se añade un módulo SPS a un diseño SPS, los estilos CSS del SPS de destino tienen prioridad sobre los estilos del módulo SPS añadido. Cuando se añaden varios módulos, tienen prioridad los estilos CSS del módulo situado más abajo en la lista de módulos. Para más información consulte la sección Archivos SPS modulares 251.

Compatibilidad de CSS con Internet Explorer

Las versiones de Internet Explorer (IE) anteriores a la versión IE 6.0 no interpretan ciertas reglas CSS igual que las versiones IE 6.0 y posteriores. Por eso el diseñador del documento SPS debería saber para qué versión de IE está diseñando el documento. IE 6.0 (y posterior) es compatible tanto con las interpretaciones antiguas como con las nuevas, lo cual permite usar las interpretaciones antiguas de CSS en las versiones más recientes de IE. La interpretación utilizada por IE 6.0 (y superior) viene determinada por un modificador del código del documento HTML. En el SPS puede especificar 787 si los documentos de salida HTML y la vista Authentic deberían diseñarse según la interpretación antigua o nueva de Internet Explorer 116. Los estilos CSS deberían definirse según la interpretación seleccionada. Para más información consulte el apartado Propiedades: compatibilidad con CSS 787.

Nota: para más información sobre la especificación CSS visite http://www.w3.org/TR/REC-CSS2/.

8.4.1 Hojas de estilos externas

Este apartado describe cómo gestionar hojas de estilos CSS externas desde la interfaz gráfica de StyleVision. Hay varias fases:

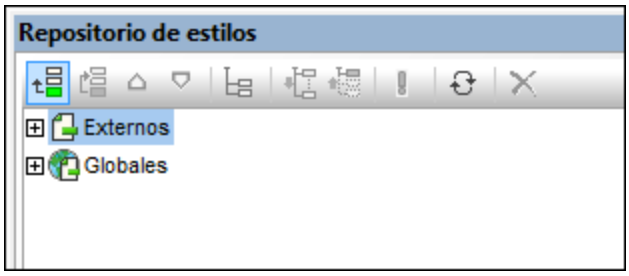
- Añadir una hoja de estilos CSS externa al SPS 391
- Ver el contenido de una hoja de estilos CSS externa y definir su aplicación a los distintos medios 392
- Cambiar el orden de prioridad 393
- Alternar el conjunto de hojas de estilos CSS y una sola hoja de estilos CSS 394

Las hojas de estilos CSS externas se pueden gestionar desde dos ventanas: el Repositorio de estilos 55 y la Vista general del diseño 45. Si algún aspecto de las hojas de estilos externas se puede ver en ambas ventanas (por ejemplo, la prioridad relativa de varias hojas de estilos), los cambios realizados en una de las dos ventanas se aplican automáticamente en la otra.

Añadir una hoja de estilos CSS externa al SPS

Siga estos pasos para añadir una hoja de estilos CSS externa al SPS:

1. En la vista **Diseño** seleccione el elemento `Externos` de la ventana Repositorio de estilos (*imagen siguiente*).



2. Haga clic en el botón **Añadir** situado en la parte superior izquierda de la barra de herramientas de la ventana (*imagen anterior*).
3. Se abre el cuadro de diálogo "Abrir". Busque y seleccione el archivo CSS necesario y haga clic en **Abrir**. El archivo CSS se incorpora al árbol **Externos** de la ventana Repositorio de estilos (*imagen siguiente*).
4. Para añadir otra hoja de estilos CSS externa repita los pasos 1, 2 y 3. La nueva hoja de estilos CSS se añade al árbol **Externos**, debajo de las hojas de estilos CSS añadidas previamente.

Nota: también puede añadir una hoja de estilos CSS externa desde la ventana Vista general del diseño 45.

Ver y modificar la estructura de hojas de estilos CSS externas

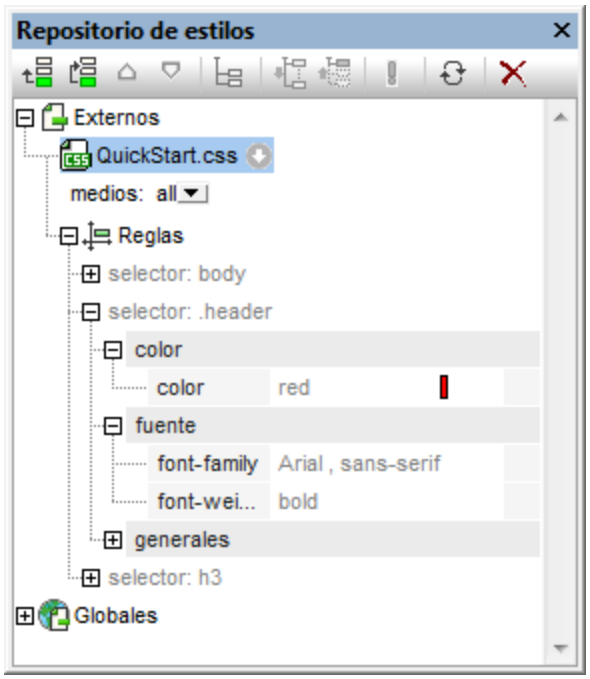
La estructura de hojas de estilos CSS externas es esta:

```
      - CSS-1.css  (La ubicación del archivo aparece al pasar el puntero del ratón por
encima)
        - Medios (se pueden definir en la ventana Repositorio de estilos)
        - Reglas (solo se pueden editar en el archivo CSS)
          - Selector-1
            - Propiedad-1
            - ...
            - Propiedad-N
          - ...
          - Selector-N
      + ...
      + CSS-N.css
```

Los medios a los que se aplica la hoja de estilos se pueden editar en la vista Repositorio de estilos. Haga clic en la flecha situada a la derecha del componente **medios** y seleccione un medio en la lista desplegable. Las reglas definidas en la hoja de estilos CSS externa aparecen en la ventana Repositorio de estilos, pero no se pueden editar. Los componentes **Hoja de estilos**, **Reglas** y **Selector** de la estructura de la ventana Repositorio de estilos se pueden expandir y contraer haciendo clic en los símbolos + y – situados a la izquierda de cada componente (*imagen siguiente*).

Para eliminar una hoja de estilos externa seleccione la hoja de estilos y haga clic en el botón **Restaurar** de la barra de herramientas de la ventana.

 © 2016-2022 Altova GmbH



Cambiar el orden de prioridad de las hojas de estilos CSS externas

Las hojas de estilos CSS externas que se asignan en la ventana Repositorio de estilos se importan al archivo HTML de salida por medio de la instrucción `@import`. Por ejemplo, en este archivo HTML se importaron 3 hojas de estilos:

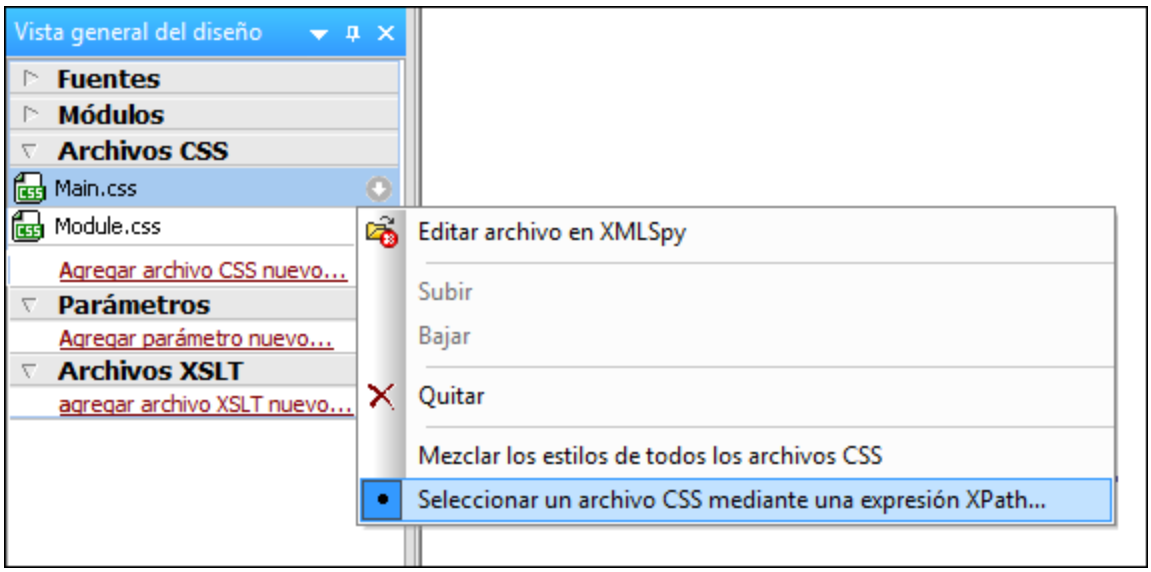
```
<html>
	<head>
		<style>
			<!--
			@import url("CSSExterna-1.css");
			@import url("CSSExterna-2.css")screen;
			@import url("CSSExterna-3.css")print;
			-->
		</style>
	</head>
	<body/>
</html>
```

El orden que tienen los archivos CSS en el archivo HTML (*ver ejemplo anterior*) es el orden que tienen las hojas de estilos en el árbol **Externos** de la ventana Repositorio de estilos y en la lista **Archivos CSS** de la ventana Vista general del diseño. Para cambiar el orden de las hojas de estilos CSS en el Repositorio de estilos seleccione una hoja de estilos y use los botones **Subir** o **Bajar** de la barra de herramientas para cambiar su posición relativa a las demás hojas de estilos. En la ventana Vista general del diseño haga clic en el botón en forma de flecha situado a la derecha de la hoja de estilos y seleccione el comando **Subir** o **Bajar**.

Importante: no olvide que la hoja de estilos con **mayor orden de prioridad** es la última hoja de estilos de la lista. Después viene la penúltima hoja de estilos y así sucesivamente. Por tanto, el orden de prioridad del ejemplo anterior sería: 1º `CSSExterna-3.css`; 2º `CSSExterna-2.css`; 3º `CSSExterna-1.css`. Cuando dos reglas CSS de hojas de estilos diferentes usan el mismo selector, se aplica la regla de la hoja de estilos con mayor orden de prioridad.

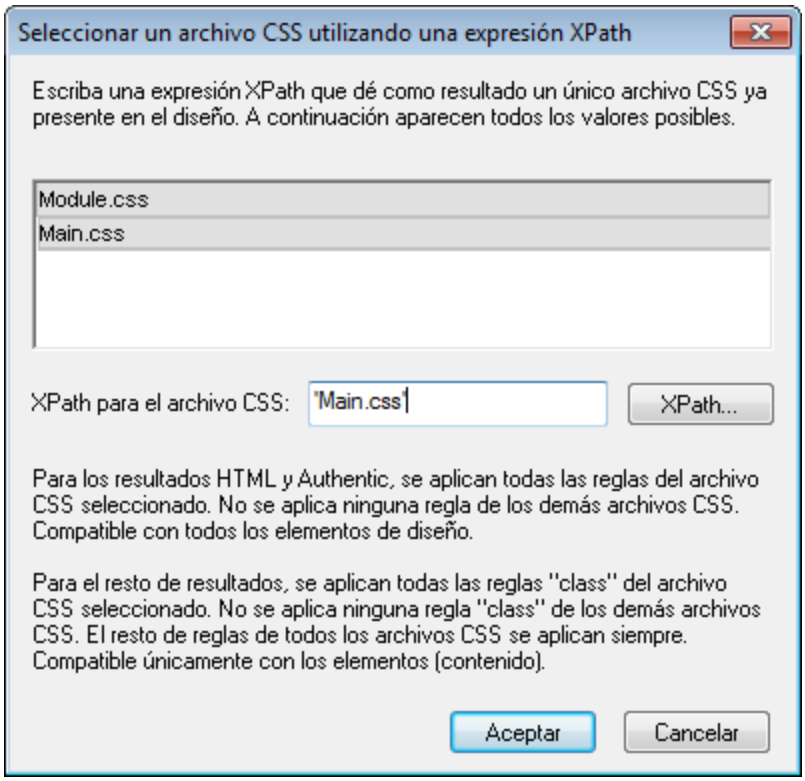
Alternar el conjunto de archivos CSS y un solo archivo CSS

En StyleVision tiene dos opciones: (i) aplicar las reglas de todos los archivos CSS y que su prioridad se determine según las reglas en cascada o (ii) aplicar las reglas de un solo archivo CSS. Puede elegir una de estas dos opciones en la ventana Vista general del diseño (*imagen siguiente*). Para ello haga clic en el botón en forma de flecha de un archivo CSS y seleccione el comando **Mezclar los estilos** o **Seleccionar un archivo CSS**. Estas opciones también están disponibles en la ventana Repositorio de estilos (en cualquier hoja de estilos externa).



Si hace clic en el comando **Seleccionar un archivo CSS mediante una expresión XPath**, se abre un cuadro de diálogo donde puede escribir la expresión XPath (*imagen siguiente*). La expresión XPath debe dar como resultado el nombre de uno de los archivos CSS del diseño SPS, tal y como aparecen en el panel superior del cuadro de diálogo. Si escribe el nombre de archivo como cadena de texto, recuerde que debe ir entre comillas simples (como todas las cadenas de texto de las expresiones XPath).

 © 2016-2022 Altova GmbH



Tenga en cuenta estos dos puntos:

- *Cuando está seleccionado un solo archivo CSS:* en los formatos de salida Authentic y HTML se aplican todas las reglas del archivo CSS seleccionado y estas reglas son compatibles con todos los componentes de diseño. En el formato de salida RTF solamente se aplican las reglas de selectores de clases del archivo CSS seleccionado. Después se aplican las reglas no de clase de todos los archivos CSS y, si hay conflictos, estos se resuelven según el orden de prioridad del archivo CSS. En el formato de salida RTF estas reglas se pueden aplicar a estos componentes de diseño: cálculos automáticos 293, el marcador de posición 135 `(contenido)` 135, los componentes de bloque (párrafos) 137 y las celdas de tablas 155.
- *Cuando se mezclan los estilos de todos los archivos CSS:* en los formatos de salida Authentic y HTML se aplican todas las reglas de todos los archivos CSS y son compatibles con todos los componentes de diseño. Si hay conflictos, estos se resuelven según el orden de prioridad del archivo CSS. En el formato de salida RTF solo se aplican las reglas de selectores de clase y, si hay conflictos, estos se resuelven según el orden de prioridad del archivo CSS.

8.4.2 Estilos globales

Los estilos globales del diseño SPS se definen en el Repositorio de estilos y se enumeran bajo el elemento **`Globales`**. Estos estilos se pasan a la vista Authentic y al documento HTML de salida como reglas CSS dentro del elemento `/html/head/style`.

En el Repositorio de estilos un *estilo global* es una regla CSS formada por un selector y propiedades CSS para el selector. Por tanto, el proceso de creación de estilos globales tiene dos fases:

- Añadir un estilo nuevo y declarar su selector CSS
- Definir propiedades CSS para el selector

Selectores compatibles

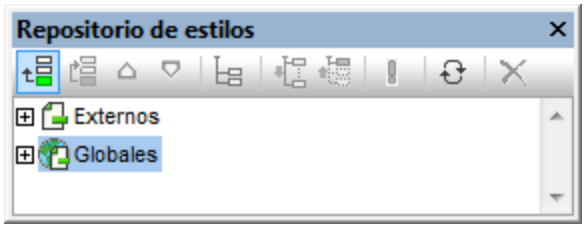
Estos son los selectores compatibles:

- *Selector universal:* escrito como `*`
- *Selectores de tipo:* nombres de elemento como `h1`
- *Selectores de atributo:* como `[class=maindoc]`
- *Selectores de clase:* por ejemplo `.maindoc`
- *Selectores de Id.:* por ejemplo `#header`

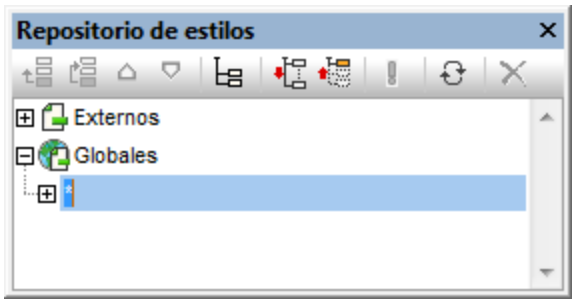
Añadir un estilo global

Siga estos pasos para añadir un estilo global al diseño SPS:

1. En la vista **Diseño** seleccione el componente **`Globales`** de la ventana Repositorio de estilos (*imagen siguiente*).



2. Haga clic en el botón **Añadir** situado en la parte superior izquierda de la barra de herramientas (*imagen anterior*). El estilo global se inserta en el árbol **`Globales`** con un selector * (que selecciona todos los elementos HTML). El selector universal * es el selector predeterminado cuando se insertan estilos globales nuevos.
3. Para cambiar de selector, haga doble clic en él y edítelo.



4. Ahora debe establecer los valores de propiedad CSS para el selector. Esto se explica en detalle en el apartado Establecer valores de estilos 400.
5. Para añadir otro estilo global repita los pasos 1, 2, 3 y 4. El nuevo estilo global se añadirá al árbol **`Globales`**, debajo de los estilos globales añadidos previamente.

Nota:

 © 2016-2022 Altova GmbH

- Para insertar estilos globales antes de un estilo global del árbol **Globales**, haga clic en el botón **Insertar** de la barra de herramientas del Repositorio de estilos. Los botones **Agregar** e **Insertar** también están disponibles en el menú contextual que aparece al hacer clic con el botón derecho en un selector global.
- Para insertar un estilo global con un selector que sea un elemento HTML haga clic con el botón derecho en un elemento del árbol **Globales** y seleccione el comando **Agregar selector | HTML |** ***NombreElementoHTML***.

Editar y eliminar estilos globales

Tanto el selector de estilo como sus propiedades se pueden editar en la ventana Repositorio de estilos.

- Para editar un selector haga doble clic en el nombre del selector, ponga el cursor en el campo de texto y escriba el nombre nuevo.
- Para aprender a definir y editar las propiedades de un estilo consulte el apartado Establecer valores de estilos 400. (Las propiedades de los estilos pueden verse de tres formas. Estas tres vistas se describen en el apartado Opciones de presentación de las definiciones 59.)

Para eliminar un estilo global, selecciónelo y haga clic en el botón **Restaurar** de la barra de herramientas del Repositorio de estilos.

Cambiar el orden de prioridad de los estilos globales

Los estilos globales que están asignados en la ventana Repositorio de estilos se crean como reglas CSS en el elemento `/html/head/style`. En el archivo HTML tendrían este aspecto:

```
<html>
	<head>
		<style>
			<!--
			h1		{ color:blue;
				  font-size:16pt;
				}
			h2		{ color:blue;
				  font-size:14pt;
				}
			.red	{ color:red;}
			.green	{ color:green;}
			.green	{ color:lime;}
			-->
		</style>
	</head>
	<body/>
</html>
```

El orden de los estilos globales en la vista Authentic y en el documento HTML equivale al orden de los estilos globales en el árbol **Globales** del Repositorio de estilos. El orden de los estilos globales en la vista Authentic y en el documento HTML es importante. Si dos selectores seleccionan el mismo nodo, tiene prioridad el selector que aparece más abajo en la lista de estilos globales. Por ejemplo, si en el documento HTML del ejemplo anterior hubiera un elemento `<h1 class="green">`, entonces 3 de los estilos globales apuntarían a este elemento: el estilo con el selector `h1` y los dos selectores de clase `.green`. Se aplicará la propiedad `color` del selector `.green` con el color `lime` porque aparece después del selector `.green` con color `green` y, por tanto, tiene prioridad. Los selectores de clase siempre tienen prioridad sobre los selectores de nodo así

que ambos selectores `.green` tendrán prioridad sobre el selector `h1`, independientemente de su posición en relación al selector `h1`. No obstante, el tamaño de fuente del estilo `h1` se aplicará al elemento `<h1>` porque no hay ningún selector con mayor prioridad que apunte al elemento `<h1>` y que tenga una propiedad `font-size`.

Para cambiar el orden de prioridad de un estilo global selecciónelo y use los botones **Subir** y **Bajar** de la barra de herramientas del Repositorio de estilos. Al hacer clic en estos botones cambia la posición del estilo global en relación a los demás estilos globales del árbol **`Globales`**. Por ejemplo, si movemos el estilo global `.green` antes del estilo `.red`, la propiedad `color` del estilo `.red` tendrá prioridad sobre la del estilo `.green`.

Sin embargo, no olvide que los selectores de clase siempre tienen prioridad sobre los selectores de tipo. Por tanto, si cambiamos el orden por `.red .green h1 h2`, entonces `h1` y `h2` siguen siendo verde (`green`).

8.4.3 Estilos locales

Cuando se definen estilos de forma local, las reglas de estilo se definen directamente en los componentes del diseño SPS. Estas reglas locales tienen prioridad sobre las reglas de estilo globales y sobre las reglas de estilo de las hojas de estilos CSS externas que seleccionan dicho componente. Los estilos definidos de forma local son estilos CSS y se definen desde la barra de herramientas Formato 750 o desde la ventana Estilos 58. (No se pueden definir en la ventana Repositorio de estilos 55 como los estilos globales.)

Definir estilos locales en la barra de herramientas Formato

En la vista **Diseño** puede seleccionar contenido y aplicarle estilos locales desde la barra de herramientas Formato (*imagen siguiente*).

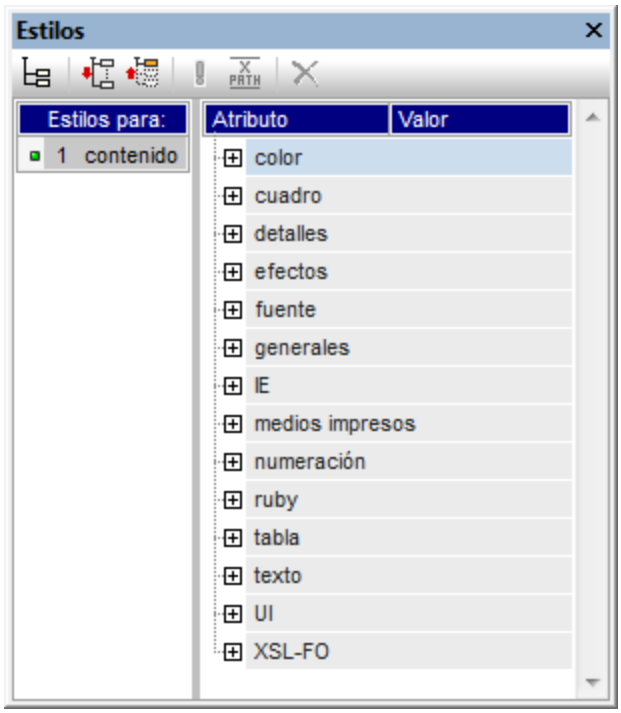
Desde esta barra de herramientas puede aplicar formatos HTML predefinidos (como `div`, `h1`, `pre`, etc.), estilo de texto, color de fondo, alineación de texto, listas e hipervínculos. Consulte el apartado Barra de herramientas Formato 750 para obtener más información.

Definir estilos locales en la ventana Estilos

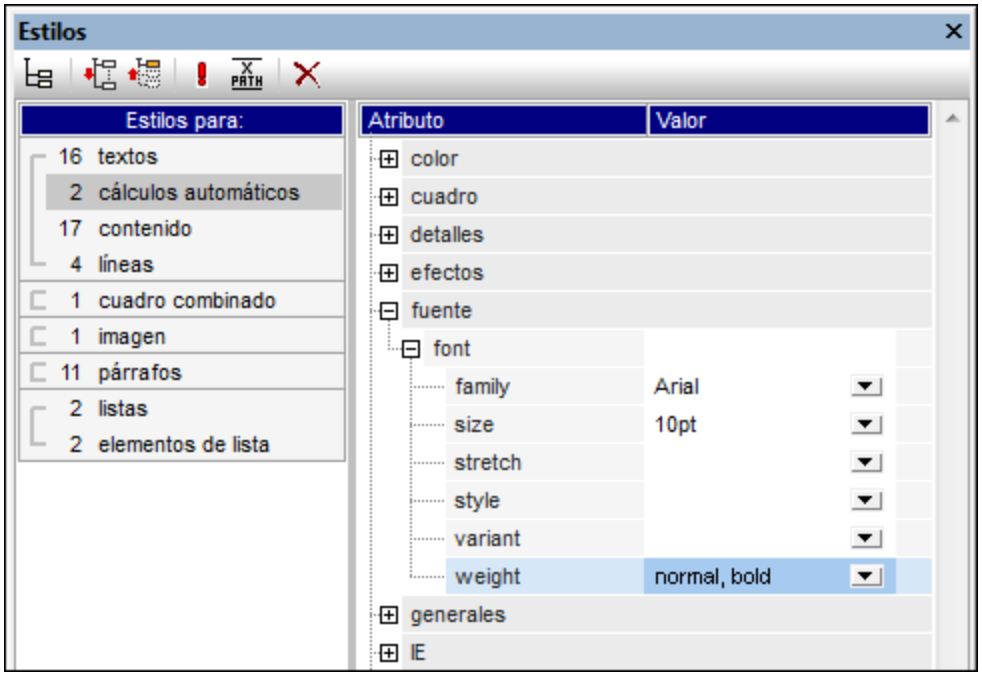
Para definir un estilo local desde la ventana Estilos son necesarios tres pasos:

1. Seleccione el componente en la vista **Diseño**. Puede aplicar estilos a cualquier componente del diseño excepto a las etiquetas de nodo. El componente seleccionado en la vista **Diseño** aparece en la columna *Estilos para:* de la ventana Estilos. En el ejemplo de la imagen siguiente se seleccionó un componente `(contenido)` en la vista **Diseño**.

 © 2016-2022 Altova GmbH



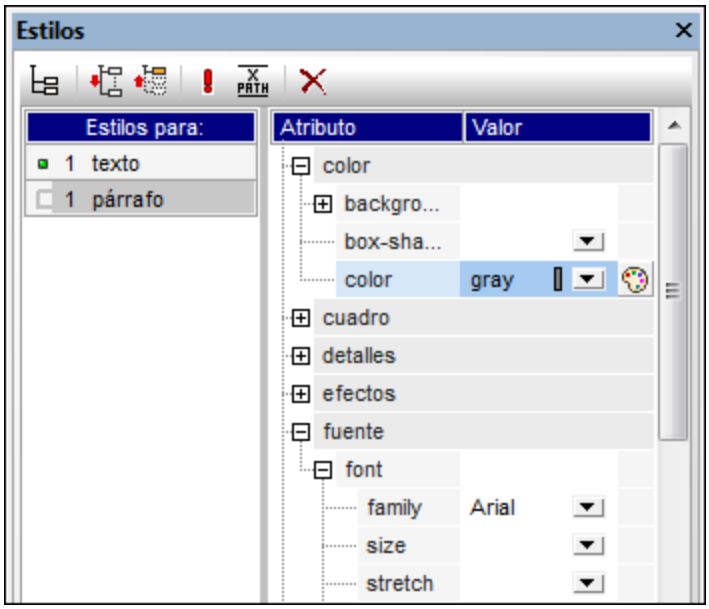
A menudo, el componente seleccionado en la vista **Diseño** incluye otros componentes. En casos así en la columna *Estilos para:* de la ventana Estilos aparecen todos los componentes de la selección. La imagen siguiente, por ejemplo, incluye todos los componentes que incluye la selección en la vista **Diseño**. A la izquierda de cada tipo de componente aparece el número de instancias del componente. Por ejemplo, en la imagen siguiente vemos que la selección de la vista **Diseño** contiene, entre otros, 16 componentes de texto y 2 cálculos automáticos. Para seleccionar un rango de componentes mantenga pulsada la tecla **Ctrl** mientras hace clic los componentes en la vista **Diseño**.



2. Ahora seleccione el tipo de componente al que quiere dar estilos en la columna *Estilos para:*. Si hay más de una instancia del tipo de componente, los estilos se aplican a todas sus instancias. Por ejemplo, si selecciona **`16 textos`**, los estilos que defina (*ver paso 3*) se aplican a los 16 componentes de texto seleccionados. Si quiere dar estilos diferentes a cuatro de estos componentes de texto, por ejemplo, entonces debe seleccionarlos por separado y darles estilos distintos. Si dos componentes del mismo tipo de componente tienen asignados estilos diferentes y ambos están seleccionados en la vista **Diseño**, los estilos de ambas instancias aparecen en la ventana Estilos. Por ejemplo, en la imagen anterior, un cálculo automático tiene `font-weight normal` y el otro cálculo automático tiene `font-weight bold`. Como ambos cálculos automáticos están seleccionados (**`2 cálculos automáticos`**) los dos valores de la propiedad `font-weight` aparecen en la ventana Estilos.
3. Tras seleccionar el tipo de componente en la columna *Estilos para:* defina los estilos en el panel derecho Definiciones de estilos 58. Consulte el apartado Establecer valores de estilos 400 para más información.

8.4.4 Establecer valores de estilos

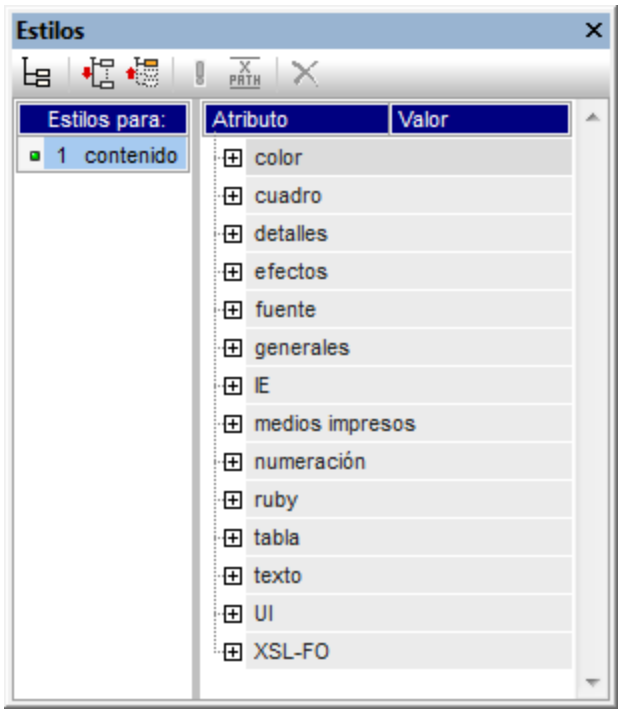
Tras seleccionar un tipo de componente en la columna *Estilos para:* de la ventana Estilos, las propiedades de los estilos se definen en el panel derecho Definiciones de estilos 58 (imagen *siguiente*). Si quiere, en la columna *Estilos para:* puede seleccionar más de un tipo de componente (mantenga pulsada la tecla **Ctrl** mientras hace clic en los componentes o mantenga pulsada la tecla **Mayús** para seleccionar un rango de componentes). Cuando se seleccionan varios componentes, los estilos definidos en el panel *Definiciones de estilos* se aplican a todas las instancias de todos los tipos de componente seleccionados.



Grupos de propiedades de estilo

Las propiedades de estilo disponibles en el panel *Definiciones de estilos* se dividen en varios grupos (*imagen siguiente*).

 © 2016-2022 Altova GmbH

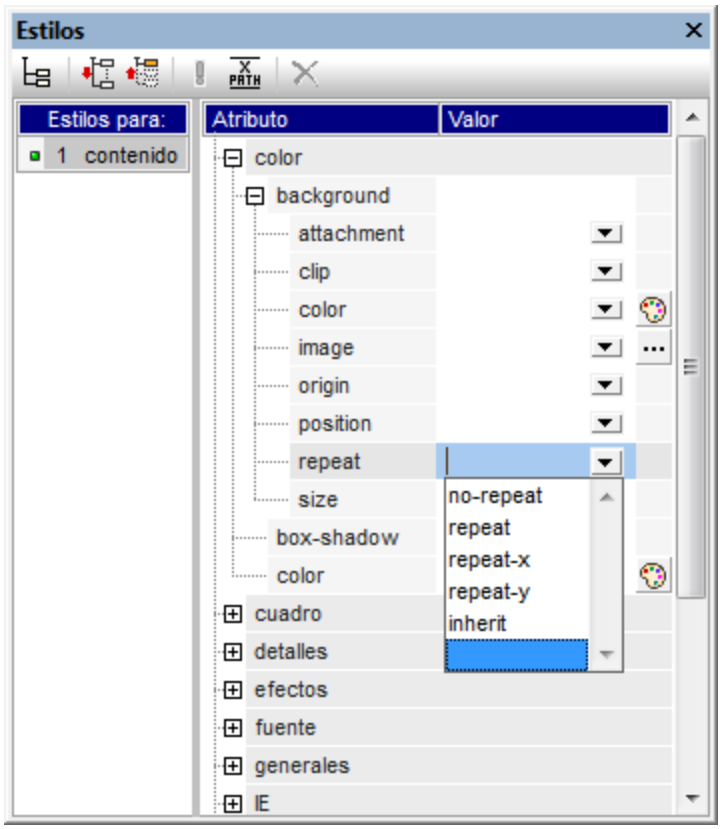


La ventana Estilos también incluye los prácticos botones **Mostrar elementos no vacíos**[58], [58]**Expandir todos**[58] y **Contraer todos**[58] con los que puede definir qué propiedades aparecen en la ventana.

Introducir valores de estilos

Los valores de las propiedades de estilo (valores de estilos) se pueden introducir de varias formas:

- **Directamente en la columna *Valor*:** seleccione una propiedad, haga doble clic en su campo *Valor* y teclee su valor. Para terminar haga clic en **Entrar** o haga clic en cualquier otra parte de la IGU.
- **Seleccionando un valor de la lista desplegable de la propiedad:** haga clic en la flecha del cuadro combinado para abrir la lista desplegable de opciones y seleccione una opción. En la imagen siguiente, por ejemplo, aparecen las opciones de la propiedad `(background-)repeat`.
- **Con el icono situado a la derecha del campo *Valor* de la propiedad.** Hay dos tipos de iconos, dependiendo del tipo de propiedad: (i) una paleta de colores para seleccionar colores (ver la propiedad `(background-)color` en la imagen siguiente) y (ii) un botón **Examinar** para buscar archivos (ver la propiedad `(background-)image` en la imagen siguiente).



- También puede asignar valores a los estilos mediante expresiones XPath 402.

Modificar o eliminar el valor de un estilo

Si introduce mal un valor de estilo o el valor no es válido, el valor aparece en color rojo. Para modificar un valor, utilice uno de los métodos descritos en el párrafo anterior (Introducir valores de estilos) 401.

Para eliminar un valor de un estilo (en otras palabras, para restaurar el valor) haga clic en el botón **Restaurar** de la barra de herramientas de la ventana Estilos. También puede hacer doble clic en el campo *Valor* de la propiedad, elimine el valor con la tecla **Suprimir** o **Retroceso** y después pulse **Entrar**.

8.4.5 Propiedades de los estilos mediante XPath

En StyleVision puede usar expresiones XPath para asignar estilos a los componentes del diseño SPS. Esto permite tomar los valores de las propiedades de estilo de los datos XML o de la expresión XPath misma. Además, gracias a la función `doc()` de XPath 2.0/3.0, puede apuntar también a los nodos de todos los documentos XML a los que tiene acceso. Gracias a esta característica de StyleVision no solo puede recuperar definiciones de estilo de los datos XML. También puede elegir estilos diferentes dependiendo de si se cumple o no una condición basada en la estructura o en el contenido de los datos XML. Por ejemplo, con la instrucción `if...else` de XPath 2.0/3.0, puede seleccionar dos colores de fondo distintos dependiendo de la posición de un elemento en la secuencia. Así, cuando estos elementos se presenten como filas de una tabla, las filas impares se pueden presentar con un color de fondo y las filas pares con otro (*imagen siguiente*). Además, dependiendo del contenido del nodo puede variar la presentación de los datos.

 © 2016-2022 Altova GmbH

Propiedades de estilo compatibles con expresiones XPath

Puede introducir expresiones XPath para estas propiedades de estilo:

- Todas las propiedades que aparecen en la ventana Estilos.
- Los grupos de propiedades *generales*, *evento* y *HTML* de la ventana Propiedades.

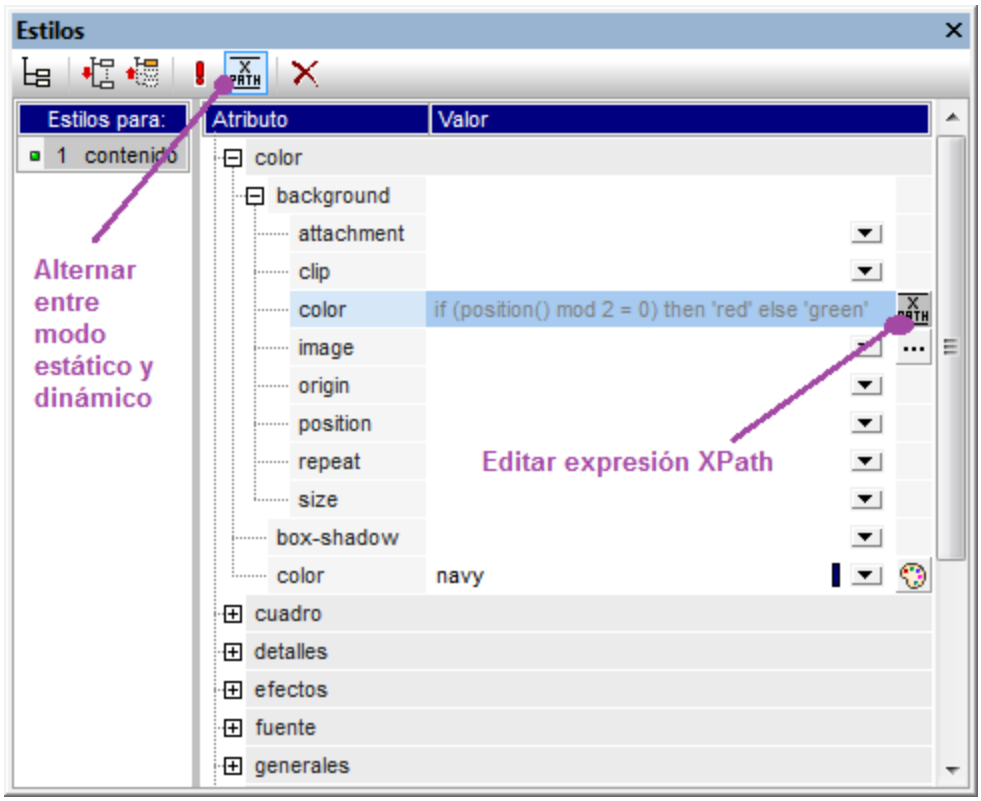
Modo estático o dinámico (XPath) para los valores de las propiedades de estilo

Las propiedades que admiten expresiones XPath 402 tienen dos modos:

- **Modo estático:** el valor de la propiedad se escribe directamente en el campo *Valor* de la ventana. Por ejemplo, para el color de fondo de un componente de diseño, puede introducir el valor `red` en la ventana directamente.
- **Modo dinámico (o modo XPath):** el valor de la propiedad se calcula con una expresión XPath. La expresión se evalúa en tiempo de ejecución y el resultado se introduce como valor de la propiedad. Por ejemplo, para el color de fondo de un componente de diseño, puede introducir esta expresión XPath: `/root/colors/color1`. En tiempo de ejecución se recupera el contenido del nodo `/root/colors/color1` y se introduce como valor de la propiedad `background-color`.

Cambiar entre el modo estático y el modo dinámico (XPath)

El modo estático es el modo predeterminado de las propiedades compatibles con expresiones XPath. Para cambiar al modo dinámico, seleccione la propiedad y haga clic en el icono **XPath** de la barra de herramientas de la ventana (*imagen siguiente*).



Si la propiedad ya tenía un valor estático, el valor se borra y se cambia al modo dinámico. Aparece el cuadro de diálogo Editar expresión XPath 729. La expresión XPath se introduce en este cuadro de diálogo. Cuando termine haga clic en **Aceptar**.

Tras introducir la expresión XPath de la propiedad, aparece el botón **Editar XPath** junto al campo *Valor* de la propiedad (*imagen siguiente*). Haga clic en este botón para editar otra vez la expresión XPath. Si quiere volver al modo estático, haga clic en el icono **XPath** de la barra de herramientas de la ventana. Al cambiar al modo estático se borra la expresión XPath.

Nota: hay dos puntos importantes que debe tener en cuenta. Primero: solamente puede haber un modo activo y un valor/una expresión en cada modo. El valor/la expresión que está activo/a se borra cuando se cambia de modo. Es decir, al cambiar de modo el campo *Valor* se queda vacío. Para recuperar la expresión/el valor anterior, use el comando **Deshacer** 794. Segundo: si vuelve a seleccionar una propiedad tras editar el SPS, la propiedad se abre en el modo en que estaba la última vez que se editó.

Crear y editar la definición XPath

La definición XPath se crea y edita en el cuadro de diálogo Editar expresión XPath 729. Hay dos maneras de abrir este cuadro de diálogo:

- El cuadro de diálogo Editar expresión XPath 729 se abre cuando se cambia del modo estático de una propiedad al modo dinámico (al hacer clic en el botón **XPath** de la barra de herramientas de la ventana). Si ya está en el modo dinámico y hace clic en el botón **XPath** de la barra de herramientas, se cambia al modo estático pero el cuadro de diálogo "Editar expresión XPath" no se abre.
- El cuadro de diálogo Editar expresión XPath 729 se abre al hacer clic en el botón **Editar expresión XPath** del campo *Valor* de una propiedad que ya tiene definida una expresión XPath. El diálogo tendrá la expresión XPath de esa propiedad.

Tras editar la expresión XPath o introducir una nueva, haga clic en **Aceptar** para terminar.

Valores devueltos por las expresiones XPath

Las principales ventajas de usar expresiones XPath para establecer el valor de una propiedad son: (i) que el valor de la propiedad puede tomarse de un archivo XML (en lugar de introducirlo directamente) y (ii) que la expresión XPath puede probar si se cumple cierta condición relacionada con el contenido o la estructura del documento XML que se está procesando y, según el resultado de la prueba, la expresión XPath selecciona un valor. Las expresiones XPath devuelven valores de dos tipos:

- *Contenido de nodos XML*
  La expresión XPath puede apuntar a nodos (i) del documento XML que procesa el SPS o (ii) de cualquier documento XML al que se tenga acceso. Por ejemplo, la expresión `Format/@color` accede al atributo `color` del secundario `Format` del nodo de contexto. El valor del atributo `color` se usará como valor de la propiedad en la que se definió la expresión XPath. Para acceder a un nodo de otro documento XML debe usar la función `doc()` de XPath 2.0. Por ejemplo, la expresión `doc('Styles.xml')//colors/color-3` recupera el valor del elemento `color-3` del archivo XML `Styles.xml` y usa su valor para la propiedad en la que se definió la expresión XPath.

- *Expresión XPath*
  El valor de la propiedad puede venir dada por la expresión XPath propiamente dicha en lugar de venir del documento XML. Por ejemplo, el color de fondo de un elemento que se representará como una fila en el documento de salida puede alternarse dependiendo de la posición de la fila, de si es par o impar. Esto se consigue con la expresión XPath 2.0/3.0 `if (position() mod 2 = 0) then 'red' else`

 © 2016-2022 Altova GmbH

`'green'`. Observe que el valor devuelto de esta expresión es la cadena `red` o la cadena `green` y que se establecerá como valor de la propiedad en la que se definió la expresión XPath. En el ejemplo que acabamos de dar, los valores de la propiedad se introdujeron como literales de cadena. Pero también pueden venir de un documento XML. Por ejemplo: `if (position() mod 2 = 0) then doc('Styles.xml')//colors/color-1 else doc('Styles.xml')//colors/color-2`. En cambio, la expresión XPath podría ser una simple cadena como `'green'`. Pero esto sería como escribir el valor estático `green` en la propiedad.

8.4.6 Estilos compuestos

Un estilo compuesto es un grupo de propiedades de estilo de texto CSS asociadas a un atributo de un nodo del documento XML. Además se considera un estilo compuesto cualquier grupo de propiedades de estilo de texto CSS almacenadas en la hoja de estilos. Los estilos compuestos se pueden aplicar a estos componentes de diseño:

- Cálculos automáticos 293
- Marcador de posición (contenido) 135
- Elementos de diseño de bloque (párrafos) 137
- Celdas de tablas 155

Ventajas de los estilos compuestos

Las ventajas de los estilos compuestos son:

- Las propiedades de estilo están en los datos XML y por tanto el usuario puede editarlas. En la vista Authentic, la característica RichEdit permite editar los estilos compuestos de forma gráfica con ayuda de una barra de herramientas. Para más información consulte el apartado Estilos compuestos en Authentic 408.
- Se pueden combinar las propiedades almacenadas en los datos XML con las propiedades asignadas en el diseño SPS y asignarlas a los componentes de diseño mencionados antes.
- Mientras diseña el documento SPS, el diseñador puede cambiar rápidamente los estilos compuestos asociados con un elemento.

Introducir el estilo compuesto en el atributo XML

Un estilo compuesto (formado por varias propiedades de estilo) se introduce como valor de atributo de un elemento en el documento XML de origen. Por ejemplo, el atributo `desc-style` del documento XML de ejemplo que aparece a continuación tiene un estilo compuesto:

```
<Desc desc-style="font-family:Verdana; font-size:12pt; color:blue">
```

En un elemento puede establecer más de un estilo compuesto. De ser así, cada estilo compuesto debe introducirse en un atributo distinto:

```
<Desc styleBlue="font-family:Verdana; font-size:12pt; color:blue"
      styleRed ="font-family:Verdana; font-size:12pt; color:red">
```

Cuando en un elemento hay varios estilos compuestos, puede alternar los estilos compuestos del elemento cambiando el valor de la propiedad `estilo compuesto` del componente de diseño (*imagen siguiente*).

Nota: los atributos que se usan para acceder a los estilos compuestos deben definirse en el esquema fuente o de lo contrario el documento XML no será válido.

Propiedades de estilo de texto CSS compatibles

Estos son los estilos CSS que se pueden usar en los estilos compuestos:

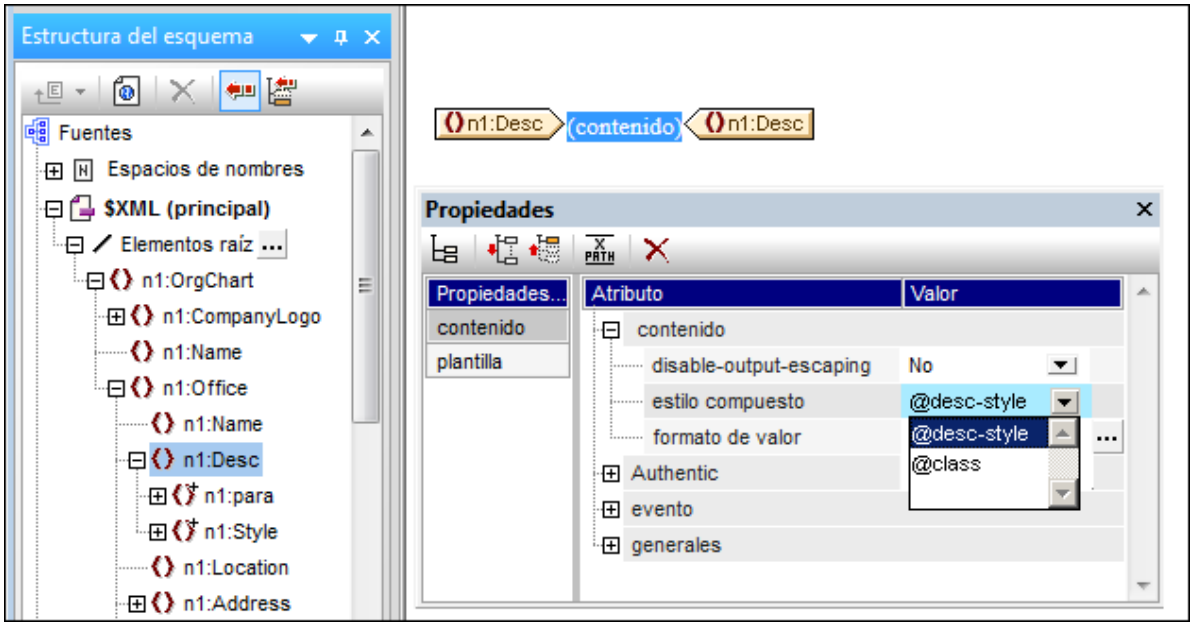
`font-family`	`font-size`	`font-weight`	`font-style`
`color`	`background-color`	`text-align`	`text-decoration`

Establecer un atributo como valor de estilo compuesto

Si establece el estilo compuesto de un componente de diseño como atributo, el usuario de la vista Authentic podrá editar este estilo compuesto. El usuario de la vista Authentic puede poner el cursor en cualquier parte del resultado de texto del componente de diseño y usar la barra de herramientas RichEdit de la vista Authentic para editar el estilo compuesto del componente de diseño.

Siga estos pasos para definir un atributo como estilo compuesto de un componente de diseño:

1. En la vista **Diseño** seleccione el componente de diseño al que quiere asignar un atributo como estilo compuesto. En la imagen siguiente, por ejemplo, está seleccionado el marcador de posición `(contenido)` del elemento `Desc`.



2. En el cuadro combinado de la propiedad `estilo compuesto` del componente **`contenido`** (ventana Propiedades) (*imagen anterior*), aparecen los atributos del elemento de contexto. Seleccione el atributo que desea usar como estilo compuesto del componente de diseño. (No olvide que también puede dejar el campo vacío si prefiere no aplicar ningún estilo compuesto. En ese caso la característica RichEdit de la vista Authentic no se habilitará para el componente de diseño.)

© 2016-2022 Altova GmbH

En la vista Authentic el usuario puede usar la barra de herramientas RichEdit para modificar el estilo compuesto de este componente de diseño.

Establecer una expresión XPath como valor de estilo compuesto

También puede usar una expresión XPath como valor de la propiedad `estilo compuesto`. En este caso, sin embargo, como el estilo compuesto se almacena en el SPS (y no en el documento XML), el estilo compuesto no se podrá editar en la vista Authentic.

Para establecer una expresión XPath como valor de la propiedad `estilo compuesto`, haga clic en el icono **XPath** de la barra de herramientas de la ventana XPath y, en el cuadro de diálogo que aparece, escriba la expresión XPath correspondiente. La expresión XPath se evaluará como plantilla de valor del atributo. El valor devuelto será el valor de un atributo HTML `style` (y de su equivalente en los formatos de salida no HTML).

Por ejemplo, imagine que creamos la siguiente expresión XPath en el marcador de posición `(contenido)` del elemento `n1:Person`.

```
if (number(n1:Shares) gt 1000) then 'color:red' else 'color:green'
```

Lo que hace esta expresión es: si el elemento `n1:Person` tiene un elemento secundario `n1:Shares` con un valor numérico mayor que 1000, entonces el contenido del elemento `n1:Person` se genera en color rojo. De lo contrario, todos los elementos `n1:Person` se generan en color verde. El valor devuelto por la expresión XPath se pasa al documento de salida como valor del atributo HTML `style` (o de su equivalente en los formatos de salida no HTML).

En la hoja de estilos XSLT generada a partir del SPS, esta expresión XPath se evaluará como plantilla de valor del atributo:

```
<span style="{if (number(n1:Shares) gt 1000) then &apos;color:red&apos; else
&apos;color:green&apos;}">
```

En el documento de salida HTML, se generará una de estas líneas, dependiendo de si se cumple o no la condición:

```
<span style="color:red">
```

o

```
<span style="color:green">
```

Nota: las plantillas de valor de atributo son construcciones XSLT que permiten leer el valor de un atributo como una expresión XPath. Estas plantillas están delimitadas por llaves y permiten asignar el valor del atributo de forma dinámica.

8.5 Flexibilidad de estilos en Authentic

Puede diseñar un SPS para que el usuario de la vista Authentic pueda aplicar estilos al texto (seleccionando un fragmento de texto y definiendo sus propiedades de estilo). Estas propiedades de estilo pueden estar predefinidas por el diseñador del SPS o pueden ser definidas por el usuario de la vista Authentic. En ambos casos el usuario de la vista Authentic puede tener la opción de dar estilos al texto.

Estas son las opciones de estilo de texto disponibles:

- Estilos compuestos 408: conjunto de propiedades de estilos CSS definidos en el atributo de un elemento del documento XML. Al definir este atributo como valor de la propiedad `estilo compuesto` de un componente de diseño, el estilo compuesto se puede editar con la característica RichEdit 410 de la vista Authentic. Consulte el apartado Estilos compuestos 408 para ver cómo funciona esta característica.
- RichEdit 410: si en el diseño SPS se crea un componente RichEdit a partir de un elemento, el usuario de la vista Authentic podrá seleccionar los fragmentos de texto del elemento y darles estilo con ayuda de la característica RichEdit de la vista Authentic. Gracias a esta característica, el usuario de la vista Authentic puede especificar la fuente del texto, su peso, estilo, decoración, tamaño, color, color de fondo y alineación del texto. Consulte el apartado RichEdit 410 para obtener más información.
- Iconos de estado del texto 414: el diseñador del SPS puede crear un botón en la barra de herramientas de la vista Authentic y asociar el botón a un nombre de elemento. Estos botones se denominan *iconos de estado del texto.* Cuando el usuario de la vista Authentic selecciona un fragmento de texto en la vista Authentic y hace clic en el icono de estado del texto, el elemento asociado con el icono se aplica alrededor del texto seleccionado (el botón solamente se habilita si el esquema permite crear este elemento alrededor del texto seleccionado). Como consecuencia el formato de texto definido para este elemento (en una plantilla global del elemento) se aplica al fragmento de texto seleccionado.

Puede usar todos estos métodos para dar estilo al texto en su SPS.

En cualquier caso debe tener en cuenta que:

- estos tres métodos exigen que los elementos / atributos de estilo estén presentes en el documento XML. Como consecuencia, el esquema en el que está basado el SPS debe permitir el uso de los elementos y atributos en los niveles jerárquicos en los que se necesitan.
- los estilos de texto aplicados por el usuario de la vista Authentic aparecerán tanto en la vista Authentic como en los demás formatos de salida.

Nota: los tres mecanismos mencionadas en este apartado se derivan de propiedades de estilo almacenadas en el archivo XML y se suman a las propiedades de estilo asignadas en el diseño SPS.

8.5.1 Estilos compuestos

Los estilos compuestos permiten al usuario de la vista Authentic aplicar estilos al resultado de estos componentes de diseño:

- Cálculos automáticos 293
- Marcador de posición (contenido) 135
- Elementos de diseño de bloque (párrafos) 137

 © 2016-2022 Altova GmbH

- Celdas de tablas 155

Para ver una descripción general de esta característica consulte el apartado Estilos compuestos 405. Este apartado describe cómo configurar estilos compuestos para que el usuario de la vista Authentic pueda editarlos.

Introducir el estilo compuesto en el atributo XML

Un estilo compuesto predeterminado (formado por varias propiedades de estilo) se introduce como valor de atributo de un elemento en el documento XML de origen. Por ejemplo, el atributo `desc-style` del documento XML de ejemplo que aparece a continuación tiene un estilo compuesto:

```
<Desc desc-style="font-family:Verdana; font-size:12pt; color:blue">
```

El diseñador del SPS ahora puede dar al usuario de la vista Authentic la opción de editar estas propiedades de estilo. Es decir, el usuario de la vista Authentic puede controlar el estilo del texto del componente de diseño. Para ello el diseñador del SPS debe definir el atributo que contiene el estilo compuesto predeterminado como estilo compuesto del componente de diseño (*imagen siguiente*).

Nota: los atributos utilizados para acceder a los estilos compuestos deben estar definidos en el esquema fuente o de lo contrario el documento XML no será válido.

Propiedades de estilo de texto CSS compatibles

Estos son los estilos CSS que se pueden usar en los estilos compuestos:

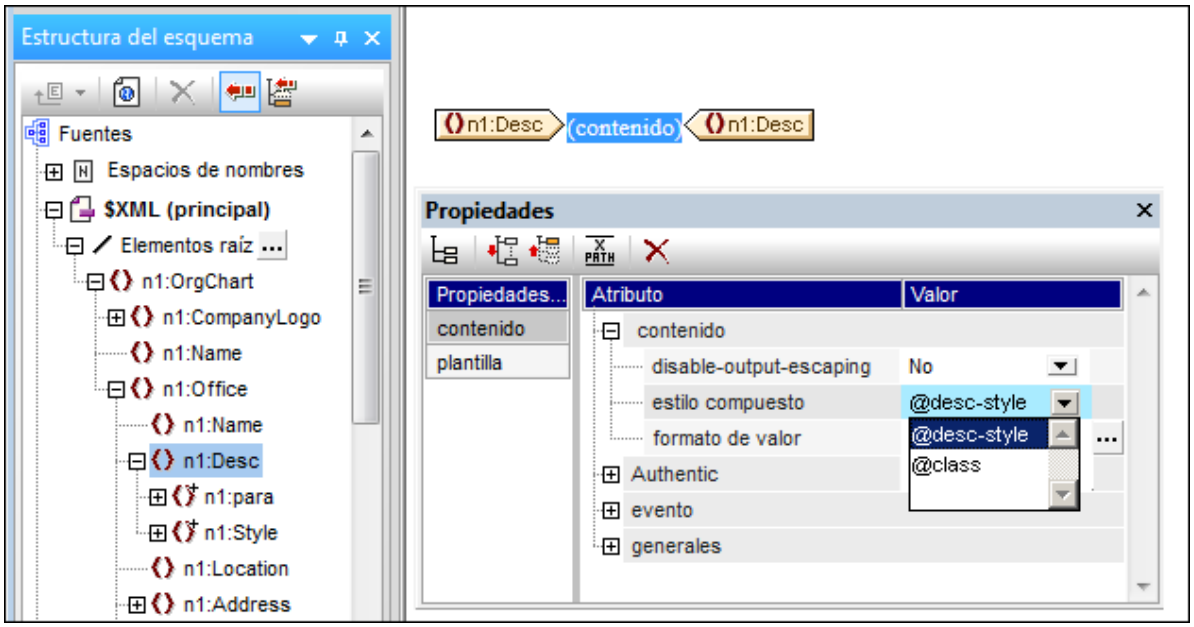
font-family	font-size	font-weight	font-style
color	background-color	text-align	text-decoration

Establecer el atributo como estilo compuesto de un componente de diseño

Si define un atributo como estilo compuesto de un componente de diseño, el usuario de la vista Authentic podrá editar este componente de diseño. Esto significa que el usuario de la vista Authentic puede poner el cursor en cualquier parte del resultado de texto del componente de diseño y usar la barra de herramientas RichEdit de la vista Authentic para editar el estilo compuesto del componente de diseño.

Siga estos pasos para definir un atributo como estilo compuesto de un componente de diseño:

1. En la vista **Diseño** del archivo SPS seleccione el componente de diseño al que quiere asignar un atributo como estilo compuesto. En la imagen siguiente, por ejemplo, está seleccionado el marcador de posición `(contenido)` del elemento `Desc`.



2. En el cuadro combinado de la propiedad `estilo compuesto` del componente **`contenido`** (ventana Propiedades) (*imagen anterior*), aparecen los atributos del elemento de contexto. Seleccione el atributo que desea usar como estilo compuesto del componente de diseño. (No olvide que también puede dejar el campo vacío si prefiere no aplicar ningún estilo compuesto. En ese caso la característica RichEdit de la vista Authentic no se habilitará para el componente de diseño.)

En la vista Authentic el usuario puede servirse de la barra de herramientas RichEdit para modificar el estilo compuesto de este componente de diseño.

8.5.2 RichEdit

Si en el diseño SPS crea un componente RichEdit a partir de un elemento, el usuario de la vista Authentic podrá marcar fragmentos de texto del elemento y darle estilos gracias a las propiedades de estilo RichEdit de la vista Authentic, así como definir el nivel de párrafo y la alineación del texto del fragmento.

Gracias a la característica RichEdit el usuario de la vista Authentic puede especificar:

- *estilos de caracteres:* la fuente, su peso, estilo, decoración, tamaño, color y color de fondo.
- *estilos de párrafos:* la alineación del texto.

Este apartado describe la característica RichEdit y se divide en tres partes:

- Cómo funciona la característica RichEdit 410
- Cómo crear un componente RichEdit a partir de un elemento en el diseño SPS 411
- Cómo usar la característica RichEdit en la vista Authentic 413

Funcionamiento de la característica RichEdit

Cuando el usuario de la vista Authentic selecciona un fragmento de texto y le aplica estilos RichEdit, el elemento de estilo RichEdit y el atributo que almacena la información sobre los estilos se crea alrededor del

fragmento de texto seleccionado. Las propiedades de estilo RichEdit seleccionadas por el usuario de la vista Authentic se insertan como valor de este atributo.

Es decir, si en el documento XML de origen hay este fragmento de texto:

```
<p> ... Altova StyleVision 2012 features a unique graphical design interface ...
</p>
```

y a parte de este fragmento se le da la propiedad RichEdit **bold** (negrita), entonces el fragmento de texto del XML será:

```
<p> ... <span style="font-weight: bold">Altova StyleVision 2012</span> features a
unique graphical design interface ... </p>
```

El elemento de estilo RichEdit del ejemplo anterior es el elemento **`span`** y su atributo que debe incluir las propiedades de estilo RichEdit es el atributo **`style`**. Pero puede cambiar el nombre de este elemento y de este atributo si quiere. Por ejemplo, al elemento le puede llamar `Style` y al atributo le puede llamar `css`:

```
<p> ... <Style css="font-weight: bold">Altova StyleVision 2012</Style> features a
unique graphical design interface ... </p>
```

Sea cual sea el nombre que elija para el elemento y el atributo, lo más importante es que **estén definidos en el esquema y que el esquema permita que aparezcan dentro del elemento** que contiene el texto al que se le aplicó el estilo RichEdit.

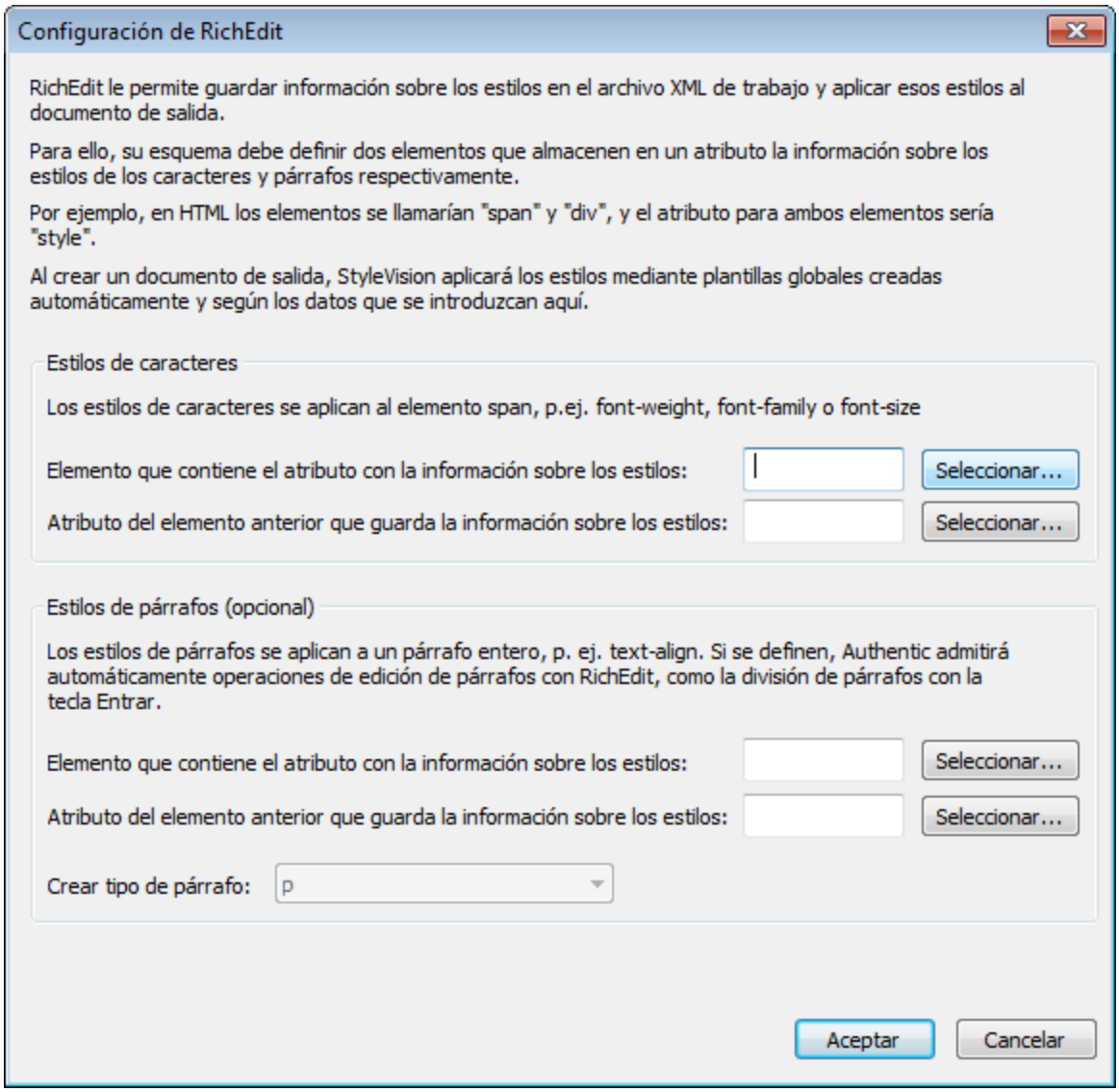
Cuando el texto se procesa con una hoja de estilos XSLT, las propiedades de estilo se pasan al resultado con el marcado propio del formato de salida.

Gracias a la característica RichEdit el usuario de la vista Authentic también puede aplicar formato de bloque (como alineación del texto). Puede seleccionar el elemento y el atributo que contendrá el formato de bloque. Algunas propiedades de formato de bloque, como la alineación del texto, están disponibles en la vista Authentic cuando el usuario edita un elemento que contiene el elemento que se definió para estilos de párrafo en el cuadro de diálogo "Configuración de RichEdit". Al igual que ocurre con los estilos de caracteres, el esquema debe permitir que el elemento de estilo de párrafo RichEdit aparezca dentro del elemento al que quiere aplicar este estilo.

Crear un componente RichEdit a partir de un elemento

Para crear un componente RichEdit a partir de un elemento:

1. Arrastre el elemento desde la Estructura del esquema y suéltelo en la posición deseada del diseño.
2. Seleccione el comando **Crear RichEdit** en el menú contextual que aparece. Se abre el cuadro de diálogo "Configuración de RichEdit" (*imagen siguiente*). Si ya creó un estilo de caracteres RichEdit para el documento, haga clic con el botón derecho en el componente RichEdit en la vista **Diseño** y seleccione el comando **Configurar elementos o atributos de RichEdit**.



3. En el cuadro de diálogo "Configuración de RichEdit", en el panel *Estilos de caracteres*, escriba el nombre del elemento de estilo y el atributo que debe contener las propiedades de estilo RichEdit para los fragmentos de texto. También puede seleccionar el elemento y el atributo en el árbol del esquema (haciendo clic en los botones **Seleccionar**).
4. Para habilitar el formato de bloque (alineación de texto), en el panel *Estilos de párrafos*, escriba el nombre del elemento de estilo y su atributo que debe contener el formato de bloque. En el cuadro combinado *Crear tipo de párrafo* puede seleccionar el formato predefinido del párrafo. Este formato predefinido se enviará a los documentos de salida.
5. Cuando termine haga clic en **Aceptar**. A partir del elemento se crea un componente RichEdit (*imagen siguiente*) y en el diseño se crea une plantilla global RichEdit **no editable** para estilos de caracteres que se llama igual que el elemento de estilo (**Style** en el ejemplo de la imagen siguiente). Si también especificó estilos de párrafos, entonces se crea también una plantilla global RichEdit no editable para los estilos de párrafos.

 © 2016-2022 Altova GmbH

Es importante tener en cuenta que:

- La plantilla global RichEdit no editable se crea cuando se crea el primer componente RichEdit en el diseño SPS. A partir de ese momento ya no se crean más plantillas globales RichEdit.
- Si después del primer componente RichEdit se crean más componentes RichEdit a partir de elementos, el cuadro de diálogo "Configuración RichEdit" no aparece. Los componentes RichEdit se identifican con un cuadro de texto llamado **`(richedit)`** (*imagen anterior*). Este cuadro de texto **`(richedit)`** aparece en vez del marcador de posición `(contenido)` habitual.
- La plantilla global RichEdit se puede volver a configurar. Es decir, puede cambiar el elemento y atributo para estilos de caracteres y estilos de párrafos. Para ello haga clic con el botón derecho en el componente RichEdit, seleccione el comando **Configurar elementos o atributos de RichEdit** y escriba el nuevo elemento y el nuevo atributo. El nombre de los nuevos elementos (para estilos de caracteres y estilos de párrafos) aparecerán en la barra de título de las plantillas globales RichEdit.

A partir de este momento el usuario de la vista Authentic puede aplicar propiedades de estilo RichEdit a los elementos de esquema con los que se crearon los componentes RichEdit en la vista **Diseño** del SPS.

Usar la característica RichEdit en la vista Authentic

Cuando el usuario de la vista Authentic pone el cursor dentro de un elemento con el que se creó un componente RichEdit, se habilitan los botones y controles de la barra de herramientas RichEdit (*imagen siguiente*). De lo contrario, a barra de herramientas aparece atenuada en gris.

Para aplicar estilos de caracteres el usuario de la vista Authentic debe seleccionar el texto al que desea aplicar estilos. Para aplicar estilos de párrafos el usuario debe poner el cursor dentro del párrafo al que desea aplicar estilos. Después debe especificar el estilo con los botones y controles de la barra de herramientas RichEdit. Si el texto seleccionado no está todavía dentro de las etiquetas del elemento de estilo, se incluirá ahora.

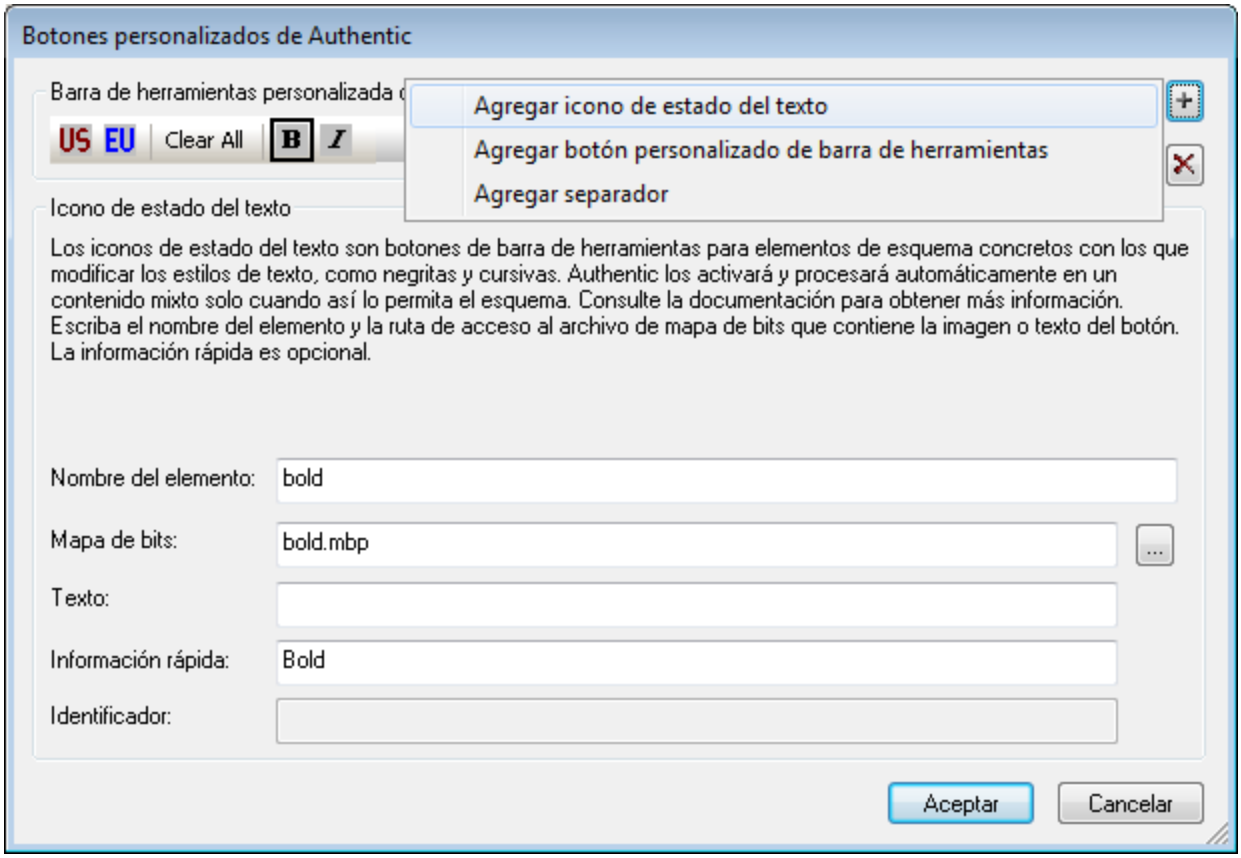
8.5.3 Iconos de estado del texto

Un icono de estado del texto es un botón de la barra de herramientas de la vista Authentic que está asociado a un nombre de elemento. Cuando el usuario de la vista Authentic hace clic en un icono de estado del texto, el fragmento de texto seleccionado se incluye dentro del elemento asociado al icono. El formato de texto definido para este elemento (en la plantilla global del elemento) se aplica al fragmento de texto seleccionado. Por tanto, el usuario de la vista Authentic puede aplicar estilos de texto con estos iconos.

El diseñador del SPS deberá crear los iconos de estado del texto y definir los estilos de los elementos asociados a los iconos.

Crear un icono de estado del texto

Para crear un icono de estado del texto para la barra de herramientas Authentic, en la vista **Diseño** seleccione el comando **Authentic | Personalizar botones de la barra de herramientas**. Se abre el cuadro de diálogo "Botones personalizados de Authentic" (*imagen siguiente*). Haga clic en el botón **Agregar** (en forma de **+**) en la parte superior derecha del cuadro de diálogo y seleccione el comando **Agregar icono de estado del texto**.



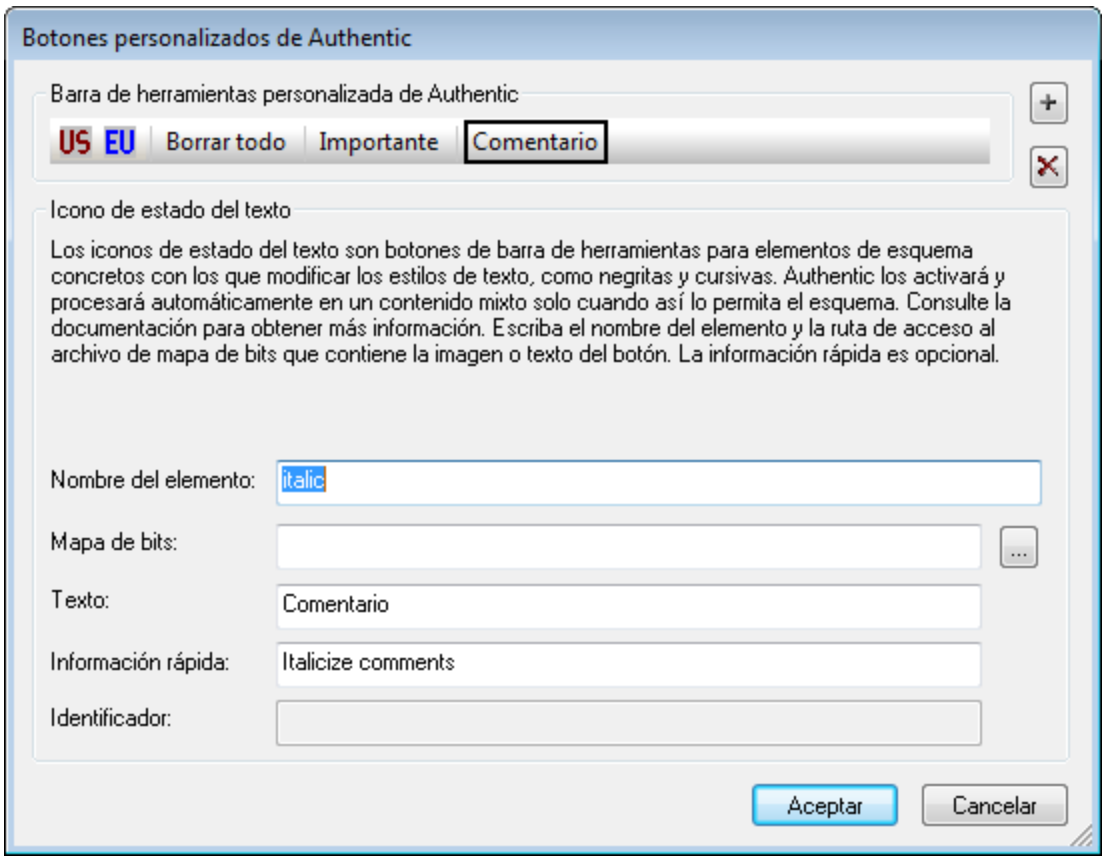
Los iconos de estado del texto toman estos parámetros:

- *Nombre del elemento:* este es el elemento en el que se incluirá el fragmento de texto seleccionado cuando el usuario de la vista Authentic hace clic en el icono de estado del texto.
- *Mapa de bits:* ubicación de la imagen para el icono de estado del texto. La ruta de acceso del archivo es relativa al SPS.

 © 2016-2022 Altova GmbH

- *Texto:* si no hay un mapa de bits, entonces se usa el texto de este campo como texto del icono.
- *Información rápida:* información adicional para el usuario de la vista Authentic que aparece la pasar el cursor del ratón por encima del icono.

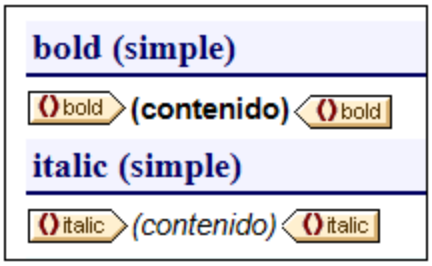
En el ejemplo anterior usamos imágenes comunes para los iconos de estado del texto para macar el texto con los elementos `bold` e `italic`. Pero puede usar cualquier otra imagen con la que esté familiarizado el usuario de la vista Authentic. Por ejemplo, los fragmentos de texto importantes pueden asociarse al elemento `bold`, mientras que los comentarios se pueden asociar al elemento `italic`. En este caso podría crear los iconos de estado del texto que aparecen en la siguiente imagen.



Observe que el icono **Comentario** está asociado al elemento `italic`. De este modo, cuando el usuario de la vista Authentic selecciona texto y después hace clic en el icono **Comentario**, el fragmento de texto se incluye dentro del elemento `italic` y se procesa tal y como se especificara para el elemento `italic`.

Definir una regla de estilo para el elemento asociado al icono de estado del texto

Las propiedades de estilo de un elemento asociado a un icono de estado del texto se pueden definir en una plantilla global (*imagen siguiente*).



La imagen anterior, por ejemplo, muestra las plantillas globales de dos elementos (`bold` e `italic`) y cada uno de estos elementos se asoció a un icono de estado del texto. El estilo se define seleccionando el componente `(contenido)` y definiendo el estilo deseado en la ventana Estilos.

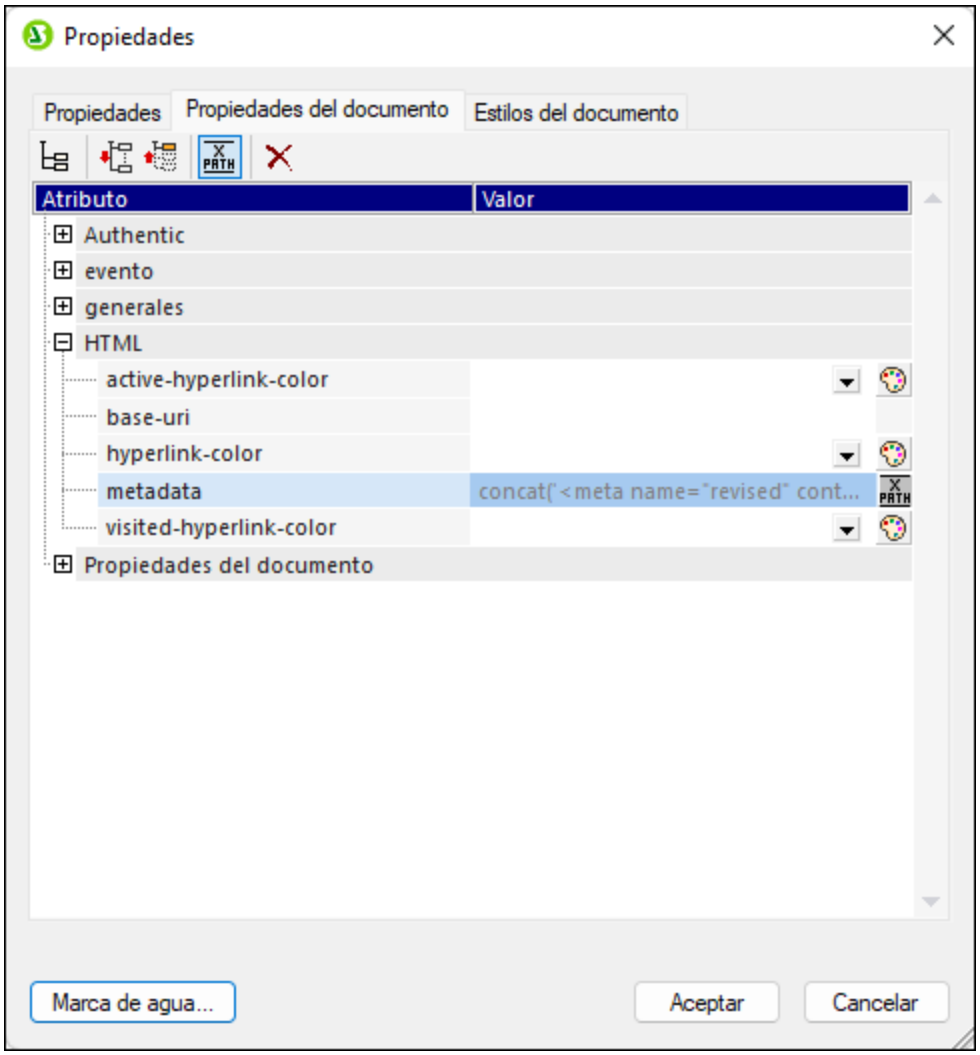
Como consecuencia, cuando el usuario de la vista Authentic inserte un fragmento de texto dentro de uno de estos elementos, la plantilla global correspondiente (con su estilo de texto) se aplicará al fragmento de texto.

© 2016-2022 Altova GmbH

8.6 Propiedades de un documento HTML

Puede definir las propiedades del documento HTML de salida en la pestaña *Propiedades del documento* del cuadro de diálogo Propiedades de la sección de documento inicial 423 o en la vista Propiedades al seleccionar la plantilla principal en la ventana Vista general del diseño 52.

Propiedades de la sección de documento inicial
Haga clic en el hipervínculo *Editar propiedades* de la barra del título de la sección de documento inicial 423 y seleccione la pestaña *Propiedades* del documento.

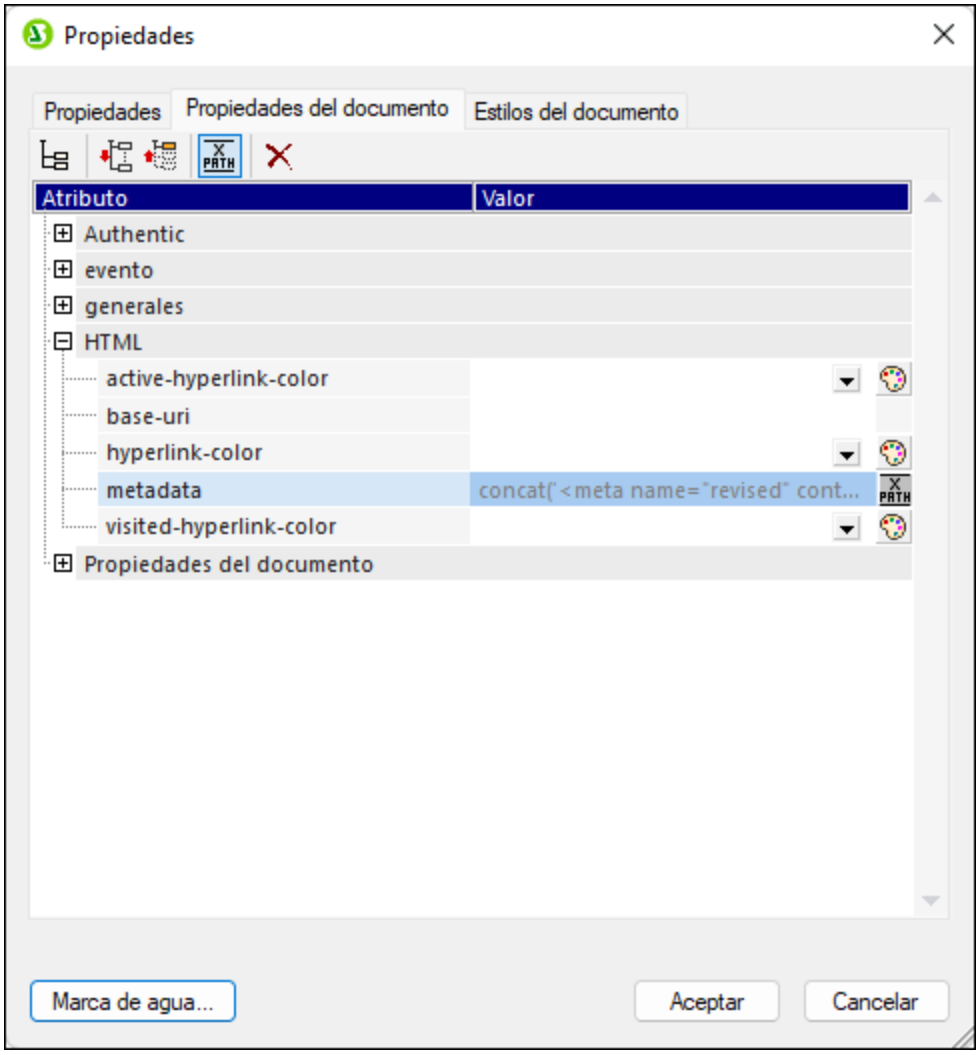


Aquí puede definir distintas propiedades del documento de salida en formato HTML que se genera. Tenga en cuenta que:

- Estas propiedades HTML se aplican a nivel del documento, por ejemplo, las propiedades `class` e `id` de la sección de *propiedades generales*.

- Las secciones *Propiedades del documento* y *HTML* contienen propiedades genéricas que tienen que ver con el documento HTML en su conjunto, como los colores de los vínculos, e información que se incluye en las **metaetiquetas** del documento HTML.
- La propiedad **metadatos** de la sección HTML permite introducir cualquier texto en el elemento **HEAD** del documento HTML (*véanse las imágenes*). El texto que se introduce aquí puede ser, por ejemplo, un **script** o un elemento **meta**, o varios. Puede introducir estos elementos HTML directamente como texto (sin comillas) o, como se ve en las imágenes, como expresión XPath. En el ejemplo de la imagen la expresión XPath define una metaetiqueta para revisar la fecha de un documento.

Propiedades de la plantilla principal
En la barra lateral Vista general del diseño 52, seleccione la plantilla principal. En la barra lateral Propiedades (*imagen siguiente*) ahora puede definir las propiedades del documento HTML de salida.



Las propiedades de este cuadro de diálogo son las mismas que las de la pestaña *Propiedades del documento* que describimos más arriba.

8.7 Diseñar medios impresos

Las propiedades de los formatos de salida para medios impresos (PDF, RTF y Word 2007+ en la edición *Enterprise Edition*; y RTF en la edición *Professional Edition*) se definen en el grupo de propiedades *Diseño de página* de la ventana Propiedades 59. En los documentos para medios impresos se pueden definir:

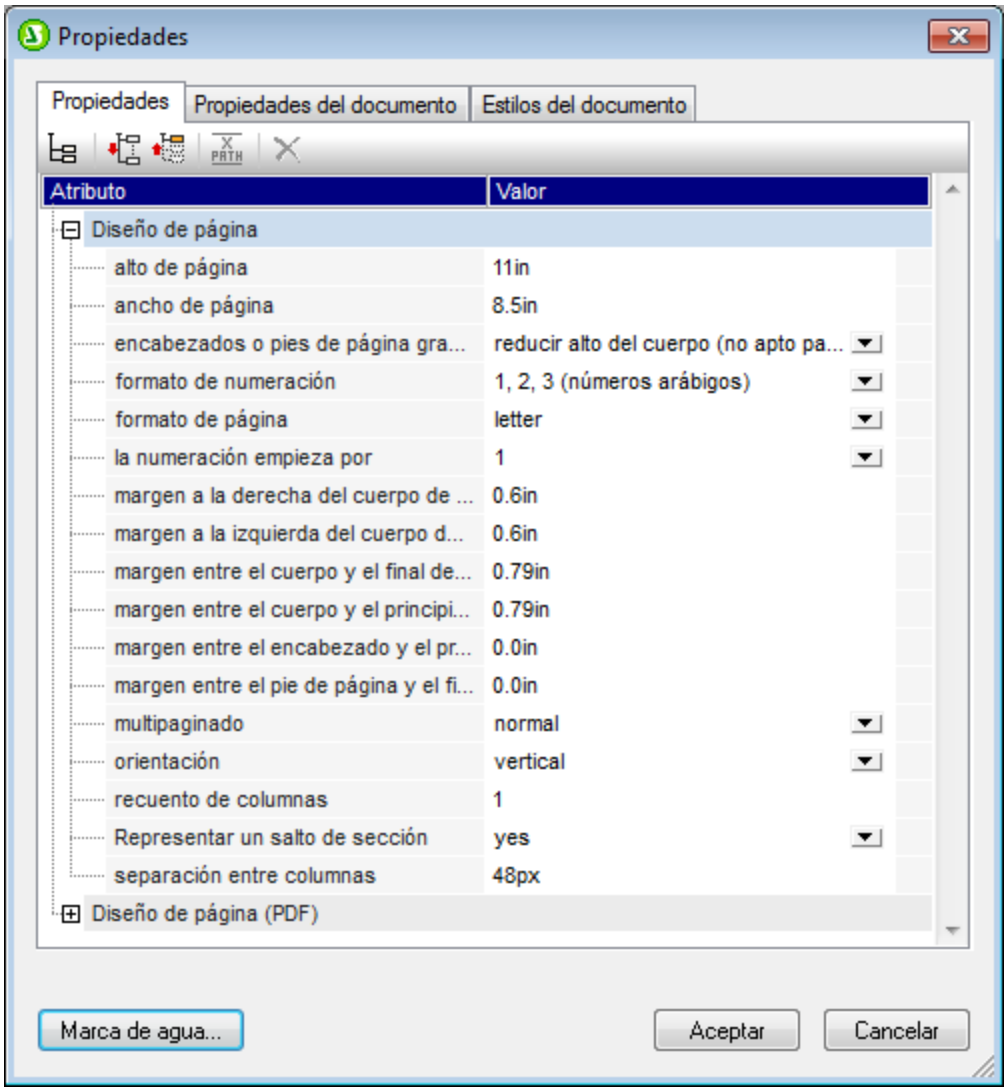
- **Secciones de documento**: cada una con distintas definiciones de página. Las propiedades que se pueden definir aparecen más abajo.
- **Dimensiones de página** (altura y ancho) y **orientación de página** (vertical y horizontal).
- Los **márgenes** para el cuerpo de la página y el espacio vertical disponible para el encabezado y el pie de página. También puede definir varias páginas como páginas contiguas (con márgenes simétricos) o con el mismo margen derecho e izquierdo en cada página.
- **Encabezados** y **pies de página** para cada sección.
- Estilos de **numeración** y el inicio de la numeración en cada sección. La numeración de páginas puede continuar en la siguiente sección.
- El número y el ancho de columnas de una página en cada sección.

Nota: como los diseños de página no se pueden aplicar al texto plano, las características de diseño que se describen en esta sección no se aplican a la salida en formato texto.

Ventana Propiedades

Las propiedades de la página se pueden definir por separado para cada sección del documento en el grupo de propiedades *Diseño de página* de la ventana Propiedades (*imagen siguiente*). Para acceder a estas propiedades haga clic en el enlace `Editar las propiedades...` de la **Sección inicial del documento** y de cada **Sección de documento** (*imagen siguiente*).

Al hacer clic en el enlace `Editar las propiedades...` aparece la ventana Propiedades (*imagen siguiente*).

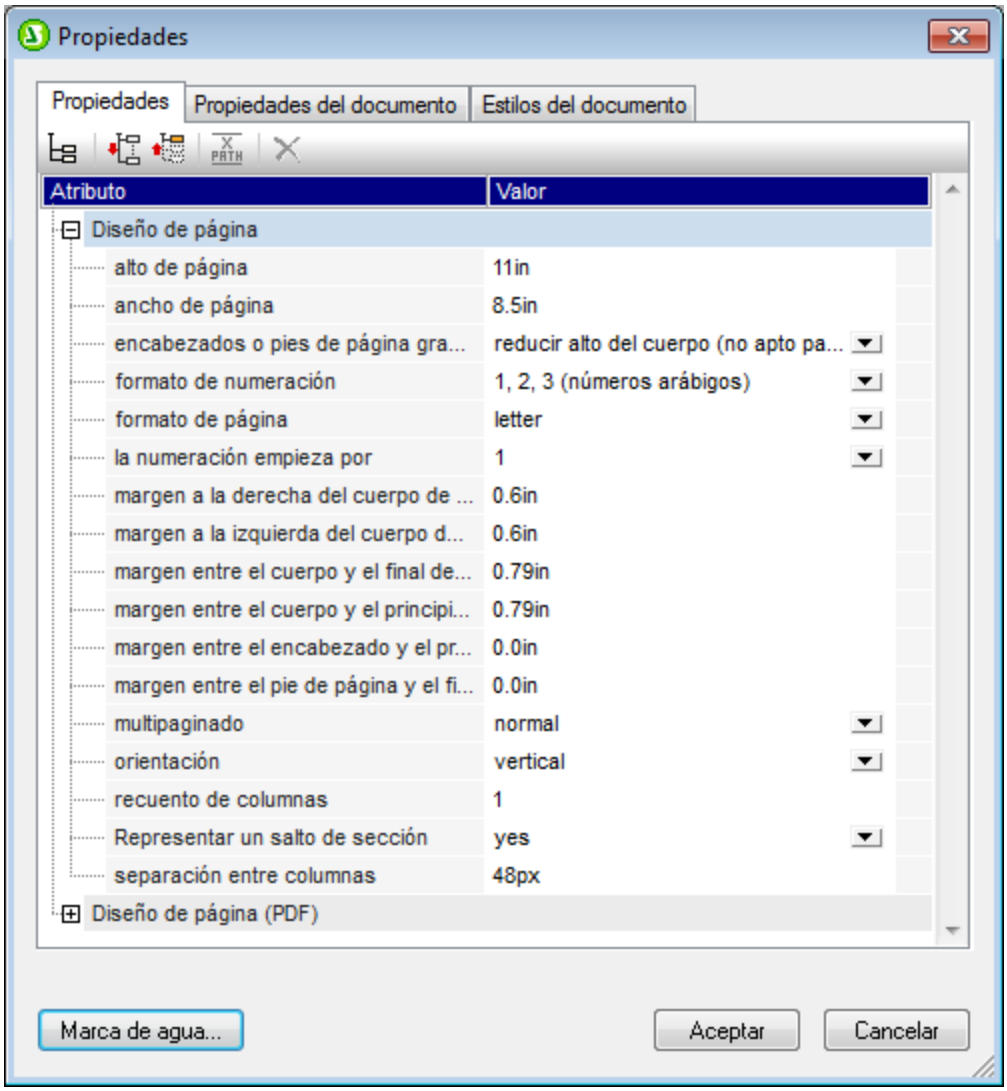


El apartado Propiedades de diseño de página 425 describe todas estas propiedades.

8.7.1 Secciones del documento

El diseño SPS puede contar con varias secciones de documento, cada una de ellas con distinta definición de página. Por ejemplo, un informe que contiene tablas de datos y texto que resume estos datos puede dividirse en dos secciones de documento: una sección de documento que incluya el texto descriptivo y que tenga orientación vertical y otra sección de documento con las tablas de datos y orientación horizontal. Una vez creadas las secciones, puede definir las propiedades de diseño de página 425 de cada una (*imagen siguiente*). Además cada sección de documento puede tener encabezados y pies de página 430 distintos.

 © 2016-2022 Altova GmbH



En StyleVision los diseños SPS nuevos incluyen por defecto una sección de documento llamada **Sección de documento inicial**. Esta sección de documento es la primera sección del diseño y no se puede eliminar. Entre las propiedades de la **Sección inicial del documento** se encuentran propiedades y estilos para todo el documento (consulte el apartado Sección inicial del documento 423).

Insertar secciones de documento

Siga estos pasos para añadir una sección de documento:

1. Ponga el cursor en la posición del diseño donde debe comenzar la nueva sección de documento.
2. Haga clic con el botón derecho y en el menú contextual seleccione **Insertar página, columna o sección de documento | Sección de documento nueva**. Este comando también está disponible en el menú **Insertar**. La sección de documento nueva se inserta en el diseño y se delimita con una barra de título de sección (*imagen siguiente*). En el documento de salida la sección de documento nueva comienza en una página nueva.

3. La sección de documento nueva tendrá las propiedades de diseño de página que tuviera la **Sección inicial del documento** cuando se creó la sección de documento nueva. Estas propiedades de diseño de página[425] de la sección de documento se pueden editar haciendo clic en el vínculo `Editar las propiedades...`. Si es necesario, puede añadir un encabezado y pie de página[430] distinto para la sección de documento (vínculo `Agregar encabezado o pie...`). Los apartados siguientes explican cómo definir propiedades de diseño de página[425] y encabezados y pies de página[430]. Cuando se añade un encabezado o un pie de página, aparecen dentro de la sección de documento, pero se pueden ocultar con el vínculo `Ocultar/mostrar encabezados o pies de página...`. Este vínculo solo aparece si se añadió un encabezado o pie de página en la sección de documento.
4. Si quiere crear una página nueva para la sección situada inmediatamente después de la **Sección inicial del documento**, vaya a las propiedades de diseño de página de la sección inicial de documento y asigne el valor `yes` a la propiedad `representar un salto de sección`. Esto hace que la primera sección de documento (no la sección inicial) empiece en una página nueva. Sin embargo, tenga en cuenta que puede suceder lo contrario. Si el valor de la propiedad es `no`, la primera sección de documento (no la sección inicial) empieza directamente después de la Sección inicial del documento, sin incluir un salto de página. Esto es más práctico si la sección inicial del documento está en blanco (p. ej. si solo contiene plantillas de procesamiento que no generan resultados). Por tanto, el valor `no` garantizaría que la primera página del documento de salida no esté en blanco (cuando al sección inicial del documento no tiene contenido y está totalmente en blanco y la siguiente sección empieza en una página nueva).

Es **importante** tener en cuenta que:

- En los documentos de salida RTF generados con diseños SPS XSLT 1.0 solamente se admiten secciones de documento que sean secundarios inmediatos de la plantilla principal. Esta restricción no afecta a los documentos de salida RTF generados con diseños SPS XSLT 2.0 o XSLT 3.0.
- En el documento de salida cada sección de documento comienza en una página nueva.
- Las propiedades de los márgenes de página también afectan al formato de salida HTML.
- Cuando en un diseño hay varias secciones de documento, los valores de la propiedad `márgenes simétricos` y de las propiedades `margen izquierdo` y `margen derecho` se toman de la **Sección inicial del documento**. Los valores de estas propiedades en las demás secciones de documento se ignoran.

Eliminar una sección de documento

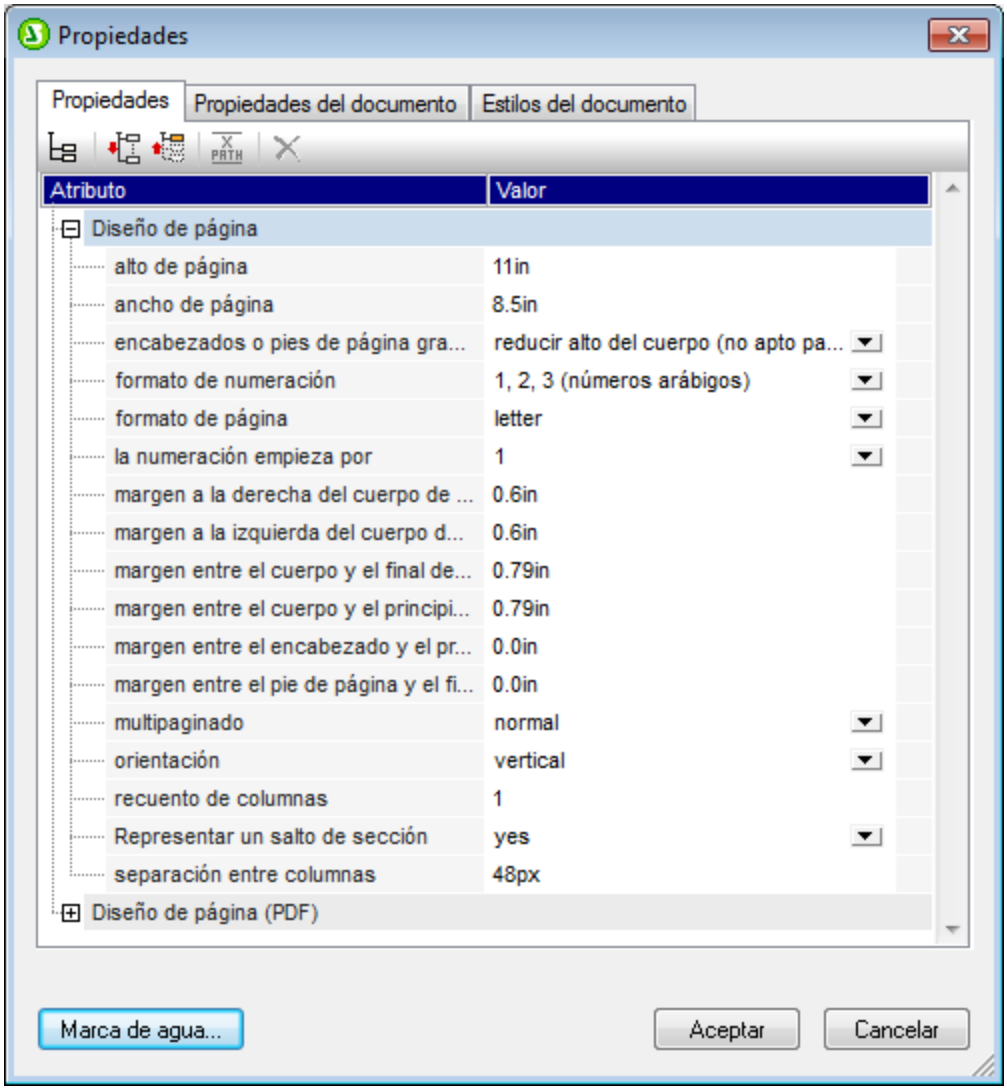
Para eliminar una sección de documento, haga clic con el botón derecho sobre las palabras **Sección del documento** y después en el comando **Editar | Eliminar** del menú contextual. La sección de documento se elimina y la barra de título de la sección desaparece. Al eliminar la sección de documento se eliminan también las propiedades de diseño de página y los encabezados y pies de página de la sección. Sin embargo, el contenido de la sección no se borra del diseño.

 © 2016-2022 Altova GmbH

8.7.1.1 Sección inicial del documento

Tanto si el diseño SPS tiene una sección de documento como si tiene varias, las propiedades del diseño en general son las de la **Sección inicial del documento** (la primera sección de documento del diseño, *imagen siguiente*). La portada[424] del documento también se define en la **Sección inicial del documento**.

Para editar las propiedades del documento haga clic en el vínculo `Editar las propiedades...` de la barra de título **Sección inicial del documento**. Esto abre el cuadro de diálogo "Propiedades" de la sección inicial del documento (*imagen siguiente*). Este cuadro de diálogo tiene tres pestañas: (i) *Propiedades* (propiedades de diseño de página básicas), (ii) *Propiedades del documento* (para documentos HTML) y (iii) *Estilos del documento*.



Propiedades de diseño de página

Las propiedades de diseño de página de la sección inicial del documento afectan a la primera sección de documento del diseño SPS. En los diseños que tengan una sola sección de documento estas propiedades se aplican a todo el diseño SPS. Cuando se crea una sección de documento nueva, se crea con las propiedades que tenga la sección inicial del documento en ese momento. Las propiedades de la sección de documento nueva se pueden editar después. Si se cambia una propiedad de la sección inicial del documento, el cambio no se refleja en las demás secciones de documento. Las secciones de documento nuevas se crean con los valores más recientes de la sección inicial del documento. El apartado Propiedades de diseño de página 425 incluye más información sobre estas propiedades. Con el botón **Marca de agua...** puede abrir el cuadro de diálogo donde puede definir una marca de agua 439 para las páginas de la sección. Por tanto, puede definir una marca de agua distinta para cada sección del documento.

Propiedades del documento

Las propiedades del documento HTML de salida se definen en la pestaña Propiedades del documento 417.

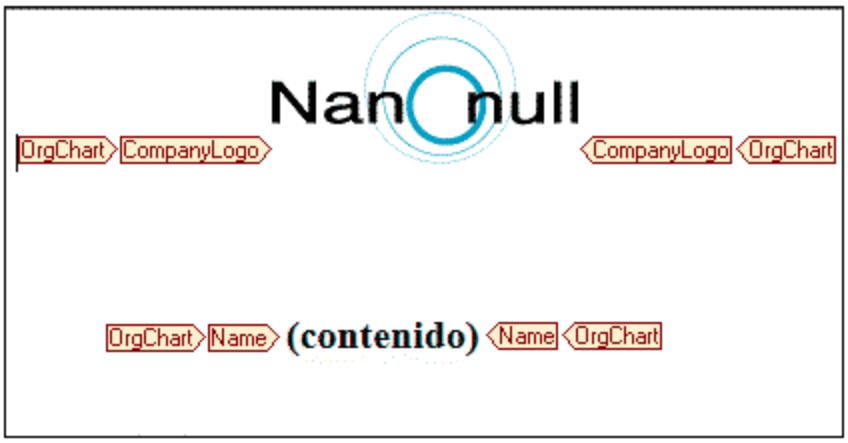
Estilos del documento

Los estilos definidos en la pestaña *Estilos del documento* se aplican a todo el documento de diseño. Si un diseño tiene varias secciones de documento, los elementos de diseño de cada sección heredan las propiedades de estilo de la sección inicial del documento. Para invalidar los estilos heredados de determinado elemento de diseño, debe especificar nuevos valores de estilos en el elemento de diseño. Para ello haga clic en el elemento de diseño y defina sus nuevos estilos en la ventana Estilos.

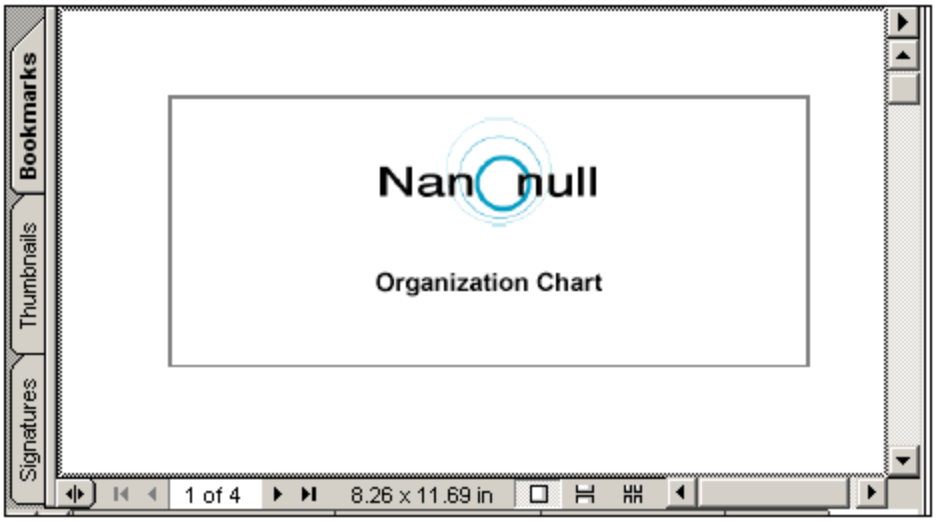
Portadas

Si el documento necesita una portada, diséñela al principio de la **Sección inicial del documento**. Para garantizar que el resto del documento comienza en una página nueva, inserte un salto de página (**Insertar | Insertar página, columna o sección de documento | Página nueva**) debajo de la plantilla de la portada. Si quiere que las propiedades de diseño de página de la portada sean distintas a las de las páginas siguientes, utilice la **Sección inicial del documento** para la portada e inserte las páginas siguientes en una sección de documento nueva.

Por ejemplo, esta portada:



Haga clic en la pestaña de vista previa **RTF** para ver el resultado de la transformación.



8.7.1.2 Propiedades de diseño de página

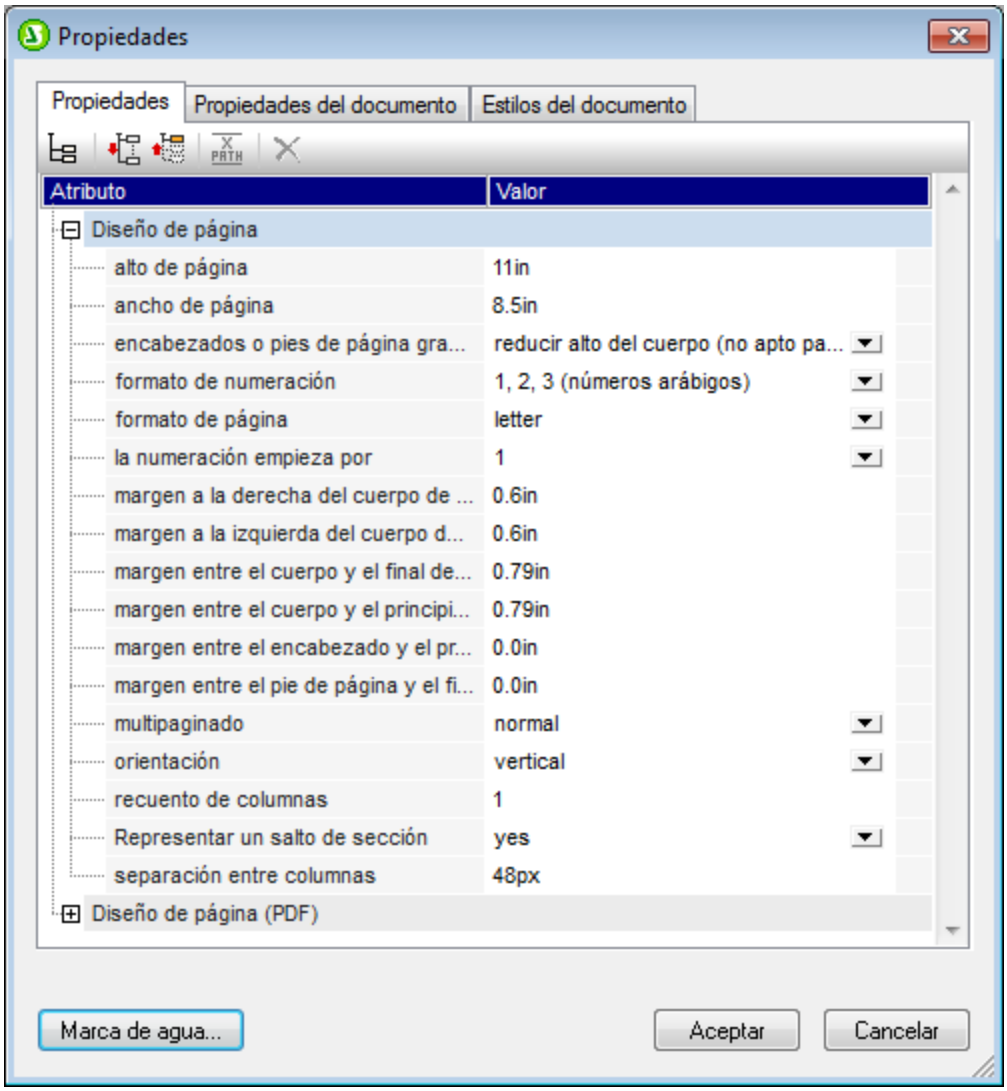
A cada sección del documento se le asignan propiedades de página en el grupo de propiedades *Diseño de página*. Si el SPS tiene una sola sección de documento, entonces las propiedades de página de esa sección son las propiedades de página de todo el documento.

Nota: debido a una limitación de FOP, las propiedades de *Diseño de página* solo están disponibles para la Sección inicial del documento cuando se usa FOP para la generación de resultados.

Editar las propiedades de página de una sección de documento

Para editar las propiedades de página de una sección de documento, haga clic en el vínculo `Editar las propiedades...` de la barra de título **Sección inicial del documento** 420 o 420 **Sección del documento** 420 (*imagen siguiente*).

Esto abre el cuadro de diálogo "Propiedades" por el grupo de propiedades *Diseño de página* (*imagen siguiente*).



También puede hacer clic simplemente en la barra de título **Sección inicial del documento** 420 o 420 **Sección del documento** 420 y las propiedades de *Diseño de página* aparecen en la ventana lateral Propiedades.

Tamaño de la página

Tres propiedades determinan el tamaño de las páginas de una sección de documento: (i) **`alto de página`**, (ii) **`ancho de página`** y (iii) **`tamaño`**. El tamaño de la página puede definirse de dos formas distintas:

- Puede seleccionar un tamaño de página predefinido en el cuadro combinado de la propiedad `formato de página`. Al seleccionar una opción en este cuadro combinado se actualizan automáticamente los valores de las propiedades `alto de página` y `ancho de página`, dependiendo de la opción elegida.
- Puede especificar los valores de las propiedades `alto de página` y `ancho de página` a mano. En este caso la propiedad `formato de página` tendrá el valor `tamaño personalizado`.

Las dimensiones válidas (para las propiedades `alto de página` y `ancho de página`) son: pulgadas (`in`), centímetros (`cm`), milímetros (`mm`), picas (`pc`), puntos (`pt`), píxeles (437 `px` 437) 437 y emes (`em`). Recuerde que (i) es obligatorio incluir la unidad de medida; (ii) que no puede haber espacios entre el número y la unidad de

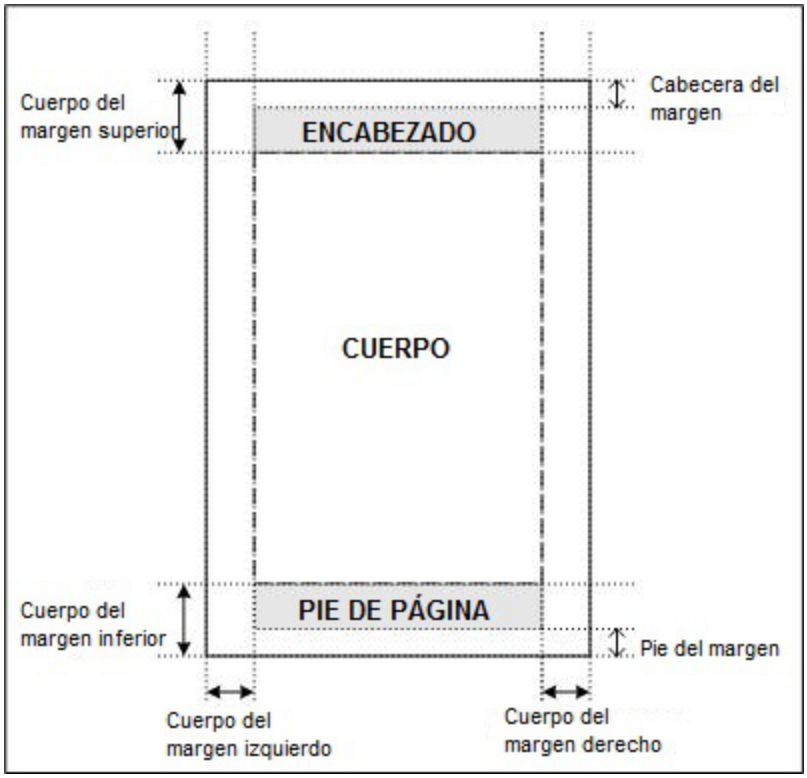
medida; (iii) y que no hay una unidad predeterminada. Si introduce un valor o una unidad no válidos, el valor de la propiedad se resalta en rojo.

Márgenes de página
Los márgenes superior, inferior, izquierdo y derecho de una página se pueden configurar con estas cuatro propiedades: `margen entre el cuerpo y el principio del documento`, `margen entre el cuerpo y el final del documento`, `margen a la izquierda del cuerpo`, `margen a la derecha del cuerpo`. Para especificar el margen escriba el número en la propiedad correspondiente seguido de la unidad de longitud: pulgadas (`in`), centímetros (`cm`), milímetros (`mm`), picas (`pc`), puntos (`pt`), píxeles (437 `px` 437) 437 o emes (`em`). Recuerde que (i) es obligatorio incluir la unidad de medida; (ii) que no puede haber espacios entre el número y la unidad de medida; (iii) y que no hay una unidad predeterminada. Si introduce un valor o una unidad no válidos, el valor de la propiedad se resalta en rojo.

El área del cuerpo del documento viene dada por los márgenes definidos aquí (*imagen siguiente*).

Margen del encabezado y del pie de página

La propiedad `margen entre el encabezado y el principio de la página` y la propiedad `margen entre el pie de página y el final` determinan la distancia entre el principio de la página y el encabezado y entre el final de la página y el pie de página respectivamente (*imagen siguiente*).



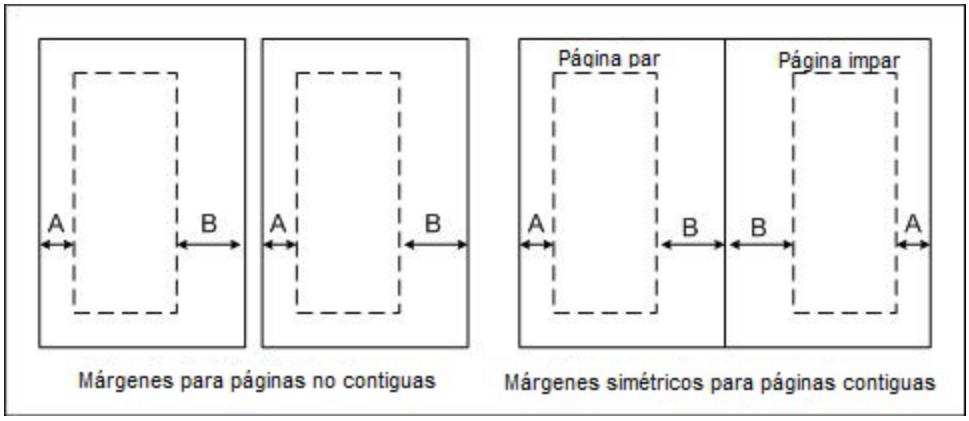
La extensión vertical del encabezado y del pie de página viene dada por su contenido. Asegúrese de que la extensión vertical del encabezado más su margen no sobrepasa el `margen entre el cuerpo y el principio de la página`. De lo contrario, el encabezado será demasiado grande y no cabrá en el espacio que tiene definido. Igualmente, asegúrese de que la suma de la extensión vertical del pie de página y su margen no sobrepasa el valor del `margen entre el cuerpo y el final de la página`.

Si el encabezado o el pie de página es demasiado grande y no cabe en el espacio que tiene definido, entonces interviene la propiedad `encabezados o pies de página grandes`, que determina cómo se tratan los encabezados y los pies de página en los documentos de salida RTF. Si define el valor `superponer cuerpo del texto` para la propiedad `encabezados o pies de página grandes`, entonces el encabezado y el pie de página sobredimensionados se superponen encima del texto del cuerpo o al contrario. Si por el contrario define el valor `reducir alto del cuerpo`, entonces la extensión vertical del cuerpo se reduce para que quepan el encabezado y el pie de página sobredimensionados.

Multipaginado

La propiedad `multipaginado` tiene dos opciones:

- Si define el valor `normal` para la propiedad `multipaginado`, entonces todas las páginas del documento de salida tendrán el mismo valor para los márgenes izquierdos y para los márgenes derechos (*imagen siguiente*).
- Si en cambio define el valor `márgenes simétricos` para la propiedad `multipaginado`, entonces se considera que las páginas del documento son contiguas (*imagen siguiente*). Esto significa que en las páginas impares (situadas en el lado izquierdo) el margen izquierdo (propiedad `margen a la izquierda`) es el margen exterior, mientras que el margen derecho (propiedad `margen a la derecha`) es el margen interior. Por su parte, en las páginas pares (situadas en el lado derecho), el margen izquierdo (propiedad `margen a la izquierda`) es el margen interior, mientras que el margen derecho (propiedad `margen a la derecha`) es el margen exterior.



StyleVision aplica los valores de las propiedades `margen a la izquierda` y `margen a la derecha` automáticamente a las páginas impares y los invierte para las páginas pares: el valor del margen interior de las páginas impares se convierte en el valor del margen exterior de las páginas pares.

Nota: si un diseño SPS tiene varias secciones de documento, el valor de los márgenes simétricos se toma de la sección inicial del documento. Los valores del margen izquierdo y derecho también se toman de la sección inicial del documento. Los valores de estas propiedades en las demás secciones se ignoran.

Orientación de la página

La propiedad `orientación` puede tener el valor `vertical` o el valor `horizontal`.

 © 2016-2022 Altova GmbH

Columnas

Las columnas y el ancho de las columnas se especifica con estas dos propiedades: `recuento de columnas` y `separación entre columnas`. El ancho de una columna es el ancho del cuerpo de la página menos la suma de la separación entre las columnas, dividido por el número de columnas.

El texto se usa para rellenar las columnas de la página una por una. Solamente se crea una página nueva si se rellenaron todas las columnas de la página. El salto de columna se puede forzar insertando una columna nueva en el punto deseado del diseño. Para forzar el salto de columna, haga clic con el botón derecho en la posición deseada y seleccione el comando **Insertar página, columna o sección de documento | Columna nueva** del menú contextual.

Numeración de las páginas

Hay dos propiedades que permiten definir la numeración de las páginas: `formato de numeración` y `la numeración empieza por`.

- Para definir el formato de numeración de las páginas seleccione una de las opciones predefinidas disponibles en el menú desplegable de la propiedad `formato de numeración`. La opción seleccionada afecta al número total de páginas 835 (si se insertó).
- Puede especificar que la numeración de las páginas empiece por un entero positivo. Este entero se especifica en la propiedad `la numeración empieza por`. Si la numeración debe continuar desde la sección anterior, entonces deje esta propiedad vacía o elija el valor `auto`.

Nota: para insertar números de página en el documento inserte un marcador de posición de número de página (comando **Insertar página, columna o sección de documento | Número de página**). El número total de páginas del documento de salida se inserta con el comando **Insertar página, columna o sección de documento | Total de páginas**.

Nota: MS Word no siempre actualiza automáticamente el número de páginas. Para actualizar a mano el número de páginas en MS Word, pulse **Ctrl+A** y **F9**.

Numeración de las páginas en el formato de salida RTF

Para que la numeración de las páginas aparezca en documentos RTF en MS Word, seleccione todo el contenido del documento en MS Word (**Edición | Seleccionar todo** o **Ctrl+A**) y después pulse **F9**. Esto hace que aparezca la numeración de las páginas (si se insertó).

Total de páginas

Para que el documento de salida incluya el número total de páginas, utilice la función total de páginas 835.

Página donde empieza la sección de documento

La propiedad `la sección empieza en` especifica si la sección de documento (excepto la **Sección inicial del documento**) empieza en la página siguiente (tanto si es una página par como impar) o si empieza en una página par o en una página impar. Por ejemplo, si la sección de documento anterior terminó en una página impar y especifica que la sección de documento actual empiece en una página impar, entonces se dejará en blanco la página par que aparece después de la sección anterior. Recuerde que es la numeración de páginas del documento la que determina si una página es par o impar. La numeración de página definida por el usuario no afecta a esta propiedad.

Salto de página después de la Sección inicial del documento

Si quiere iniciar una sección en una página nueva inmediatamente después de la Sección inicial del documento, debe asignar el valor `yes` a la propiedad `representar un salto de sección` (en las propiedades de diseño de página de la Sección inicial del documento). Esto inicia la segunda sección de documento (es decir, la primera sección posterior a la Sección inicial del documento) en una nueva página. Recuerde que esta propiedad solamente está disponible si se trata de la Sección inicial del documento.

Sin embargo, tenga en cuenta que puede suceder lo contrario. Si el valor de la propiedad es `no`, la primera sección de documento (no la sección inicial) empieza directamente después de la Sección inicial del documento, sin incluir un salto de página. Esto es más práctico si la sección inicial del documento está en blanco (p. ej. si solo contiene un grupo de plantillas que no generan resultados). Por tanto, el valor `no` garantizaría que la primera página del documento de salida no esté en blanco (que es lo que pasaría si hubiera que representar un salto de página después de una Sección inicial de documento que no genera ningún resultado).

8.7.1.3 Encabezados y pies de página: 1ª parte

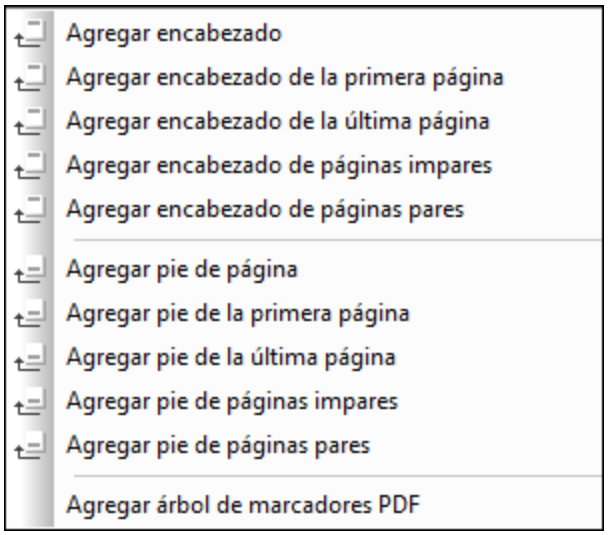
En el diseño SPS puede añadir encabezados y pies de página **en todas las secciones del documento**, incluida la sección inicial del documento.

Añadir un encabezado o pie de página a una sección de documento

Para añadir un encabezado o pie de página a una sección de documento haga clic en el vínculo `Agregar encabezado o pie...` de la barra de título **Sección inicial del documento** 420 o 420 **Sección del documento** 420 (*imagen siguiente*).

Al hacer clic en este vínculo aparece un menú contextual (*imagen siguiente*), donde puede seleccionar el tipo de encabezado o de pie de página. Puede añadir encabezados y pies de página distintos en las páginas pares e impares o puede añadir el mismo encabezado y pie de página para todas las páginas de la sección. Además puede crear una plantilla de encabezado o pie de página para la primera página (**Agregar encabezado de la primera página** / **Agregar pie de la primera página**) o para la última página de la sección de documento (**Agregar encabezado de la última página** / **Agregar pie de la última página**). Esto es ideal si necesita insertar un encabezado/pie de página distinto en la primera o la última página de una sección. Por ejemplo: cuando la primera página es la portada (en este caso podría crear un encabezado/pie de página vacíos).

 © 2016-2022 Altova GmbH

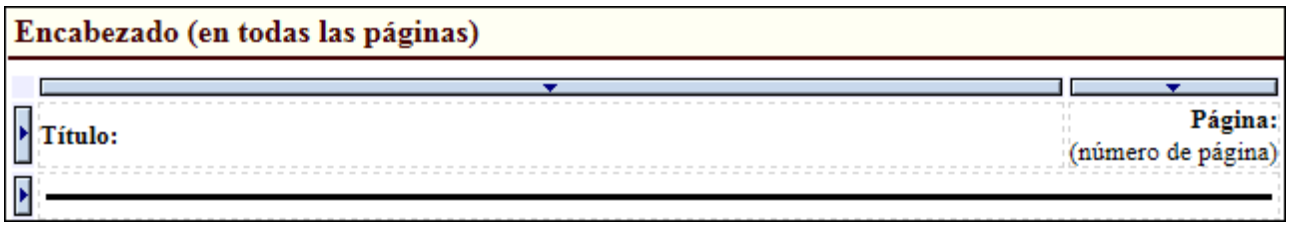


Al hacer clic en uno de los comandos del menú contextual (*imagen anterior*) se crea una plantilla para el encabezado/pie de página dentro de la sección de documento en la vista 37 **Diseño** 37. En la vista **Diseño** puede ocultar los encabezados/pies de página de una sección de documento con ayuda del vínculo `Ocultar encabezados o pies de página...` de la barra de título de la sección de documento (*imagen anterior*).

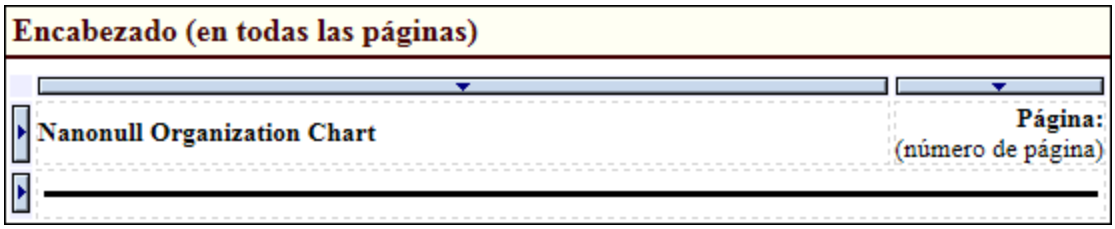
Nota: solamente el formato de salida PDF admite encabezados y pies de página diferentes en la última página.

Diseñar el encabezado o pie de página en la vista Diseño

La plantilla de encabezado/pie de página se diseña igual que cualquier otra plantilla. Puede insertar contenido estático o arrastrar componentes desde la Estructura del esquema y después aplicar estilos a los componentes. Veamos un ejemplo. Cuando se añade un encabezado nuevo, la plantilla del encabezado tiene este aspecto:

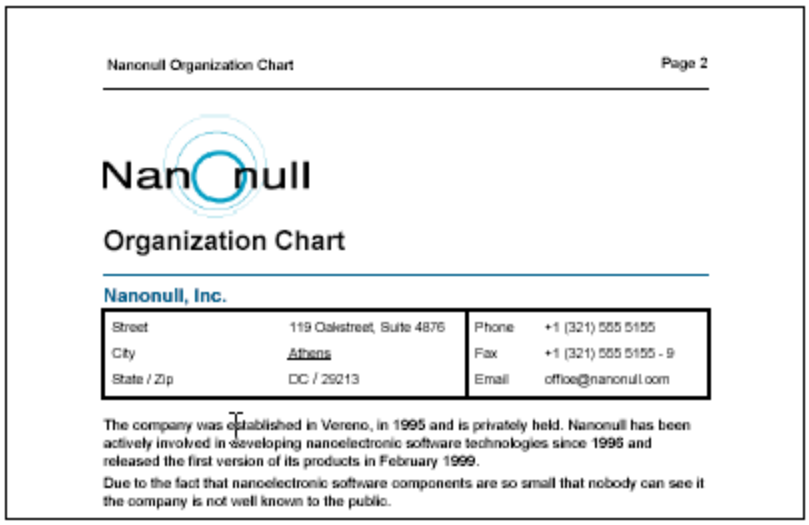


Ahora puede cambiar los componentes del encabezado y añadirle contenido estático, contenido dinámico e imágenes.



Haga clic en la pestaña de vista previa **RTF** para ver el resultado de la transformación. Por ejemplo, la imagen siguiente muestra la segunda página del diseño en la vista previa **RTF**, donde se puede observar el

encabezado de la imagen anterior. Para ver los números de página en los documentos de salida RTF seleccione **Edición | Seleccionar todo** (o **Ctrl+A**) en MS Word y después pulse **F9**.



Es muy **importante** tener en cuenta que:

- La extensión vertical del encabezado o pie de página no debe superar la extensión el margen superior o inferior correspondiente menos la extensión de la cabecera del margen o del pie del margen respectivamente (ver Propiedades de diseño de página 427 para más información). La extensión vertical del encabezado o pie de página, por tanto, depende de los márgenes superior e inferior y de la cabecera y del pie de los márgenes.
- Puede definir un encabezado o pie de página (i) bien para todas las páginas de la sección o (ii) bien para las páginas pares e impares de la sección. Además puede insertar un encabezado o pie de página distinto en la primera y en la última página de cada sección (ver el apartado Encabezados y pies de página: segunda parte 432 para más información.)
- La numeración de las páginas de una sección de documento comienza con el número especificado en las Propiedades de diseño de página 427.

Eliminar un encabezado o pie de página

Para eliminar un encabezado o pie de página de una sección de documento, haga clic con el botón derecho en la barra de título del encabezado o pie de página y seleccione **Editar | Eliminar**.

8.7.1.4 Encabezados y pies de página: 2ª parte

En este apartado explicamos cómo crear este tipo de encabezados y pies de página:

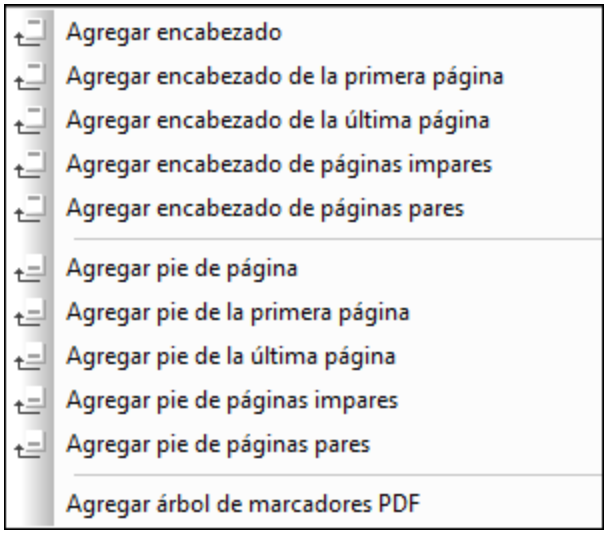
- Encabezados / pies de página distintos para páginas pares e impares 433
- Encabezados / pies de página distintos para cada sección del documento 433
- Simular encabezados y pies de página dentro de una página 433
- Encabezados / pies de página con número subtotal de páginas 434

© 2016-2022 Altova GmbH

Encabezados / pies de página distintos para páginas pares e impares

En cada sección de documento se pueden crear encabezados / pies de página distintos para las páginas pares e impares.

Para crear encabezados distintos para páginas pares e impares, haga clic en el vínculo `Agregar encabezado o pie de página...` de la barra de título de la sección de documento. En el menú que aparece seleccione los comandos **Agregar encabezado de páginas impares** y **Agregar encabezado de páginas pares** (*imagen siguiente*). Esto crea dos plantillas de encabezado: una para las páginas impares y otra para las pares. Inserte el contenido de los encabezados en cada plantilla.



Los pies de página se crean de la misma manera (con los comandos **Agregar pie de páginas impares** y **Agregar pie de páginas pares**).

Encabezados / pies de página distintos para cada sección del documento

En StyleVision puede crear encabezados / pies de página distintos para cada sección del documento. Para ello haga clic en el vínculo `Agregar encabezado o pie de página...` de la barra de título de la sección de documento o de la sección inicial del documento. En el menú contextual que aparece puede elegir el tipo de encabezado o pie de página. Tenga en cuenta que:

- Puede agregar encabezados y pies de página distintos para páginas pares e impares o puede agregar un encabezado/pie de página común para todas las páginas de la sección de documento.
- Puede añadir un encabezado/pie de página distinto en la primera y en la última página de la sección de documento.
- Los números de página de la sección de documento pueden continuar desde la sección de documento anterior o empezar por un número determinado 429.
- El número total de páginas 429 es el recuento de páginas de todo el documento y no de la sección de documento actual.

Simular encabezados y pies de página dentro de una página

También puede diseñar a mano el encabezado/pie de página dentro de un contenedor de diseño 204. El proceso consiste en diseñar una página como contenedor de diseño. El encabezado y el pie de página se

crean dentro de tablas estáticas 157 ubicadas al principio y al final de la página respectivamente. Si necesita diseñar más de una página, utilice varios contenedores de diseño separados por saltos de página 835 (**Insertar | Página, columna o sección de documento | Página nueva**).

Encabezados / pies de página con subtotal de páginas y total acumulado

Cuando un documento incluye una lista de elementos numéricos y la lista ocupa varias páginas, si quiere calcular el total de elementos puede ser práctico calcular el subtotal de cada página o el total acumulado y mostrarlo en el encabezado o en el pie de cada página. Por ejemplo, el archivo `Subtotals.sps` de la carpeta (Mis) Documentos 34, `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples\Tutorial\Subtotals\`, muestra cómo se crean totales acumulados y cómo se insertan en encabezados y pies de página.

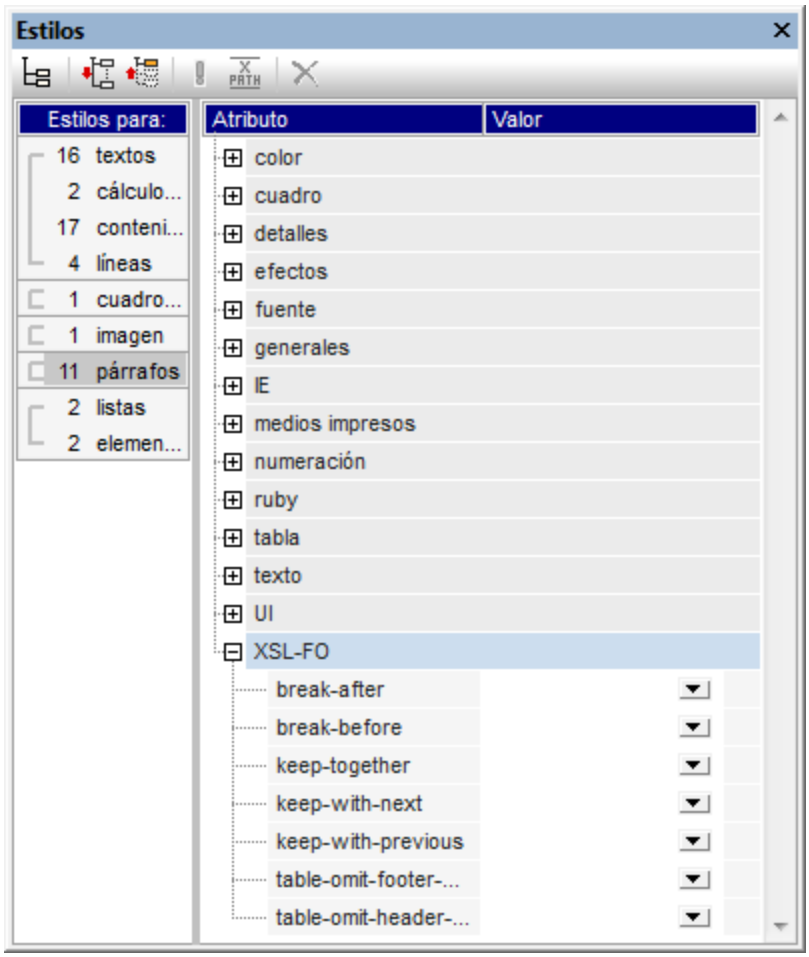
El proceso de creación del diseño SPS sería este:

- Como la enumeración está en una tabla y el tamaño de la tabla no se puede ajustar automáticamente a la página imprimida, debe especificar el número de filas que debe haber en cada página. El número de filas se puede dar en dos variables que se definen en la plantilla del nivel principal (la plantilla `$XML`) y se llaman `RowsOnFirstPage` (*Filas en la primera página*) y `RowsPerPage` (*Filas por página*).
- El recuento de páginas se obtiene al dividir el número total de elementos de la lista por el número de filas de cada página (ajustado para tener en cuenta el número de filas de la primera página, que es otro). El recuento de páginas se almacena en una variable llamada `CountOfPages` (*número de páginas*), que se define en la plantilla `$XML`.
- Se crea una plantilla definida por el usuario para la secuencia `1` a `$CountOfPages` y también se crea una tabla estática dentro de esta plantilla. En esta plantilla hay definidas dos variables que calculan qué fila es la primera fila (`$RowFrom`) y la última fila (`$RowTill`) de cada página. Las filas de la tabla las genera una plantilla definida por el usuario, que selecciona los componentes del archivo XML (elementos `file`) según su posición con respecto a los valores `$RowFrom` y `$RowTill`. Si la posición del elemento `file` es un entero situado entre los valores de las variables `$RowFrom` y `$RowTill` de la página actual, entonces se genera una fila para el elemento `file` actual.
- Los totales acumulados los genera un cálculo automático y se insertan en filas al principio y al final de las tablas. Observe que las expresiones XPath utilizadas para generar los totales acumulados al principio y al final de la página son diferentes.
- Los encabezados y pies de página en tablas, al principio y al final de la tabla principal de la página respectivamente. Los cálculos automáticos para generar totales acumulados se insertan en las plantillas del encabezado y del pie de página.
- Al final de cada página se inserta un salto de página.

8.7.2 Saltos de página

StyleVision ofrece estilos especiales para trabajar con saltos de página en el formato de salida PDF (*solamente compatible con la edición Enterprise edition*). Se trata de las propiedades del grupo *XSL-FO* de la ventana Estilos (*imagen siguiente*). Este grupo de propiedades permite especificar si el bloque actual del documento de diseño (el bloque donde está el cursor en ese momento) debería tener un salto de página/columna antes o después o si debería conservarse junto a los bloques adyacentes. También hay dos propiedades llamadas `table-omit` con las que puede establecer si en los saltos de página se conservan (se repiten) u omiten los encabezados (`table-omit-header`) y pies de página (`table-omit-footer`) de las tablas. Para más información sobre estas propiedades consulte la especificación XSL-FO.

 © 2016-2022 Altova GmbH



Los saltos de página y las líneas viudas y huérfanas de una página se pueden controlar con las correspondientes propiedades del grupo `medios impresos` de la ventana Estilos (*imagen anterior*).

8.7.3 Notas al pie

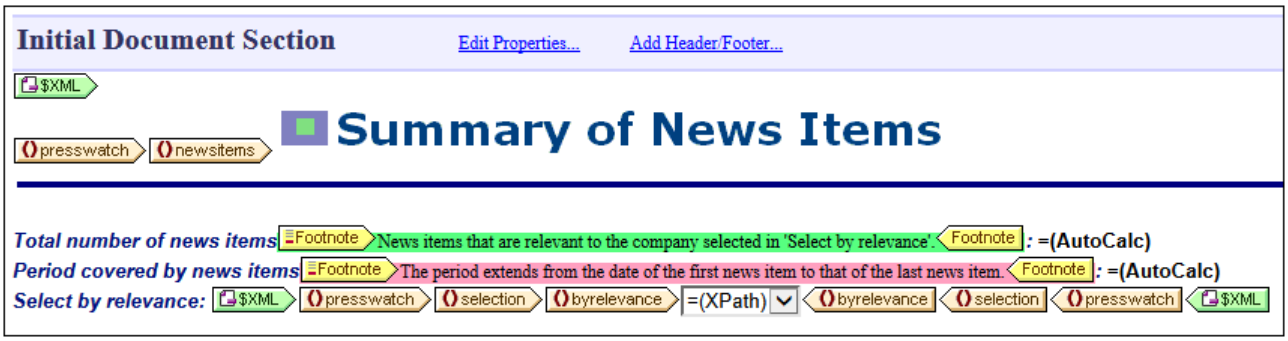
Puede añadir notas al pie en un documento añadiendo el componente `Nota al pie` (**Insertar | Insertar nota al pie**) en la posición donde quiere que aparezca el número de nota al pie. Las notas al pie se pueden usar en los formatos para medios impresos(RTF).

Cuando trabaje con notas al pie debe tener en cuenta que:

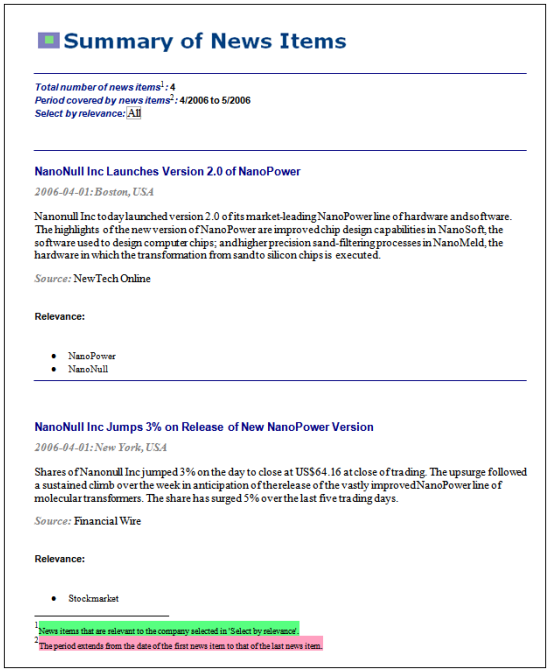
- El texto de la nota al pie debe colocarse entre las etiquetas del componente `Nota al pie` y puede aplicar formato al texto de la nota al pie.
- En el documento de salida el número de la nota al pie aparece en la posición donde se añadió la nota al pie. El texto, sin embargo, aparece al final de la página junto con el número de nota al pie que le corresponde.
- En el documento de salida el texto de la nota al pie tendrá el formato del texto que se insertó dentro del componente `Nota al pie` en el diseño.
- En el documento de salida las notas al pie se numeran automáticamente.
- Si se generan varios documentos de salida, la numeración de las notas al pie empieza desde cero en

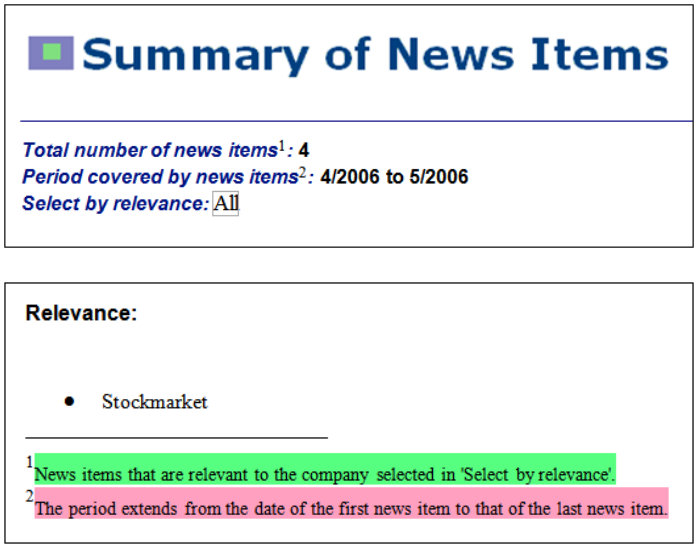
cada documento de salida.

En el ejemplo de la siguiente imagen se insertaron tres componentes `Nota al pie`. El texto de la nota al pie se insertó entre las etiquetas del componente y se le aplicó formato.



En las siguientes imagen puede ver el resultado. A la izquierda puede ver la página entera y a la derecha puede ver en detalle el número de las notas al pie y el texto de la nota al pie.





Nota: no se puede aplicar formato a los números de las notas al pie.

© 2016-2022 Altova GmbH

8.7.4 Resolución en píxeles

Si utiliza píxeles como medida de unidad en su diseño SPS, debería tener en cuenta que los píxeles son una medida de resolución de pantalla. Las medidas absolutas correspondientes para medios impresos pueden diferir de lo que aparece en pantalla. Este apartado explica:

1. Por qué hay dos formas de definir el tamaño en píxeles: (i) *tamaño abstracto*, en píxeles y (ii) *tamaño real*, que se obtiene resolviendo el tamaño abstracto en función de una resolución de pantalla concreta.
2. Qué hace StyleVision para resolver este problema.

De píxeles a puntos

Para entender qué factores afectan a las dimensiones abstractas y reales de las medidas definidas en píxeles es necesario señalar que:

- El píxel es una unidad relativa: su tamaño depende de la resolución de la pantalla. Cuanto mayor es la resolución, más pequeño es el píxel. La resolución de pantalla se da en ppp (puntos por pulgada). Un punto en una pantalla es un píxel. Es decir, si la resolución de pantalla es de 72 ppp, entonces hay 72 píxeles (puntos) en una pulgada de una línea de píxeles. Si la resolución de pantalla es 96 ppp, entonces hay 96 píxeles en una pulgada. El número de píxeles que hay en una pulgada de la pantalla depende de la resolución de la pantalla. Las resoluciones de pantalla más comunes son 72 ppp, 96 ppp y 120 ppp. Cuanto más ppp, más pequeños son los píxeles.
- El punto es una unidad absoluta de longitud utilizada sobre todo en la industria gráfica: 72 puntos equivalen a una pulgada.
- De lo anterior se deduce, por tanto, que un píxel equivale a un punto si la resolución de pantalla es de 72 ppp. En las demás resoluciones de pantalla se puede calcular la longitud absoluta. La tabla que aparece a continuación muestra la longitud absoluta (en puntos) de 100 píxeles en diferentes resoluciones de pantalla.

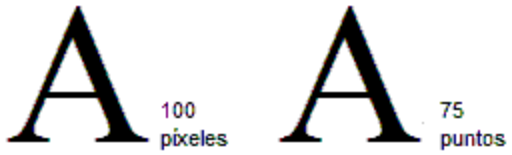
Resolución	Píxeles	Puntos	Factor
72 ppp	100	100	1,00
96 ppp	100	75	0,75
120 ppp	100	60	0,60

Para convertir píxeles en puntos en cada resolución de pantalla podemos usar un factor que viene dado por la relación entre puntos por pulgada (72) y píxeles por pulgada, es decir, 72 dividido por ppp. La tabla anterior muestra los factores de conversión de cada resolución de pantalla. Por ejemplo, 72 dividido por 96 = 0.75 y 72 dividido por 120 = 0.60. Si multiplica la longitud en píxeles por el factor de conversión correspondiente, obtiene la longitud absoluta en puntos. Por ejemplo, 100 píxeles en una pantalla de 96 ppp equivale a 75 puntos.

Como 72 puntos equivalen a una pulgada, puede obtener la longitud en pulgadas dividiendo la longitud en puntos por 72. Por ejemplo, 100 puntos es igual a 100 dividido por 72 pulgadas = 1,389.

Resolución de pantalla y longitud absoluta

Como decíamos, la longitud absoluta en puntos solamente puede ser igual al número de píxeles utilizados para definir la longitud cuando la resolución de pantalla es 72 ppp. La resolución de pantalla en sistemas Windows no suele ser 72 ppp, sino 96 ppp. Esto significa que el número de puntos de una longitud definida en píxeles en estas pantallas es 75% del número de píxeles. Por ejemplo, en la imagen siguiente de la vista **Diseño** de StyleVision, estos caracteres miden 100 píxeles y 75 puntos respectivamente.



El motivo por el que tienen la misma altura es que la resolución de pantalla en la que aparecen es 96 ppp. Con esta resolución de pantalla, 100 píxeles equivalen a 75 puntos en términos absolutos.

Por tanto, cuando utilice píxeles como unidad de longitud, debe tener en cuenta la resolución de pantalla de su monitor (que suele ser 96 ppp en sistemas Windows) porque determina la longitud absoluta de la longitud definida en píxeles que ve en su pantalla.

Resolución de documentos impresos

Cuando el diseño SPS se transforma en documentos de salida PDF, RTF o Word 2007+, las longitudes en píxeles no absolutas deben convertirse en longitudes absolutas, como puntos o pulgadas (porque los píxeles son una unidad de medida de resolución de pantalla solamente). La pregunta es: ¿qué factor (o resolución de pantalla) usamos para convertir los píxeles en puntos?

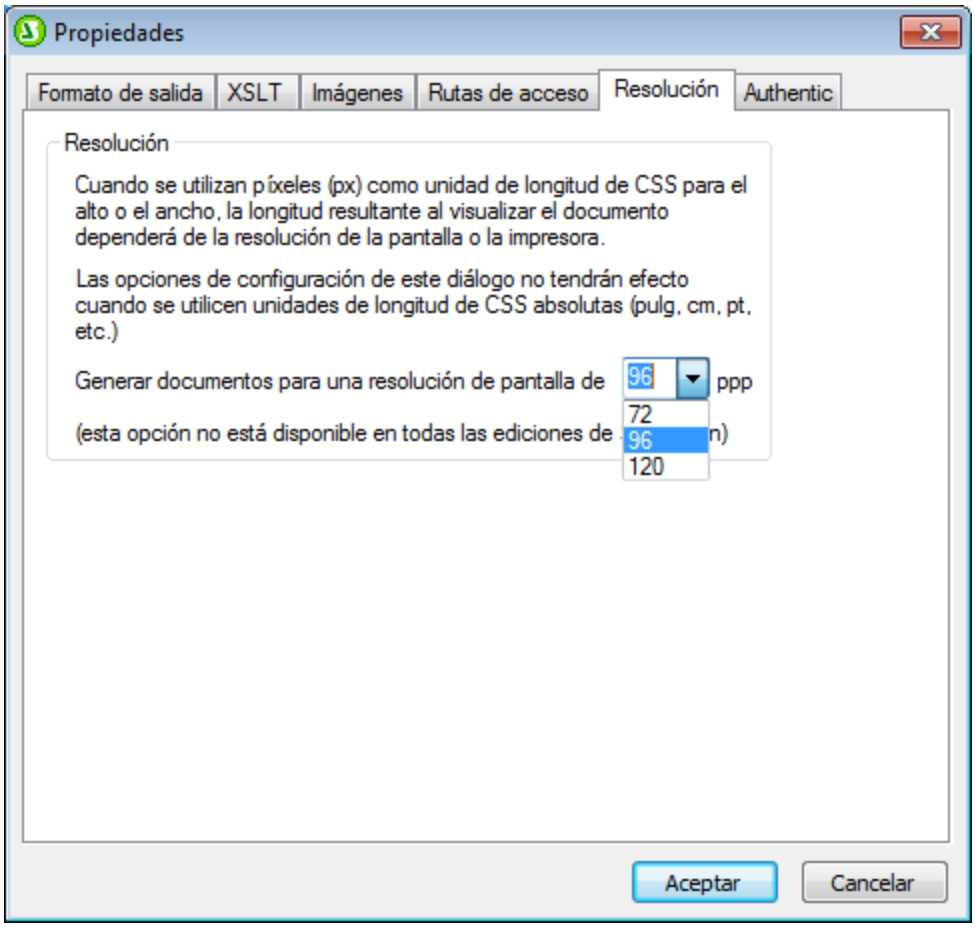
En StyleVision puede seleccionar los ppp de salida de cada diseño SPS. Por ejemplo: si selecciona 96 ppp, un carácter de 100 píxeles aparecerá como un carácter de 75 puntos en el formato impreso (*ver tabla más arriba*). Si selecciona 72 ppp, el mismo carácter se representa en formato de impresión como un carácter de 100 puntos. Este sistema se puede aplicar a todas las longitudes definidas en píxeles.

Nota: la conversión en medidas absolutas no se aplica al formato de salida HTML. Para HTML las unidades en píxeles originales se pasan sin cambios al archivo HTML.

Establecer la resolución del formato de impresión de un SPS

Para establecer la resolución del formato de impresión de un SPS, haga clic en **Archivo | Propiedades**. En el cuadro de diálogo "Propiedades" haga clic en la pestaña *Resolución* (*imagen siguiente*).

 © 2016-2022 Altova GmbH



En esta pestaña puede seleccionar la resolución de impresión necesaria. La tabal siguiente muestra el factor de conversión de cada una de las opciones del cuadro combinado:

Resolución	Píxeles	Puntos	Factor
72 ppp	100	100	1,00
96 ppp	100	75	0,75
120 ppp	100	60	0,60

Si multiplica el número de píxeles por el factor de conversión, obtiene la longitud absoluta en puntos. Recuerde que la conversión en unidades absolutas solo se aplica a formatos de salida de impresión. Los documentos de salida HTML conservan las definiciones de píxeles originales. Además, la resolución elegida en la pestaña *Resolución* del cuadro de diálogo "Propiedades" no cambia la resolución de su monitor.

8.7.5 Marcas de agua

Una marca de agua es una imagen o un texto que aparece en el fondo de cada página de una sección de documento.

Estas son las opciones disponibles:

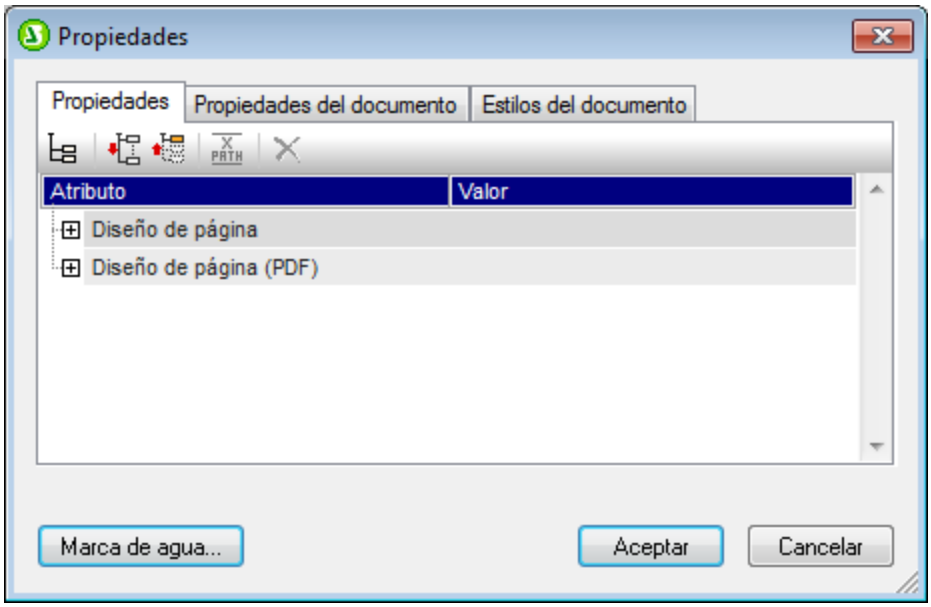
- Debe definir la marca de agua de cada sección de documento por separado. Por tanto, cada sección de documento puede tener una marca de agua diferente.
- Por cada sección de documento puede definir qué condición (con XPath) debe cumplirse para que la marca de agua de la sección se habilite. Por ejemplo, la condición podría ser que el atributo `drafts` de algún elemento tenga el valor `true`. Si la condición no se cumple o si la condición da como resultado `false()`, la marca de agua se deshabilita.
- Una marca de agua puede ser un texto personalizado en StyleVision o una imagen seleccionada por el usuario.
- En el diseño puede especificar la ubicación y el aspecto de las marcas de agua de la página.

Nota: las marcas de agua solamente son compatibles con formatos de salida para medios impresos (es decir, no funcionan con HTML ni con la vista Authentic).

Crear una marca de agua

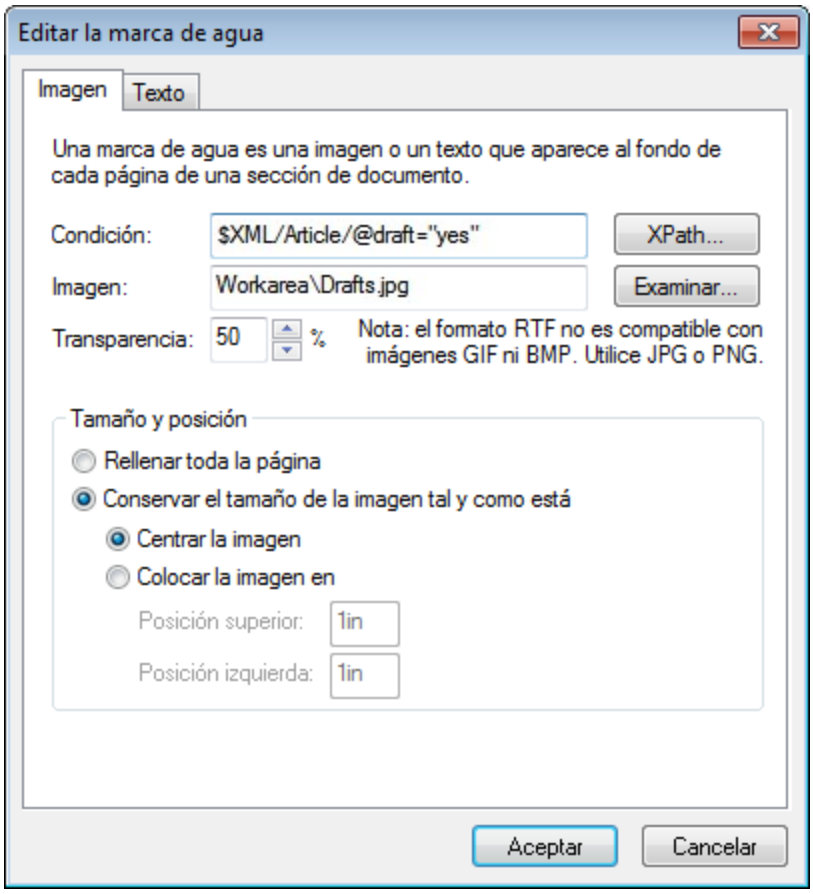
Si quiere puede crear una marca de agua distinta para cada sección de documento. Para crear (o editar) una marca de agua, haga clic en el vínculo `Editar las propiedades...` de la barra de título **Sección inicial del documento** o **Sección del documento** (*imagen siguiente*).

Se abre el cuadro de diálogo "Propiedades" (*imagen siguiente*). Haga clic en el botón **Marca de agua** situado en la parte inferior del diálogo.



Esto abre el cuadro de diálogo "Editar la marca de agua" (*imagen siguiente*), donde puede especificar las propiedades de la marca de agua de la sección de documento seleccionada. Dependiendo de si quiere usar una imagen o simplemente un texto como marca de agua, seleccione la pestaña *Imagen* o *Texto*. Si quiere definir una marca de agua con imagen y texto, rellene ambas pestañas. En ese caso se crearán ambos tipos de marca de agua en el documento de salida.

 © 2016-2022 Altova GmbH



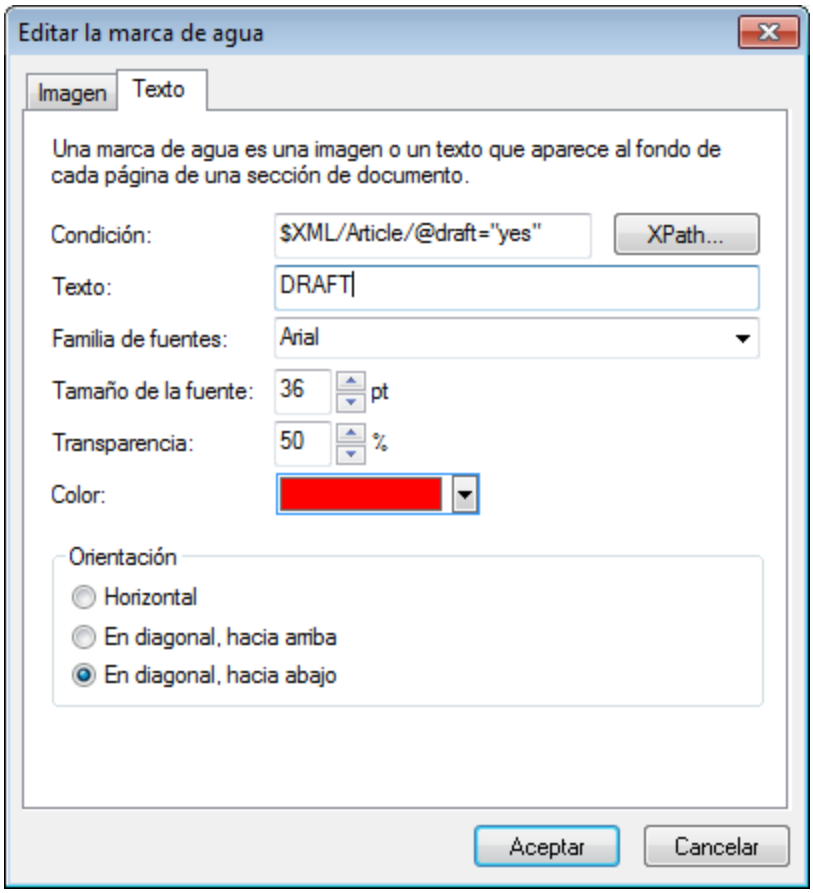
Parámetros de la imagen de la marca de agua

La pestaña *Imagen* (ver más arriba) contiene los parámetros que definen la imagen de la marca de agua para la sección seleccionada. Una expresión XPath especifica la condición que habilita la imagen de marca de agua. Si prefiere no especificar ninguna condición, escriba `true()` en el cuadro de texto *Condición*. Esto hace que el resultado de la condición sea `true()` y se habilite la marca de agua. Si la condición obtiene el resultado `false()` o no se especifica ninguna condición, entonces la imagen de marca de agua no se habilita. El paso siguiente consiste en buscar la imagen (la ruta de acceso del archivo de imagen se inserta en el cuadro de texto *Imagen*.

También puede definir el nivel de transparencia de la imagen, su tamaño y su posición en la página. Si marca la casilla *Rellenar toda la página*, la imagen se expande hasta rellenar una dimensión (a lo alto o a lo ancho). Cuando termine de definir la imagen de marca de agua haga clic en **Aceptar**.

Parámetros del texto de la marca de agua

La pestaña *Texto* (*imagen siguiente*) incluye los parámetros que definen el texto de la marca de agua de la sección seleccionada. Una expresión XPath especifica la condición que habilita el texto de la marca de agua. Si prefiere no especificar ninguna condición, escriba `true()` en el cuadro de texto *Condición*. Esto hace que el resultado de la condición sea `true()` y se habilite la marca de agua. Si la condición obtiene el resultado `false()` o no se especifica ninguna condición, entonces el texto de la marca de agua no se habilita.



El siguiente paso es escribir el texto que quiere usar como marca de agua y especificar su formato (familia de la fuente, tamaño, transparencia y color) y su orientación. Tenga en cuenta que, sea cual sea su orientación, el texto se estira hasta ocupar toda la página. Cuando termine haga clic en **Aceptar**.

Quitar o deshabilitar una marca de agua

Para quitar o deshabilitar una marca de agua abra el cuadro de diálogo "Editar la marca de agua" (*imagen anterior*) y elimine la condición o escriba la expresión XPath `false()` en el cuadro de texto *Condición*. Recuerde que este cuadro de texto está en las dos pestañas del cuadro de diálogo.

© 2016-2022 Altova GmbH

9 Archivos SPS: otras funciones

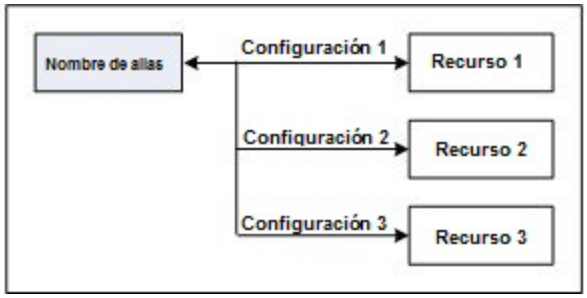
Además de las características de edición de contenidos 134, estructura 219, presentación 374 y características avanzadas 292 descritas hasta ahora, StyleVision también ofrece un variado abanico de funciones adicionales. Estas se describen en los apartados de esta sección:

- Recursos globales 444. Esta característica ofrece gran flexibilidad a la hora de elegir recursos. Por ejemplo, puede asignar un alias a varios recursos (a varios archivos y bases de datos, por ejemplo). Cuando se usa un alias como fuente (XML, XSD, etc.) de un diseño SPS, puede cambiar de recurso eligiendo otro recurso del mismo alias.
- Propiedades de nodos Authentic 462. Los nodos del documento XML puede tener propiedades para la vista Authentic. Puede definir que los nodos sean editables o no editables, que aparezcan con etiquetas de marcado, que muestren información al pasar el cursor del ratón encima de ellos, etc.
- Al hacer clic, reemplazar el nodo primario por 464. El usuario de la vista Authentic puede seleccionar el valor para el nodo primario de un botón o hipervínculo. Puede diseñar el SPS para que la presentación cambie dependiendo de la selección realizada por el usuario de la vista Authentic.
- Validación adicional 467. Puede probar un nodo con ayuda de una expresión XPath que devuelva un valor booleano que indique si los datos introducidos por el usuario en el nodo son válidos o no. Esta prueba se suma a la validación de documentos con respecto al esquema.
- Trabajar con fechas 491. En la vista Authentic puede añadir selectores de fecha gráficos para asegurarse de que las fechas introducidas por el usuario se ajustan al formato correcto de XML Schema. Además, puede manipular y dar formato a las fechas.
- URI de entidades sin analizar 469. Puede almacenar identificadores URI en entidades sin analizar del DTD en el que se basa el documento XML. Así podrá usar imágenes e hipervínculos que apunten a estos URI
- Usar scripts 497. StyleVision contiene un editor de JavaScript donde puede definir funciones JavaScript. Estas funciones se pueden usar después como controladores de eventos en el diseño SPS y se pueden aplicar al documento HTML de salida.
- Importación HTML 502. Puede importar archivos HTML en StyleVision y crear archivos XML, XSD y SPS a partir de ellos.
- Crear archivos a partir de XSLT 471. En StyleVision puede crear diseños SPS a partir de hojas de estilos XSLT-para-HTML o XSLT-para-FO. La estructura y los estilos de la plantilla XSLT se recrearán en el SPS. Después puede modificar los componentes de diseño y añadir contenido y formato en el diseño SPS.

9.1 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, puede generar un archivo XML a partir de una asignación de datos de Altova MapForce y utilizarla en StyleVision como archivo XML de trabajo en una hoja de estilos SPS.

El uso de los recursos globales de Altova conlleva dos procesos:

- Definir los recursos globales 444: 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 455: 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 StyleVision.

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.

9.1.1 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:

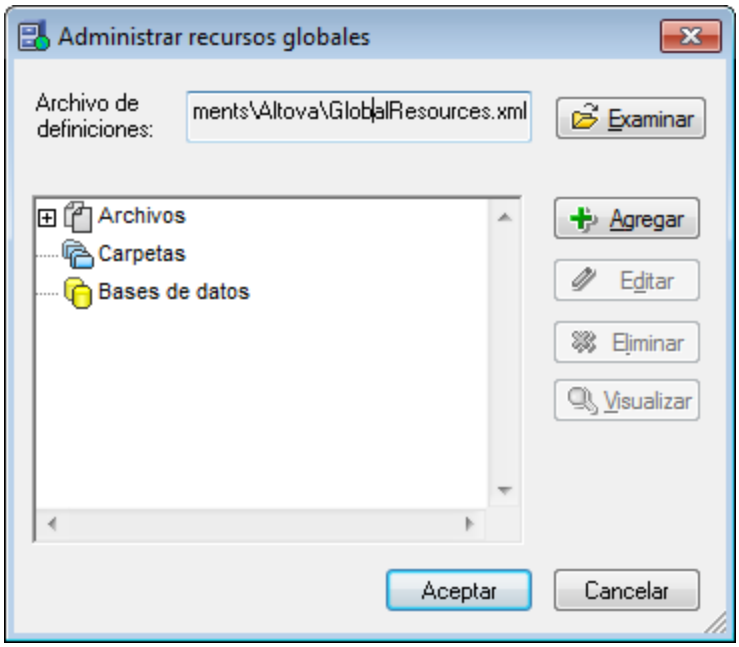
- Seleccionando el comando de menú **Herramientas | Recursos globales**.

 © 2016-2022 Altova GmbH

- 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*[447], *carpetas*[452] *y bases de datos*[444] *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.

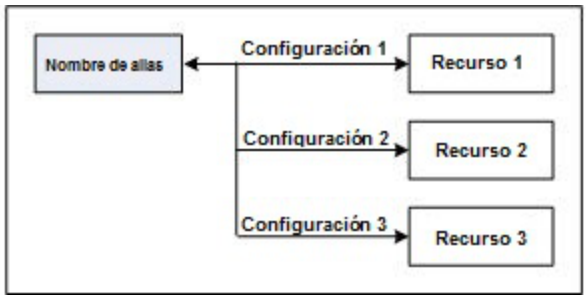
Para **editar un recurso global**, selecciónelo y haga clic en **Editar**. En el cuadro de diálogo que aparece puede realizar todos los cambios necesarios (*consulte los subapartados archivos*[447], *carpetas*[452] *y bases de datos*[444] *para más información*).

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*).



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

 © 2016-2022 Altova GmbH

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.

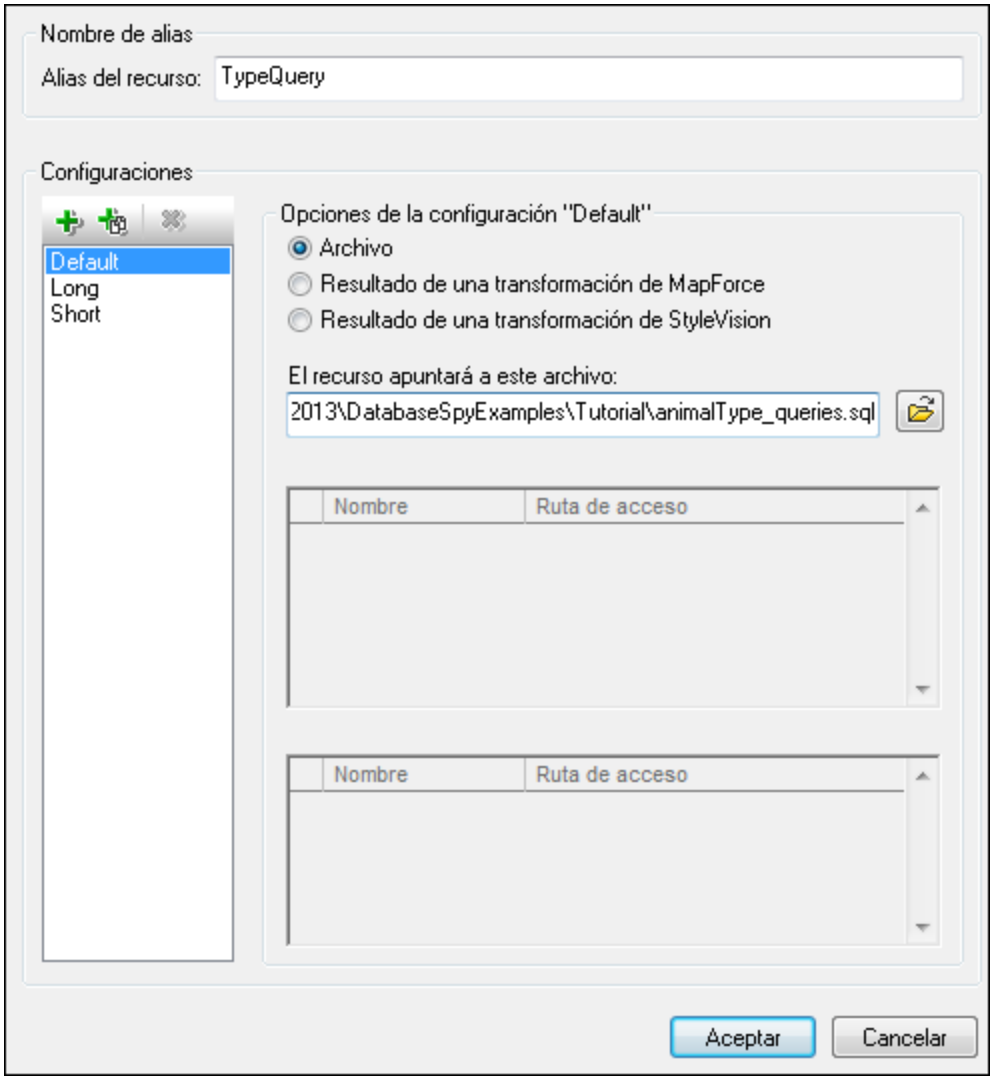
9.1.1.1 Archivos

El cuadro de diálogo "Recurso global" para archivos (*imagen siguiente*) se abre con el comando **Agregar | Archivos** del cuadro de diálogo Administrar recursos globales 444. 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) 444.

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.

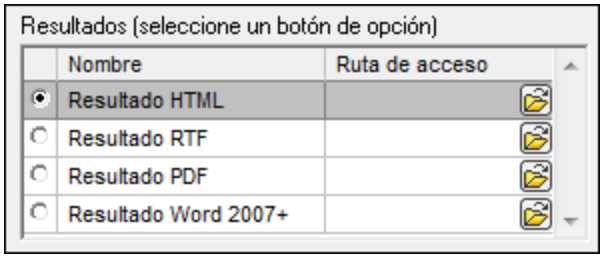
Abrir: examina el sistema de archivos para buscar el archivo que se debe usar para crear el recurso global.

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).



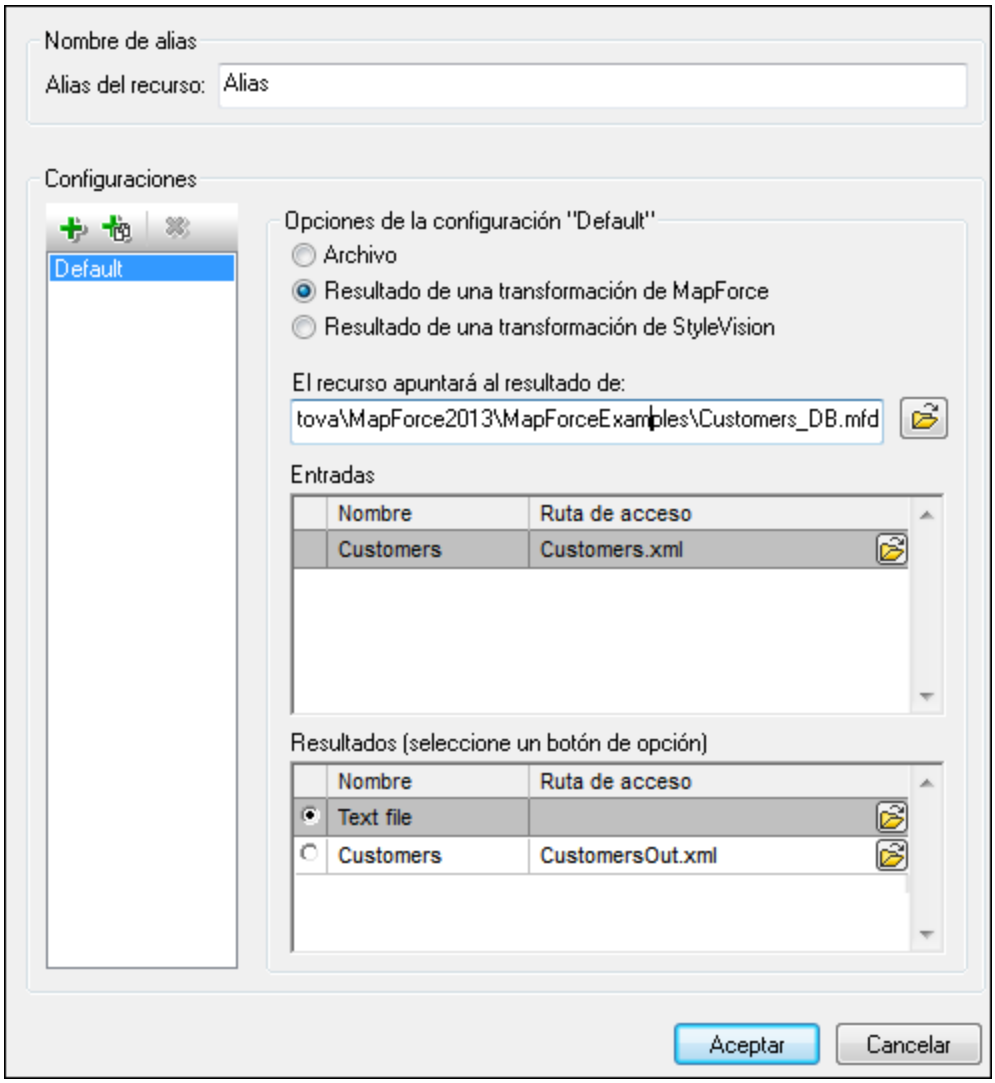
Seleccione una opción (por ejemplo, en la imagen anterior se seleccionó el resultado en formato HTML). Si el recurso es el resultado de una transformación, el resultado se puede guardar como archivo o como recurso global. Haga clic en el icono y seleccione **Examinar...** (si desea guardar el resultado 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 444.

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.

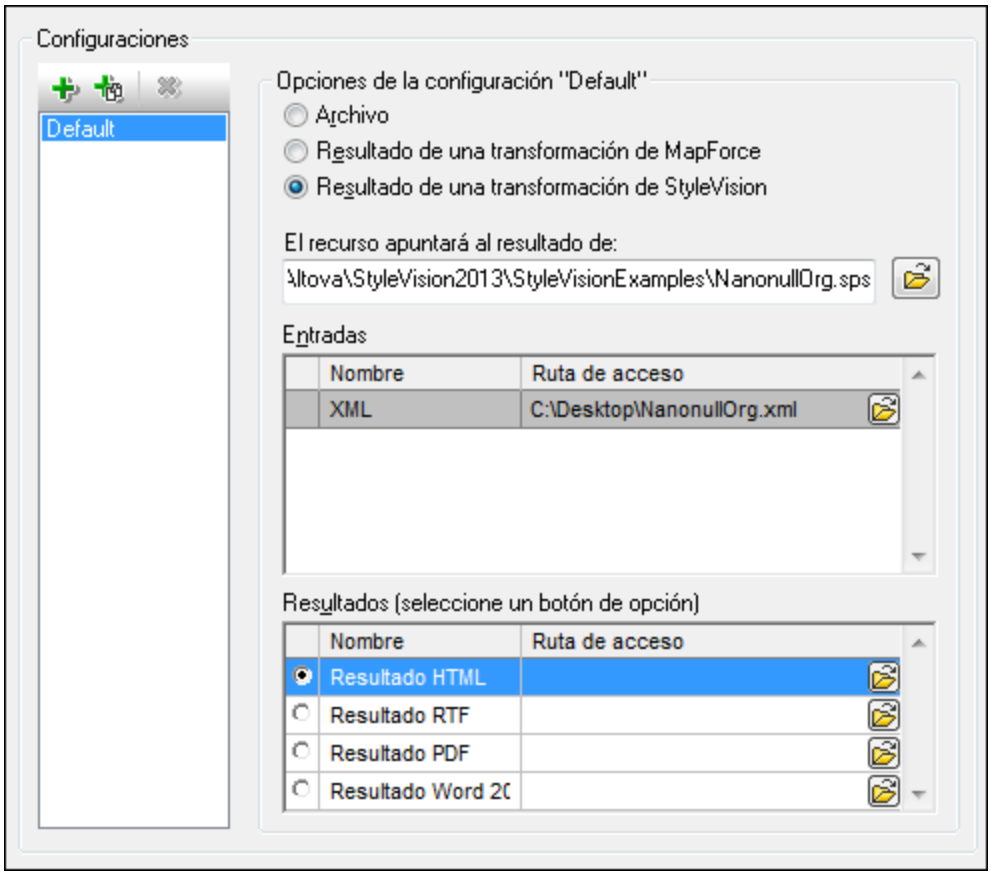
Nota: la ventaja de usar esta opción (resultado de una transformación de MapForce) es que la transformación se ejecuta al invocarse el recurso global. Esto significa que el recurso global contendrá los datos más recientes (de los archivos de entrada).

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*).

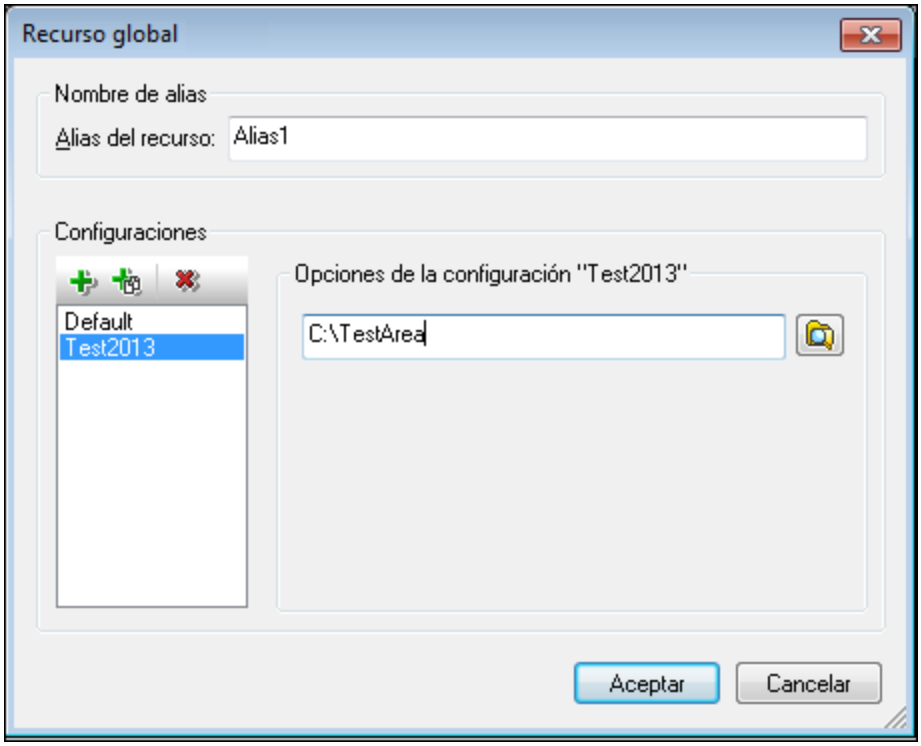
- ***Archivos de entrada***. El archivo de entrada puede venir dado por el archivo SPS. Si es así, el archivo de entrada aparecerá automáticamente en el panel *Inputs*. En este panel puede elegir el archivo de entrada que quiera. Si el panel está vacío, entonces añada un archivo.
- ***Archivos de resultados.*** 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.

9.1.1.2 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.

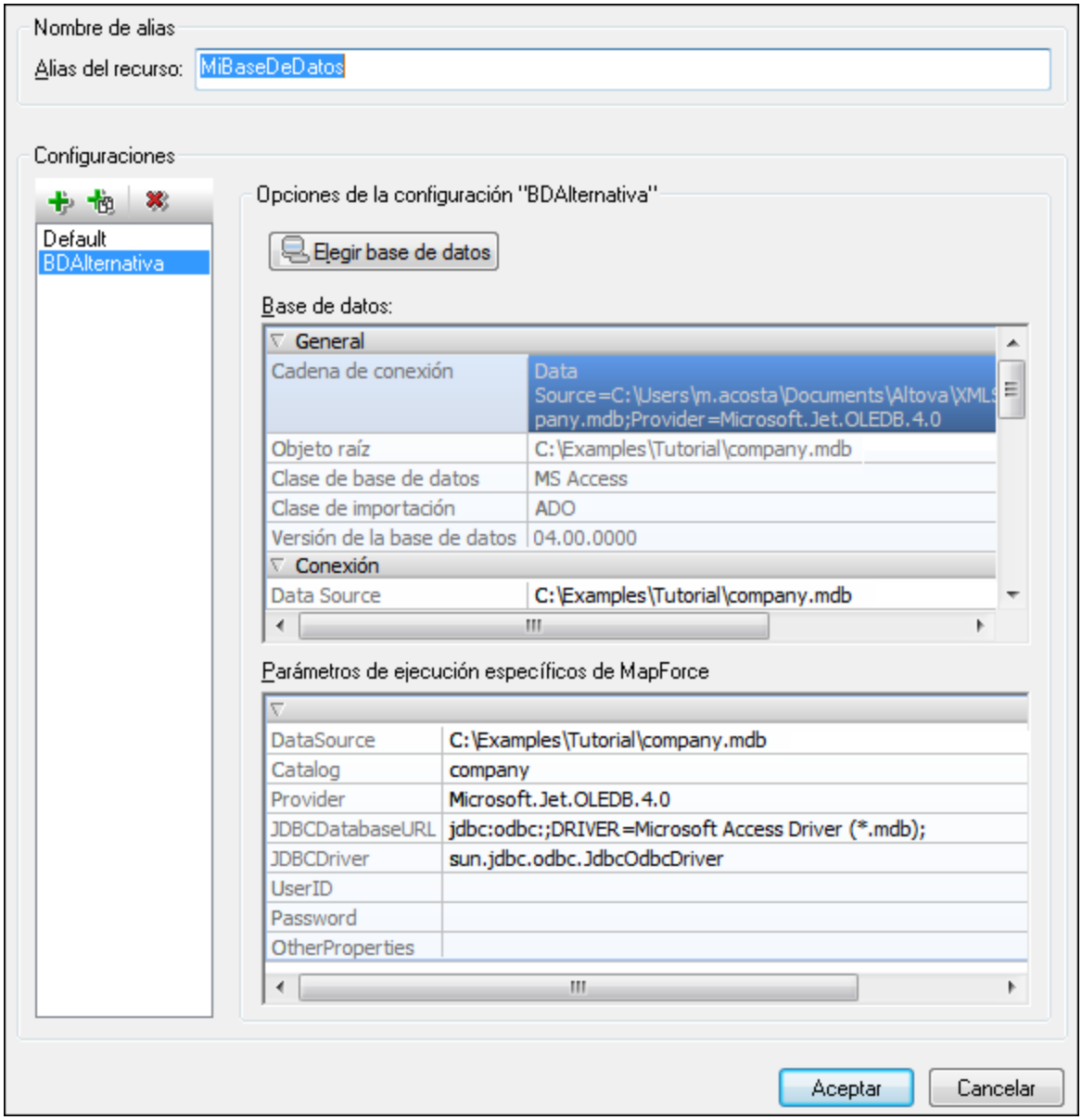
Definir el alias

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*.

9.1.1.3 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.

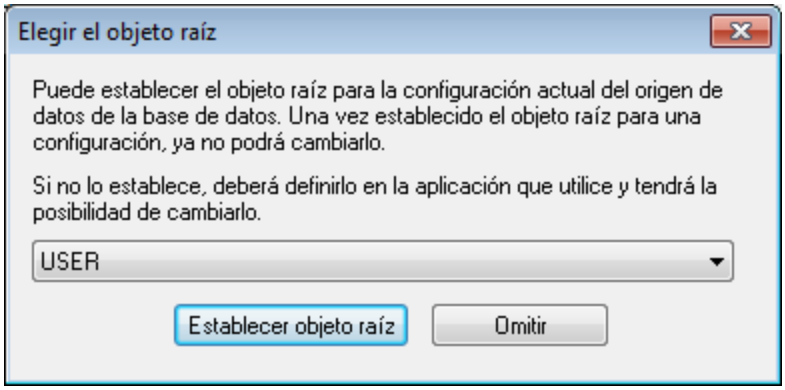
Eliminar: elimina la configuración seleccionada.

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 523. Si ya se estableció una conexión a una base de datos desde StyleVision, 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".

9.1.2 Usar recursos globales

Hay varios tipos de recursos globales (de tipo archivo, carpeta y base de datos). Dependiendo del caso, en StyleVision puede usar un tipo de recurso global u otro. Por ejemplo, puede usar recursos globales de tipo archivo o carpeta para un archivo XML de trabajo o para un archivo CSS. O puede usar un recurso de tipo base

de datos para crear una SPS nueva basada en una BD. Algunos casos de uso de los recursos globales en StyleVision se describen aquí: Archivos y carpetas 456 y Bases de datos 459.

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:

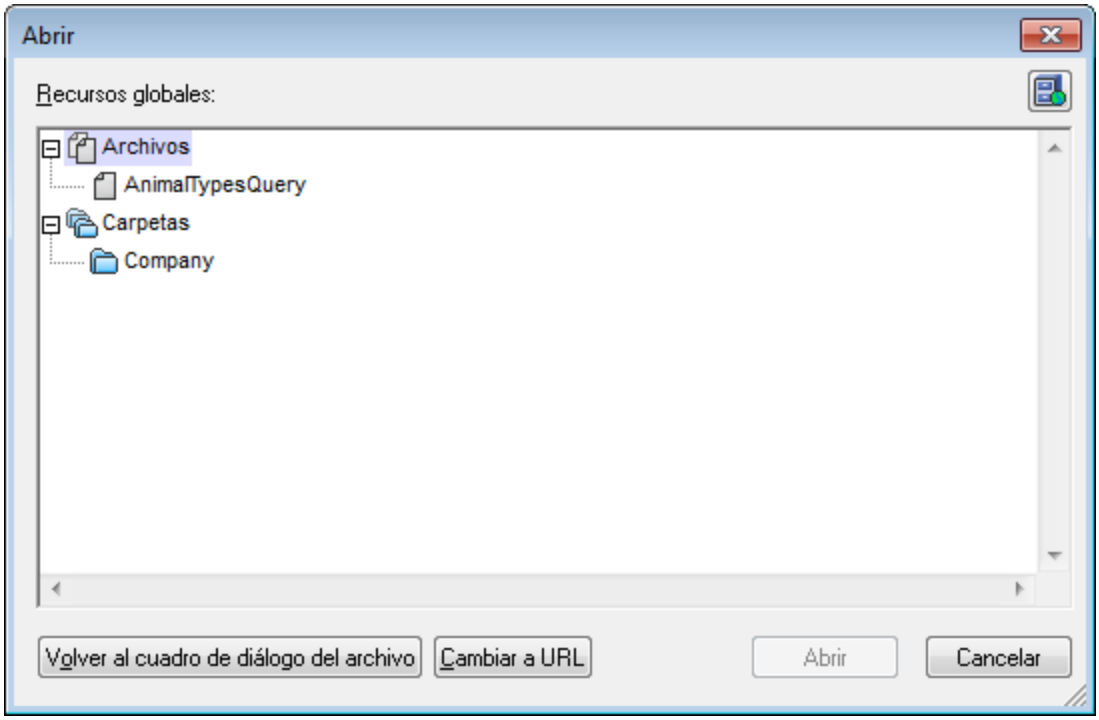
- El *archivo XML de recursos globales* seleccionado en el cuadro de diálogo Recurso global 444. Las definiciones de recursos globales presentes en el archivo XML de recursos globales están a disposición de todos los archivos que estén abiertos en la aplicación. Las únicas definiciones disponibles son 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** 870 o en la barra de herramientas Recursos globales 759. 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.

9.1.2.1 Asignar archivos y carpetas

En este apartado explicamos cómo se asignan recursos globales de tipo archivo y carpeta. Los recursos globales de tipo archivo no se asignan igual que los recursos globales de tipo carpeta. En los casos de uso 457 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 444.

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 444.

Casos de uso

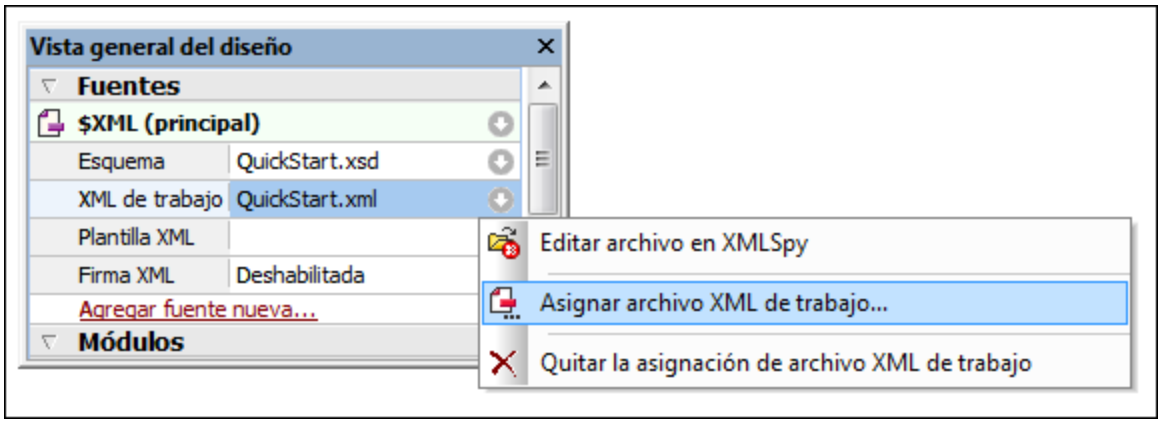
Puede usar recursos globales de tipo archivo y carpeta en estos casos:

- Al agregar y modificar fuentes de esquema y archivos XML de trabajo 456 y plantillas XML 456
- Al guardar archivos como recurso global 458
- Al agregar módulos y archivos CSS 458
- Al agregar recursos globales a un proyecto 459

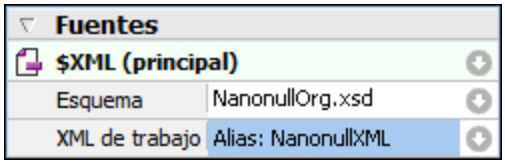
Esquemas, archivos XML de trabajo y plantillas XML

En la barra lateral Vista general del diseño (*imagen siguiente*) los menús contextuales del esquema, del archivo XML de trabajo, de la plantilla XML incluye una opción que abre el cuadro de diálogo "Abrir", en el que puede asignar el esquema 45 o archivo XML de trabajo 45 a través de un recurso global. Al hacer clic en el

botón Cambiar a recursos globales, aparece un diálogo con todos los recursos globales de tipo archivo definidos en el archivo XML de recursos globales que está activo en StyleVision. (Para más información sobre cómo configurar el archivo XML de recursos globales activo, consulte el apartado Definir recursos globales 444.)



Si selecciona un recurso global como fuente de archivo, este aparece en la entrada correspondiente de la barra lateral Vista general del diseño (*imagen siguiente*).



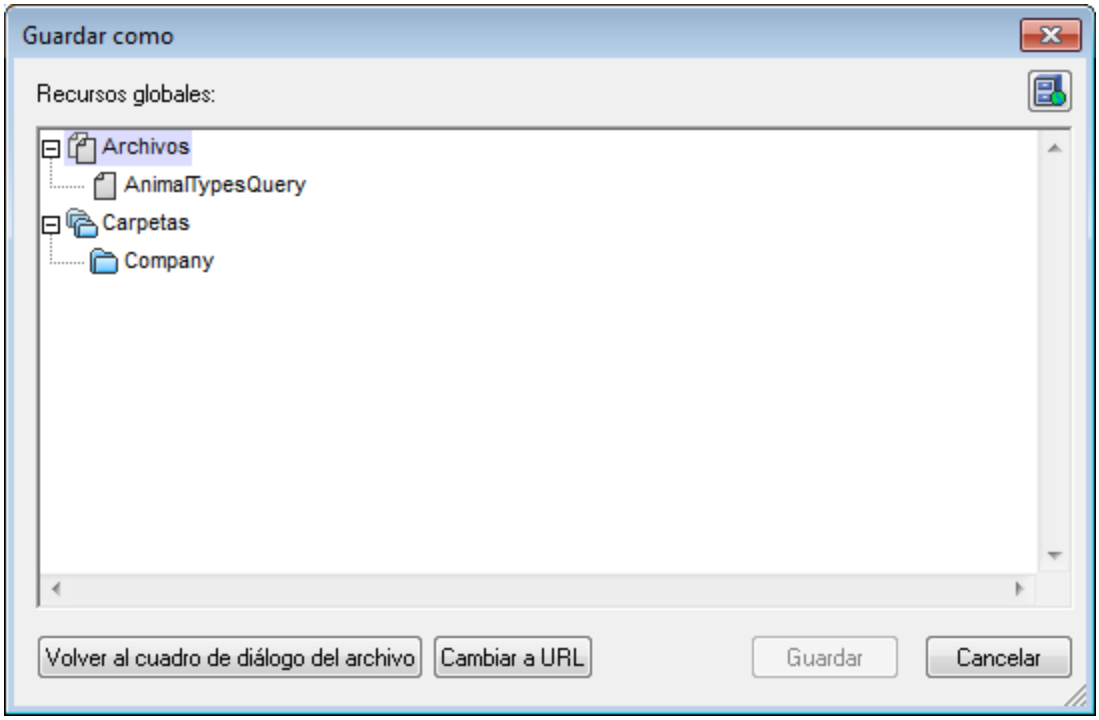
Agregar módulos y archivos CSS desde un recurso global

En la barra lateral Vista general del diseño 45 los comandos **Agregar módulo nuevo** y **Agregar archivo CSS** nuevo abren el cuadro de diálogo "Abrir". En este cuadro de diálogo puede pulsar el botón **Cambiar a recursos globales** para seleccionar un recurso global. Después puede cambiar de módulo y de archivo CSS con solo cambiar la configuración activa.

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.

 © 2016-2022 Altova GmbH



Seleccione un alias y pulse el botón **Guardar**. Si el alias es un alias de archivo 447 el archivo se guarda directamente. Si el alias es un alias de carpeta 452, 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 460.

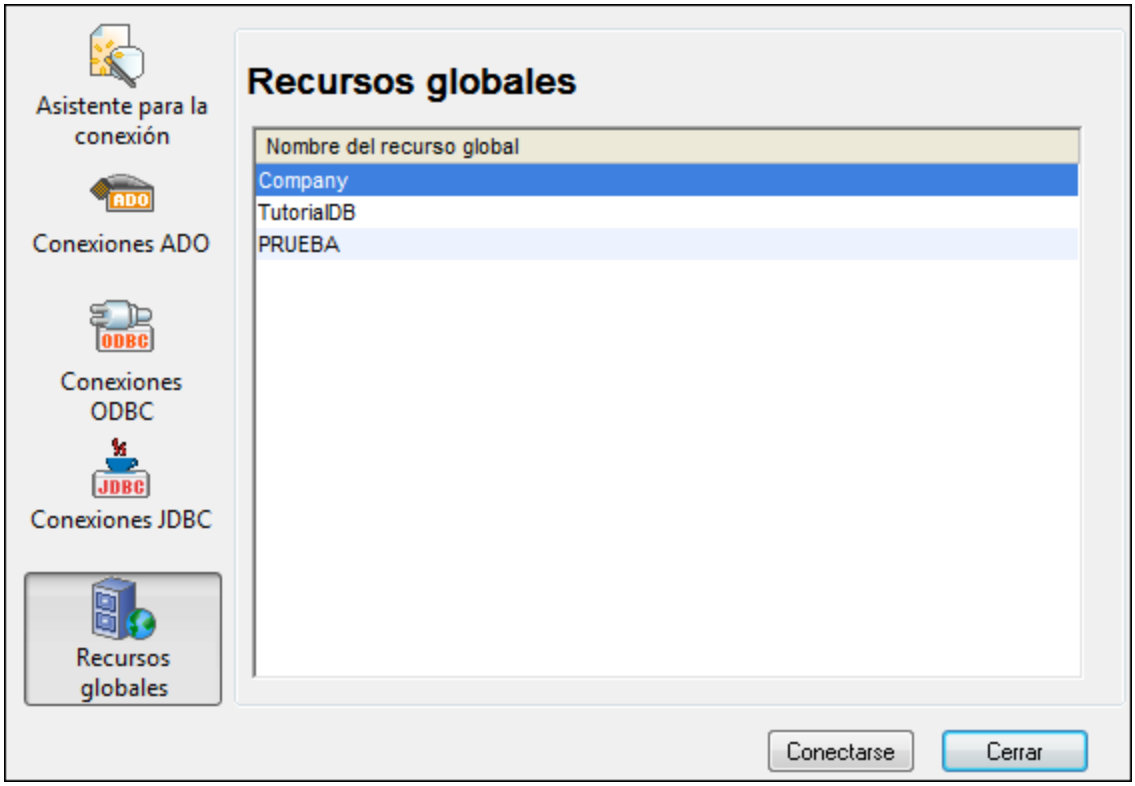
Nota: cada configuración apunta a una ubicación de archivo concreta, que se indica en la definición de la configuración. Si el archivo que desea guardar como recurso global no tiene la misma extensión de archivo que el archivo guardado en la ubicación de archivo de la configuración, entonces puede haber errores de edición y validación al abrir este recurso global en StyleVision. El motivo es que, al abrir el archivo, StyleVision asume el tipo de archivo indicado en la definición de la configuración.

Uso de recursos globales en proyectos

También puede agregar recursos globales al proyecto activo con el comando **Proyecto | Agregar recurso global al proyecto**. Como resultado aparece un cuadro de diálogo con todos los recursos globales de tipo archivo del archivo XML de recursos globales 444 activo. Seleccione un recurso global y haga clic en **Aceptar** para agregarlo al proyecto. El recurso global aparece en la barra lateral Proyecto y puede utilizarse como cualquier otro archivo.

9.1.2.2 Asignar bases de datos

Si crea un archivo SPS a partir de una BD con el comando **Archivo | Nuevo a partir de una base de datos**, puede elegir usar un recurso global de BD (*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 444. 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.

9.1.2.3 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.

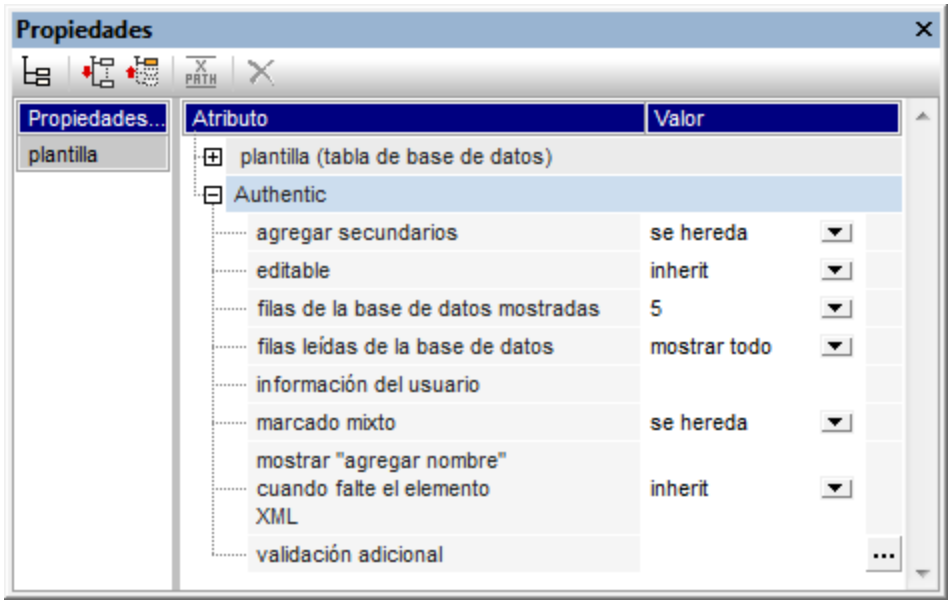
 © 2016-2022 Altova GmbH

De este manera, al cambiar de configuración activa, puede cambiar los archivos de origen que se asignaron a través de recursos globales.

9.2 Propiedades de nodos Authentic

Gracias a las propiedades de nodo Authentic podrá **representar un nodo en la vista Authentic**. Por ejemplo, el nodo se puede representar con las etiquetas de marcado XML, puede ser no editable y puede mostrar información al usuario cuando se pasa el cursor del ratón por encima. Puede definir propiedades de nodo Authentic en varios componentes SPS y las propiedades disponibles en cada caso dependen del tipo de componente seleccionado.

Seleccione el componente del diseño al que quiere asignar propiedades de nodo Authentic. Después, en la ventana Propiedades (grupo de propiedades *Authentic*), defina las propiedades de nodo Authentic necesarias (*imagen siguiente*). Otra opción es hacer clic con el botón derecho en la plantilla de nodo y seleccionar **Editar propiedades de Authentic** en el menú contextual.



Definir las propiedades de Authentic

A continuación enumeramos las propiedades de Authentic con las que puede controlar el comportamiento de los nodos que se representan en la vista Authentic.

agregar secundarios
Disponible si el nodo seleccionado es un elemento. Sirve para definir qué elementos secundarios del elemento seleccionado se insertan cuando se añade el elemento seleccionado en la vista Authentic. Los valores de esta propiedad son: `todos`, `obligatorios`, `ninguno`.

editable
Define si el nodo se puede editar o no. Por defecto, todos los nodos son editables. Esta propiedad está disponible si el nodo seleccionado es un elemento, atributo o contenido. Los resultados de cálculos automáticos no se pueden editar porque el valor se calcula con la expresión XPath insertada en el cálculo automático.

filas de la base de datos mostradas y leídas
Las propiedades `filas de la base de datos mostradas` y `filas leídas de la base de datos` definen respectivamente cuántas filas de la BD se mostrarán en la vista Authentic y cuántos registros de BD se deben

 © 2016-2022 Altova GmbH

cargar. Por tanto, la propiedad filas `leídas de la base de datos` permite acelerar la carga y la presentación de los datos.

información del usuario
El texto que introduzca aquí aparece al pasar el puntero del ratón por encima del nodo seleccionado. Esta propiedad está disponible si el nodo seleccionado es un elemento, atributo, contenido o cálculo automático. Si tanto el nodo de elemento/atributo como su contenido tienen definida esta propiedad, entonces se usa la información de usuario del nodo de contenido.

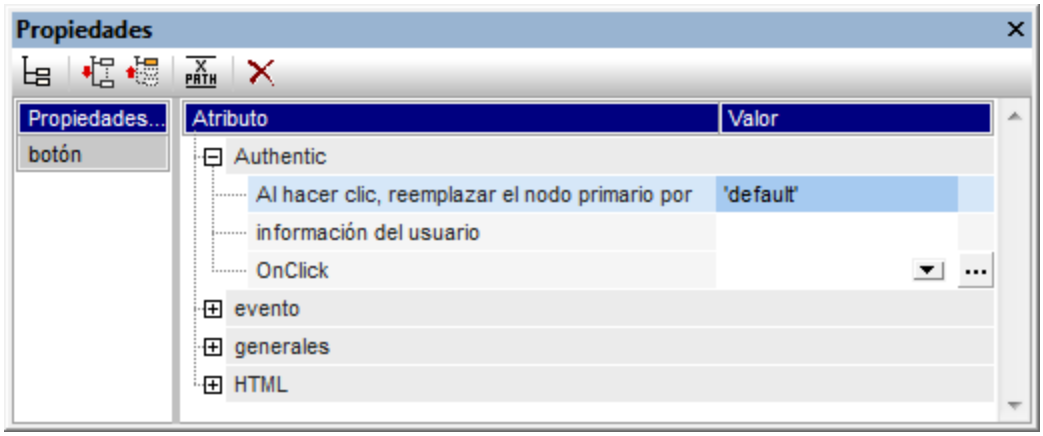
marcado mixto
Disponible si el nodo seleccionado es un elemento o un atributo. Permite definir el marcado de los nodos cuando se elige el modo *Marcado mixto* en la vista Authentic. Los valores de esta propiedad son: `marcado grande` (etiquetas con el nombre del nodo), `marcado pequeño` (etiquetas sin el nombre de nodo) y `ocultar marcado`.

mostrar "agregar nombre" cuando falte el elemento XML
Determina si la vista Authentic muestra o no el aviso *`Agregar [nombre de elemento/atributo]`* cuando falta el elemento o atributo seleccionado. El aviso aparece por defecto en la vista Authentic. Esta propiedad está disponible cuando el nodo seleccionado es un elemento o un atributo.

validación adicional
Además de validar el documento XML con un esquema, StyleVision ofrece una validación adicional para los nodos y los cálculos automáticos. Puede usar una expresión XPath para definir el rango de validez del contenido XML del nodo o del cálculo automático. Si el valor XML del nodo no es válido, se le comunica al usuario de la vista Authentic por medio de un mensaje de error cuando se valida el documento XML con **F8**. Para más información consulte el apartado Validación adicional 467.

9.3 Al hacer clic, reemplazar el nodo primario por

La propiedad `Al hacer clic, reemplazar el nodo primario por` está en el grupo de propiedades *Authentic* de la ventana Propiedades (*imagen siguiente*). Esta propiedad permite definir el contenido de texto del nodo primario de un botón 197 o hipervínculo 366.



Cuando el usuario de la vista Authentic hace clic en un botón o hipervínculo, el texto especificado en esta propiedad se insertará como contenido XML del nodo primario del botón o hipervínculo. Es decir, gracias a esta propiedad, el diseñador del SPS puede dar al usuario de la vista Authentic la opción de realizar ciertas selecciones (haciendo clic en un botón o en un hipervínculo) que modificarán la presentación del documento XML en la vista Authentic.

Instrucciones de uso

Siga estos pasos para usar esta característica en su diseño:

1. En la vista **Diseño** del documento SPS inserte el componente de diseño (el botón o el hipervínculo) dentro del nodo cuyo contenido de texto debería modificar el usuario de la vista Authentic.
2. Seleccione el componente de diseño recién insertado y en la ventana Propiedades escriba el valor de la propiedad `Al hacer clic, reemplazar el nodo primario por` (*imagen anterior*). El valor puede ser una cadena de texto o una expresión XPath (haga clic en el icono **XPath** de la barra de herramientas y después inserte la expresión XPath). Por ejemplo, en la imagen anterior, el valor es la expresión XPath `'default'`. Cuando el usuario de la vista Authentic haga clic en el componente de diseño en la vista Authentic, el texto `default` se insertará como contenido de texto del nodo primario del componente de diseño. El mismo resultado se obtiene introduciendo la cadena de texto `default` en lugar de la expresión XPath.
3. Ahora que conoce qué contenido tendrá el nodo primario cuando el usuario de la vista Authentic haga clic en el componente de diseño, puede definir un procesamiento basado en el contenido de texto del nodo primario. Por ejemplo, con ayuda de una condición 303 puede especificar que, si el contenido del nodo primario es `default`, se aplique cierto formato predeterminado o que, si el contenido es `ocultar`, se oculte el texto de un nodo. Gracias a esta característica puede ahorrarse el uso y la implementación de scripts 497.

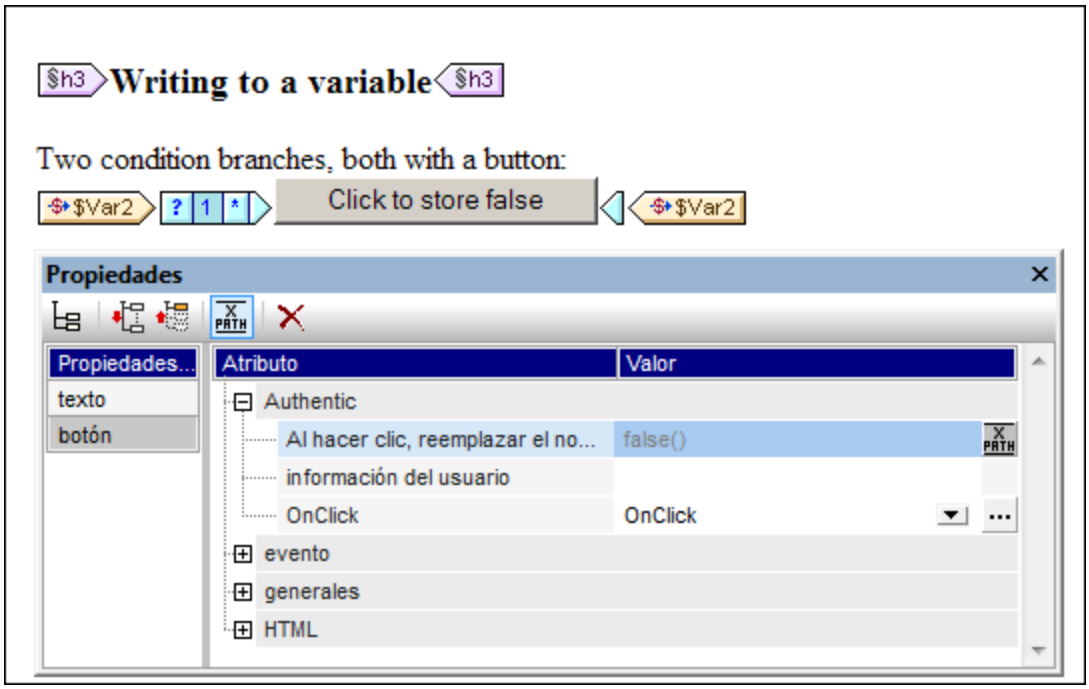
Nota: para que esta propiedad funcione bien, el esquema fuente debe permitir que el nodo primario tenga contenido de texto.

 © 2016-2022 Altova GmbH

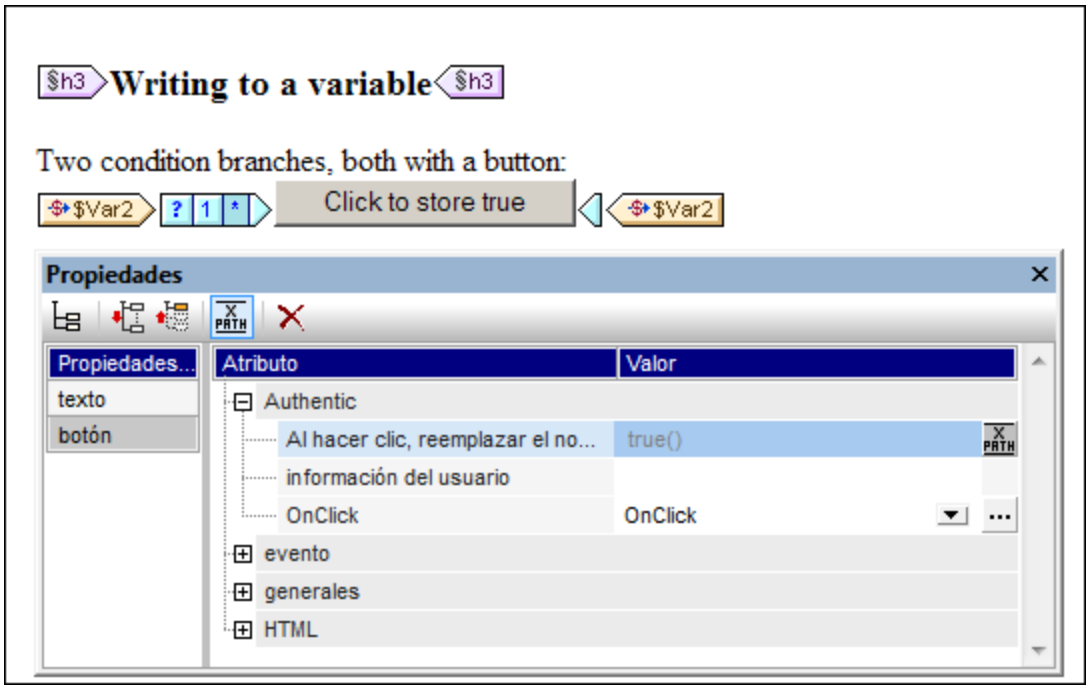
Usar la propiedad con variables

El nodo primario también puede ser una variable 327 definida en un nodo antecesor. El valor de la variable lo puede definir el diseñador del SPS con la propiedad `Al hacer clic, reemplazar el nodo primario por`. Cuando el usuario de la vista Authentic haga clic en el componente de diseño (botón 197 o hipervínculo 366), el valor definido para la propiedad `Al hacer clic, reemplazar el nodo primario por` del componente de diseño se asigna a la variable. Ahora puede definir opciones de procesamiento alternativas dependiendo del valor de la variable.

Por ejemplo: cuando el valor de la variable `Var2` sea el booleano `true()`, aparece un botón que permite cambiar el valor de la variable por el valor booleano `false()` (*imagen siguiente*). Este procesamiento se especifica en la primera rama de la condición 303 que aparece en la imagen siguiente.



Por su parte, la rama *Otherwise* de la condición indica un procesamiento que muestra un botón para cambiar el valor de la variable por el valor booleano `true()` (*imagen siguiente*). Ahora puede definir otras reglas de procesamiento para cuando la variable `Var2` tenga el valor `true()` y para cuando tenga el valor `false()`.



Resumiendo: gracias a esta característica el diseñador del SPS puede dar la opción al usuario de la vista Authentic de seleccionar varias opciones que después produzcan un procesamiento concreto del documento.

© 2016-2022 Altova GmbH

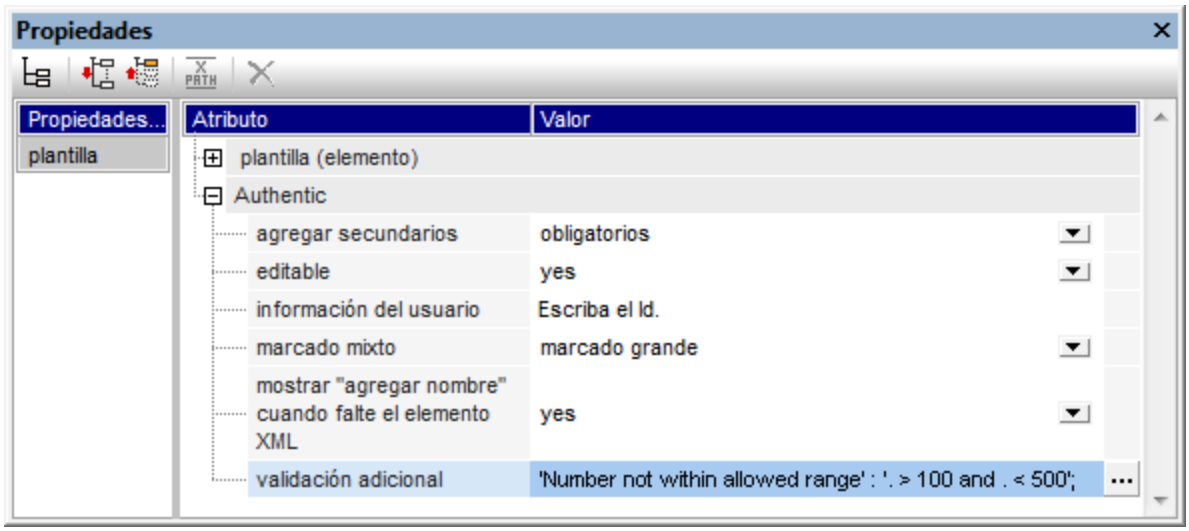
9.4 Validación adicional

La propiedad `validación adicional` está disponible si el componente seleccionado es un nodo de elemento/atributo, contenido (marcador de posición `(contenido)`), un dispositivo de entrada de datos o un cálculo automático. El valor de la propiedad `validación adicional` es una expresión XPath que define el rango de validez del contenido XML del nodo o del cálculo automático. Si el valor XML del nodo / cálculo automático está dentro del rango, entonces el nodo / cálculo automático es válido. Si el valor XML no es válido (está fuera del rango), se le comunica al usuario de la vista Authentic por medio de un mensaje de error cuando se ejecute la función de validación (**F8**). El mensaje de error es el texto que introduzca el diseñador del SPS en el campo *Mensaje de error si resultado = "false"* del cuadro de diálogo "Agregar validación".

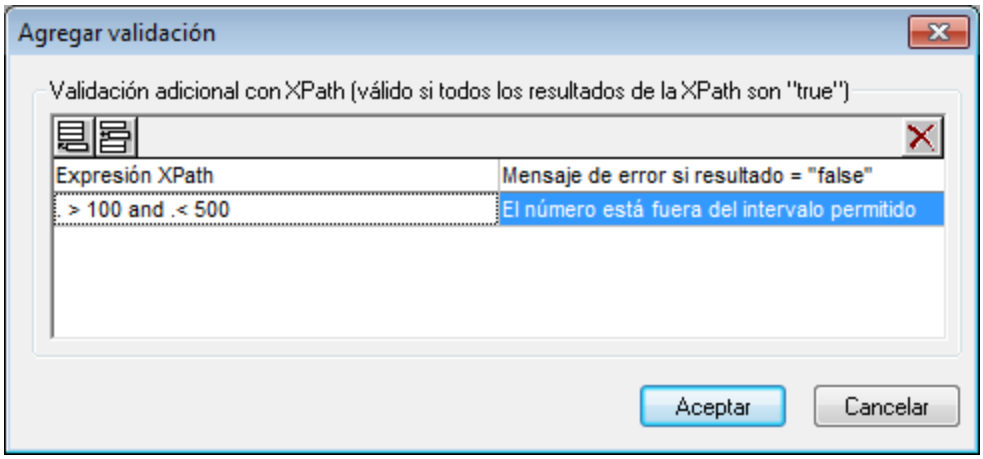
Configurar la validación adicional

Siga estas instrucciones para establecer una regla de validación adicional:

1. Seleccione el componente para el que desea definir la regla de validación adicional.
2. En la ventana Propiedades expanda el grupo de propiedades *Authentic*. Haga clic en el botón **Edición** ... de la propiedad `validación adicional` (*imagen siguiente*).



3. Se abre el cuadro de diálogo "Agregar validación" (*imagen siguiente*), haga clic el botón **Agregar** de la esquina superior izquierda para añadir una regla de validación adicional.



4. En la columna *Expresión XPath* escriba la expresión que define el rango de validez de los datos XML para el componente.
5. En la columna *Mensaje de error si resultado = "false"* escriba el mensaje de error que debe aparecer si los datos no son válidos.
6. Para terminar haga clic en **Aceptar**.

© 2016-2022 Altova GmbH

9.5 URI de entidades sin analizar

Si trabaja con una DTD que tiene declarada una entidad sin analizar, puede usar el URI asociado a la entidad para definir el destino de imágenes e hipervínculos en el diseño SPS. Esto permite usar el mismo URI varias veces en el diseño SPS. StyleVision aprovecha la función XSLT `unparsed-entity-uri` para pasar el URI de la entidad sin analizar desde la DTD hasta los documentos de salida. Esto significa que el URI está disponible en los archivos de salida (HTML, RTF), pero no en la vista Authentic. Las URIs de entidades sin analizar no se procesan para la salida en texto.

Para que esta característica funcione correctamente es necesario que la DTD, el documento XML y el SPS estén bien editados:

1. En la DTD deben estar declaradas las entidades sin analizar 469 con (i) el URI y (ii) la notación (que le dice a StyleVision qué tipo de recurso es la entidad).
2. En el documento XML se debe hacer referencia 469 a la entidad sin analizar. Esto se hace dando los nombres de las entidades sin analizar pertinentes.
3. En el diseño SPS puede usar las entidades sin analizar para apuntar a imágenes 183 e hipervínculos 366. Esto se hace accediendo a los valores de nodo dinámico pertinentes como entidades sin analizar 469.

Declarar entidades sin analizar y hacerles referencia

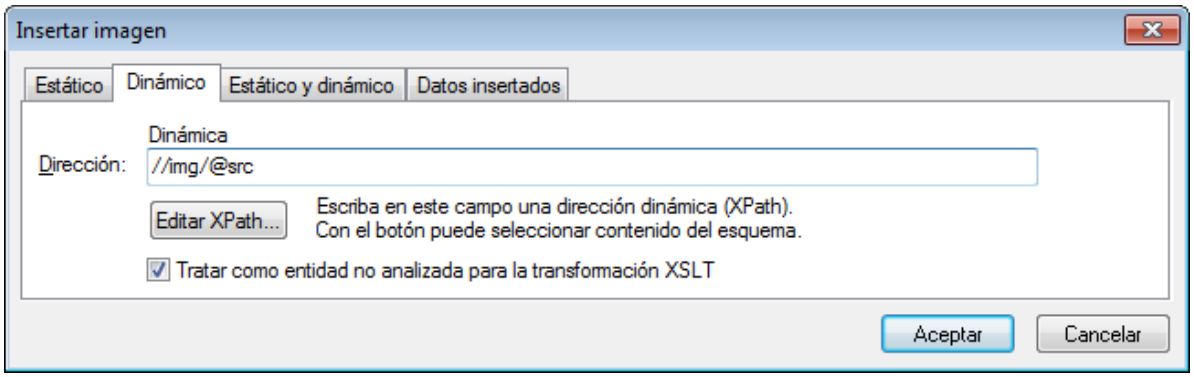
A continuación aparece un fragmento de un documento XML que incluye un subconjunto DTD interno que declara dos entidades sin analizar: una con la notación `GIF` (lo cual indica que se trata de una imagen GIF) y otra con la notación `LNK` (lo cual indica que se trata de un vínculo). Los nodos `img/@src` y `link/@href` del código XML hacen referencia a las entidades sin analizar al dar su nombre.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE document SYSTEM "UEURIDoc.dtd" [
<!ENTITY Picture SYSTEM "nanonull.gif" NDATA GIF>
<!ENTITY AltovaURI SYSTEM "http://www.altova.com" NDATA LNK>
]>
<document>
   <header>Example of How to Use Unparsed Entity URIs</header>
   <para>...</para>
   <img src="Picture"/>
   <link href="AltovaURI">Link to the Altova Website.</link>
</document>
```

Imágenes e hipervínculos del SPS que utilizan entidades sin analizar

Las imágenes y los hipervínculos del diseño SPS que hacen referencia a los URI de entidades sin analizar se usan de la siguiente manera:

1. Primero debe insertar la imagen o el hipervínculo desde el menú **Insertar**.
2. Después en el cuadro de diálogo de inserción (*imagen siguiente*), seleccione la pestaña *Dinámico* y escriba la expresión XPath que selecciona el nodo que contiene el nombre de la entidad sin analizar. En el documento XML del ejemplo, estos nodos serían `//img/@src` y `//link/@href`.



3. Después marque la casilla *Tratar como entidad no analizada* situada en la parte inferior del cuadro de diálogo. Esto hace que el contenido del nodo seleccionado se lea como entidad sin analizar. Si se declaró una entidad no analizada con ese nombre, el URI asociado a la entidad se utiliza para encontrar el recurso (imagen o hipervínculo).

Cuando se procese la hoja de estilos, el URI asociado con el nombre de entidad se reemplaza con el nombre de entidad.

Nota: si el URI es un URI relativo, el procesador XSLT lo amplía a un URI absoluto aplicando el URI base de la DTD. Por ejemplo, si la entidad sin analizar está asociada al URI relativo "`nanonull.gif`", este URI se ampliaría a `file:///c:/Carpeta/nanonull.gif`, siendo `Carpeta` el directorio donde está la DTD.

© 2016-2022 Altova GmbH

9.6 Diseño nuevo a partir de archivos XSLT, XSL-FO y FO

El diseño SPS puede estar basado en archivos XSLT diseñados para generar documentos HTML o en archivos XSLT con comandos XSL-FO para generar archivos PDF o FO. Esto significa que no es necesario diseñar documentos SPS desde cero, sino que pueden partir de archivos XSLT ya existentes.

Pasos para crear un diseño SPS a partir de una hoja de estilos XSLT

Siga estas instrucciones para crear un diseño SPS a partir de un archivo XSLT, XSLT-para-FO o FO:

1. Seleccione el comando **Archivo | Nuevo | Nuevo a partir de un archivo XSLT, XSL-FO o FO**.
2. Se abre el cuadro de diálogo "Abrir". Seleccione el archivo XSLT o FO que deesa usar como base del SPS.
3. En el siguiente cuadro de diálogo que aparece debe seleccionar el esquema fuente para el SPS.
4. A continuación se crea un documento SPS basado en la estructura y en el formato del archivo XSLT o FO y se abre en la vista 37 **Diseño** 37.
5. Ahora puede modificar el SPS insertando y editando el contenido. Por ejemplo, puede arrastrar nodos desde la Estructura del esquema 48, modificar el estilo y la presentación y usar funciones de StyleVision como cálculos automáticos 293 y plantillas condicionales 303.
6. Ahora puede guardar el archivo SPS y usar un archivo XML de trabajo 45 para ver una vista previa en diferentes formatos de salida 39.
7. Por último puede generar hojas de estilos y archivos de salida 782 con el comando **Guardar archivos generados** 782.

Ejemplo

Los archivos necesarios para el ejemplo que usamos a continuación están en la carpeta (Mis) Documentos 34, `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\StyleVision2023\StyleVisionExamples/Tutorial/NewFromXSLT`. Esta carpeta incluye estos archivos: `SimpleExample.xslt`, `SimpleExample.xsd` y `SimpleExample.xml`.

El archivo XML es este:

Archivo XML utilizado en los gráficos de ejemplo: YearlySales.xml

```
<?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>
```

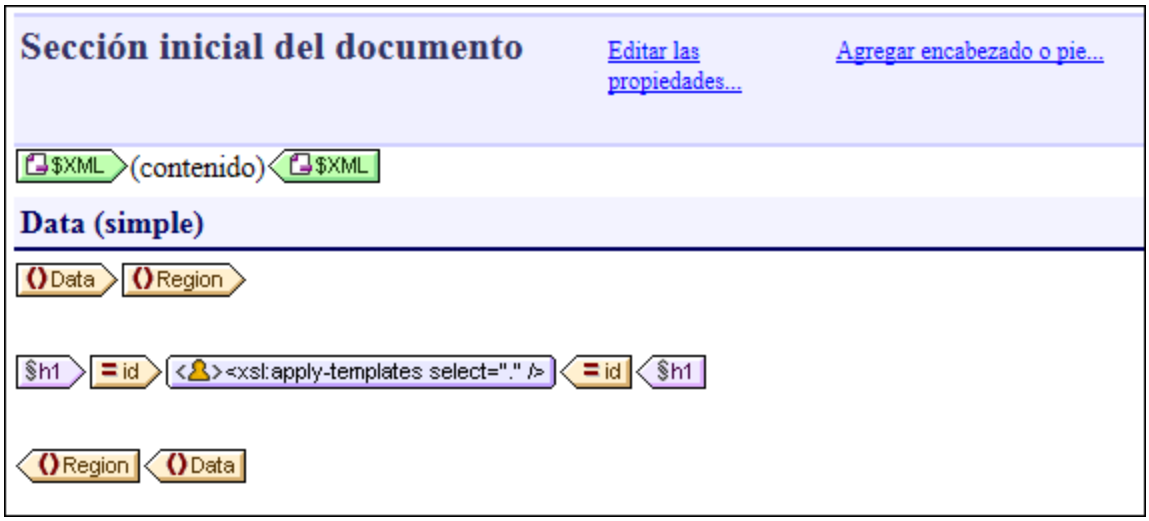
```
            <Year id="2009">95000</Year>
            <Year id="2010">80000</Year>
        </Region>
        <Region id="Asia">
            <Year id="2005">10000</Year>
            <Year id="2006">25000</Year>
            <Year id="2007">70000</Year>
            <Year id="2008">110000</Year>
            <Year id="2009">125000</Year>
            <Year id="2010">150000</Year>
        </Region>
</Data>
```

Y el archivo XSLT es este:

```
<?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">
   <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
   <xsl:template match="/">
      <html>
         <head>
            <title>Simple Example for New From XSLT</title>
         </head>
         <body>
            <xsl:apply-templates/>
         </body>
      </html>
   </xsl:template>
   <xsl:template match="Data">
      <xsl:for-each select="Region">
         <h1 style="color:red">
            <xsl:apply-templates select="@id"/>
         </h1>
      </xsl:for-each>
   </xsl:template>
</xsl:stylesheet>
```

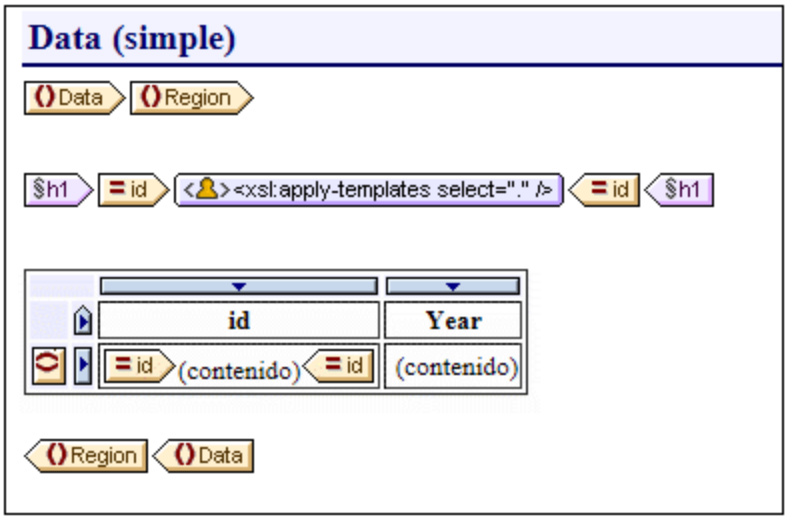
Siga los pasos 1, 2, 3 y 4 de las instrucciones para crear un archivo SPS y abrirlo en la vista **Diseño** (*imagen siguiente*):

 © 2016-2022 Altova GmbH



Observe que las dos plantillas del archivo XSLT se crearon en el diseño SPS. Ahora cambie a la vista previa **HTML** (*imagen siguiente*) y vea que el estilo del elemento `h1` (`color:red`) también se pasó al diseño SPS.

En la vista **Diseño** seleccione el elemento `h1` y cambie su color por el color negro (en la ventana Estilos, grupo de propiedades *Color*). Después, desde la Estructura del esquema, arrastre el elemento `Year` hasta la vista **Diseño** y póngalo en la posición del diseño que muestra la imagen siguiente. En el menú contextual que aparece seleccione el comando **Crear tabla**. Invierta el contenido de las dos columnas para que **Year** sea la primera columna de la tabla.



Y ahora puede seguir realizando cambios en el contenido, la estructura y las propiedades de presentación del documento. Después puede ver una vista previa de la transformación y guardar los archivos de salida con el comando **Guardar archivos generados**[782].

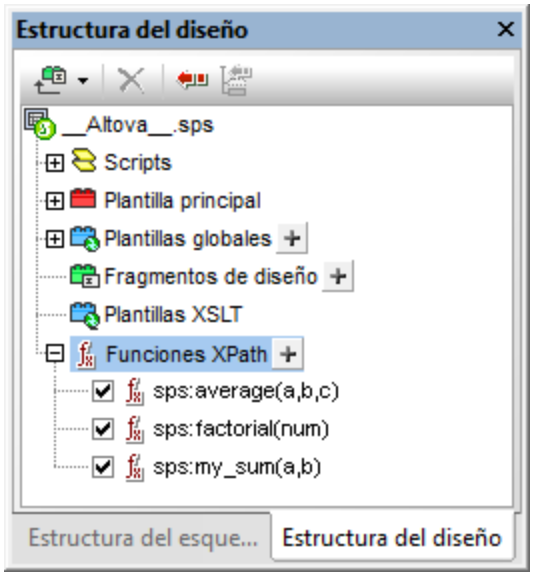
© 2016-2022 Altova GmbH

9.7 Funciones XPath definidas por el usuario

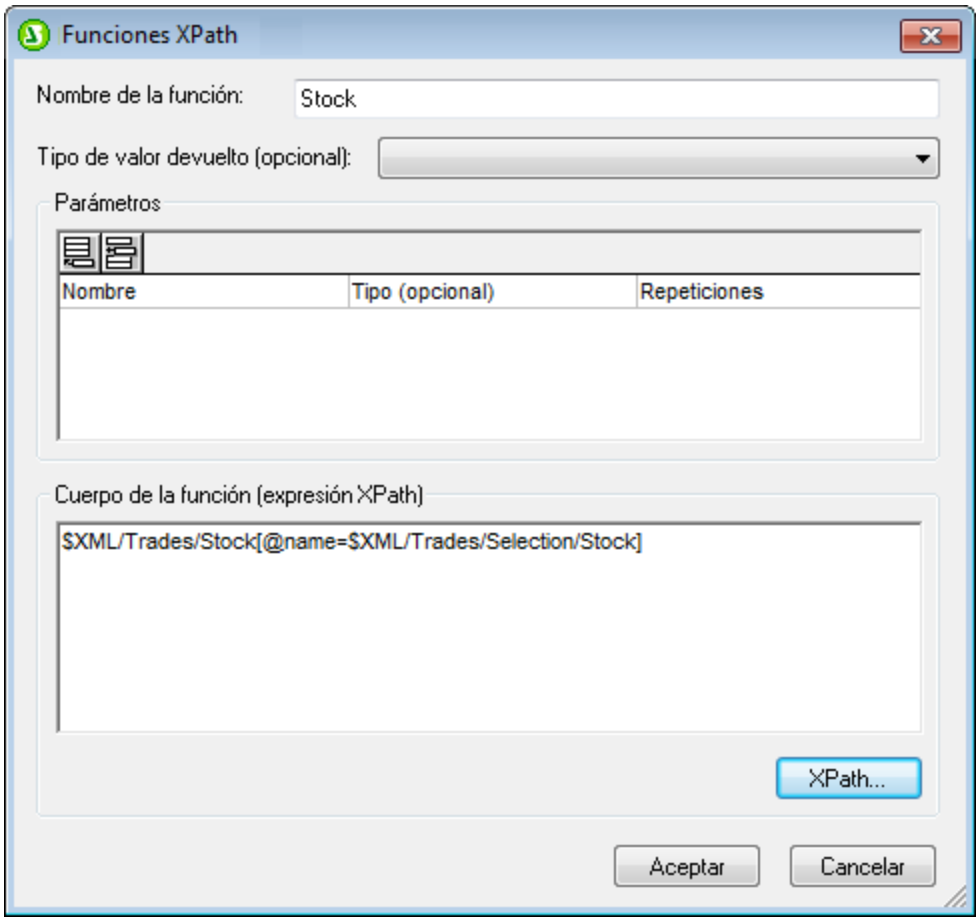
El diseñador SPS puede definir funciones **XPath 2.0/3.0** personalizadas. Las funciones XPath definidas por el usuario se pueden utilizar en cualquier componente de diseño que acepte expresiones XPath, como cálculos automáticos, condiciones y cuadros combinados, por ejemplo.

Definir y editar funciones XPath

Las funciones XPath definidas por el usuario se crean en la ventana Estructura del esquema y en la ventana Estructura del diseño (*imagen siguiente*). Todas las funciones XPath definidas por el usuario de un diseño SPS se enumeran bajo el elemento **`Funciones XPath`** de las ventanas Estructura del esquema y Estructura del diseño.



Para crear una función XPath definida por el usuario, haga clic en el icono del elemento **`Funciones XPath`**. Esto abre el cuadro de diálogo "Funciones XPath" (*imagen siguiente*). Si quiere editar una función del diseño SPS, haga doble clic en ella en la lista **`Funciones XPath`** y se abrirá el cuadro de diálogo "Funciones XPath", donde podrá editar la función.



Una vez creada, la función XPath definida por el usuario se puede usar en cualquier parte del diseño.

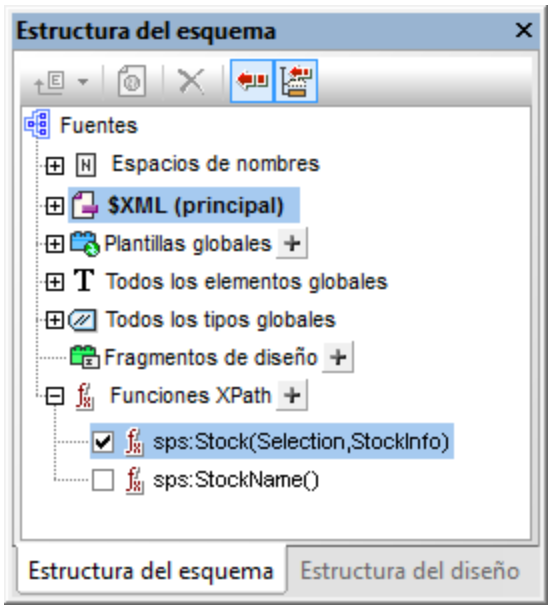
Espacio de nombres de las funciones XPath definidas por el usuario

Las funciones XPath definidas por el usuario se crean en el espacio de nombres: `http://www.altova.com/StyleVision/user-xpath-functions`. Este espacio de nombres está enlazado al prefijo `sps:`, de modo que debe llamar a las funciones XPath definidas por el usuario con este prefijo de espacio de nombres. Por ejemplo: `sps:MiFunción()`.

Habilitar y deshabilitar funciones XPath definidas por el usuario

Las funciones XPath definidas por el usuario se pueden habilitar o deshabilitar con solo activar o desactivar la casilla situada a la izquierda de la función en la lista **`Funciones XPath`** (*imagen siguiente*).

© 2016-2022 Altova GmbH



Esto es muy práctico si por ejemplo hay dos funciones con el mismo nombre. Esto puede pasar cuando, por ejemplo, un módulo SPS importado contiene una función que se llama igual que otra función del SPS.

Llamar a una función XPath definida por el usuario

Puede llamar a funciones XPath definidas por el usuario desde expresiones XPath y desde cualquier posición del diseño. Por ejemplo, podemos llamar a la función XPath definida por el usuario `sps:MiFunción` con esta expresión XPath dentro de un cálculo automático:

```
sps:MiFunción()/@name.
```

Esta expresión XPath se evalúa de la siguiente manera:

1. La función XPath `sps:MiFunción()` se evalúa. Digamos que la función se define así: `$XML/Trades/Stock[@name=$XML/Trades/Selection/Stock]`. Cuando la función se evalúa devuelve el elemento `/Trades/Stock` que tiene un atributo `name` cuyo valor coincide con el contenido del elemento `/Trades/Selection/Stock`.
2. El resultado del paso 1 se devuelve a la expresión XPath en la llamada a función. Ahora el valor del atributo `name` de este elemento `/Trades/Stock` se devuelve como valor del cálculo automático.

Eliminar una función

Para eliminar una función, selecciónela en la lista **`Funciones XPath`** de la Estructura del esquema o Estructura del diseño y después haga clic en el icono **Eliminar elemento** de la barra de herramientas de la ventana. Otra opción es hacer clic con el botón derecho en la función y seleccionar el comando **Quitar**.

9.7.1 Definir una función XPath

La definición de una función XPath exige: (i) un nombre (una cadena de texto) y (ii) una definición (una expresión XPath).

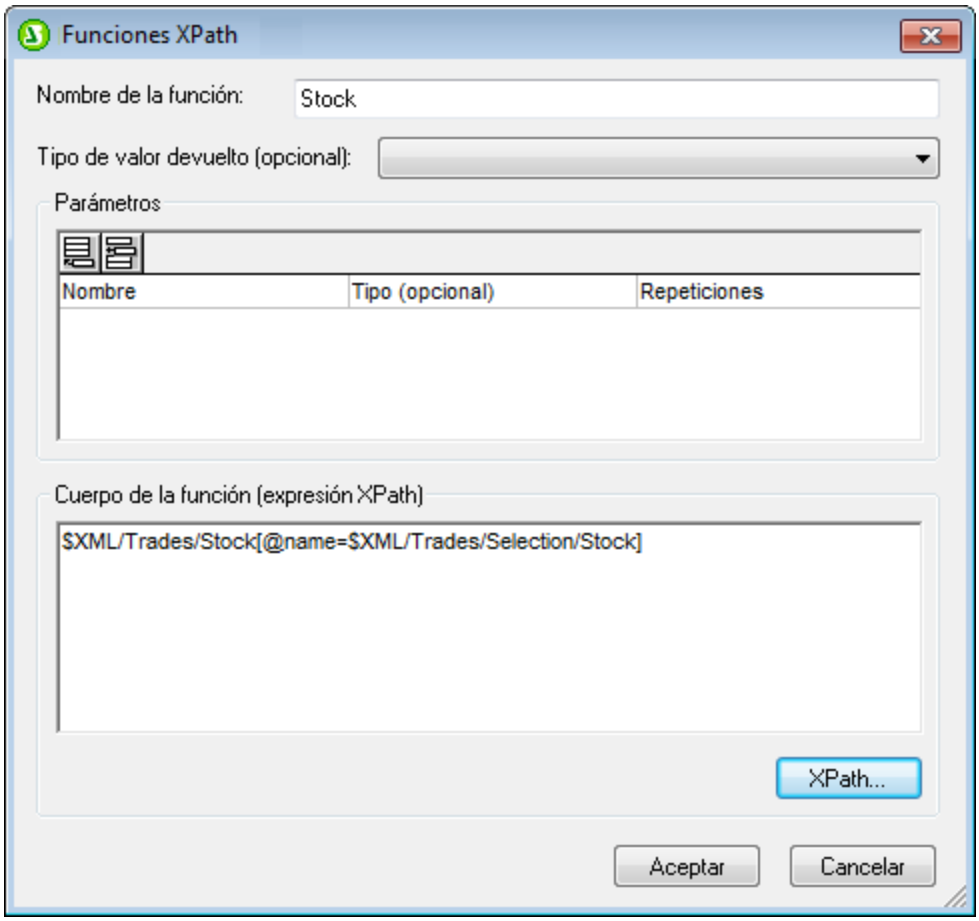
Además puede definir **parámetros** para la función. Una función XPath definida por el usuario también puede tener un **tipo devuelto** opcional, que se especifica seleccionando un tipo de la lista desplegable del cuadro combinado *Tipo de valor devuelto* en el cuadro de diálogo "Funciones XPath"(*imagen siguiente*). Un tipo devuelto sirve para comprobar si el tipo de datos del valor devuelto coincide con del tipo de datos seleccionado. Recuerde que el valor devuelto no se convierte en el tipo de datos seleccionado. Si hay un conflicto entre los tipos de datos, se devuelve un error. Elegir el tipo devuelto es opcional y, por tanto, si no especifica el tipo devuelto, no se realiza ninguna comprobación de tipos.

Una vez creada, la función XPath definida por el usuario se puede usar en cualquier parte del diseño. En la hoja de estilos XSLT, la función se crea como elemento `xsl:function`, que es un elemento secundario del elemento `xsl:stylesheet` (*ver ejemplo siguiente*).

```
<xsl:stylesheet>
   ...
   <xsl:function name="sps:Stock">
       <xsl:sequence select="$XML/Trades/Stock[@name=$XML/Trades/Selection/Stock]"/>
   </xsl:function>
   <xsl:function name="sps:Average" as="xs:decimal">
       <xsl:param name="a" as="xs:integer"/>
       <xsl:param name="b" as="xs:integer"/>
       <xsl:param name="c" as="xs:integer"/>
       <xsl:sequence select="avg( ($a, $b, $c) )"/>
  </xsl:function>
</xsl:stylesheet>
```

La función `sps:Stock` de la imagen siguiente y del fragmento de código anterior devuelve el elemento `/Trades/Stock` que tiene un atributo `name` cuyo valor coincide con el contenido del elemento `/Trades/Selection/Stock`. La función `sps:Average` del fragmento de código anterior devuelve la media de los tres valores de parámetro de entrada. La definición de la función usa la función XPath 2.0/3.0 `avg()`. El tipo de datos devuelto debe ser del tipo `xs:decimal`, que es el tipo de datos devuelto por la función `avg()` cuando se evalúan valores de entrada de tipo `xs:integer`. Si se especifica el tipo devuelto al definir la función, entonces se comprueba el tipo de datos del valor devuelto para ver si coincide con el tipo especificado. Si no coincide, se devuelve un error.

© 2016-2022 Altova GmbH



Definir la función

Para definir una función haga clic en el icono de la lista **Funciones XPath** de la Estructura del esquema o de la Estructura del diseño. Esto abre el cuadro de diálogo "Funciones XPath" (*imagen anterior*). Si desea editar una función ya existente, haga doble clic en el nombre de la función en la lista **Funciones XPath**. Después escriba un nombre y una definición para la función en el panel *Cuerpo de la función* del cuadro de diálogo "Funciones XPath" (*imagen anterior*). Si quiere, también puede introducir definiciones de parámetros (ver los dos apartados siguientes Parámetros y secuencias 483 yParámetros y nodos 488 para más información). Además puede especificar el tipo devuelto de la función (*ver más arriba*).

Lo más importante que debe tener en cuenta cuando escriba la expresión XPath que define la función XPath es que no hay nodo de contexto para la expresión XPath. Si la expresión XPath debe encontrar un nodo, entonces puede indicar el nodo de contexto de la expresión de dos maneras distintas:

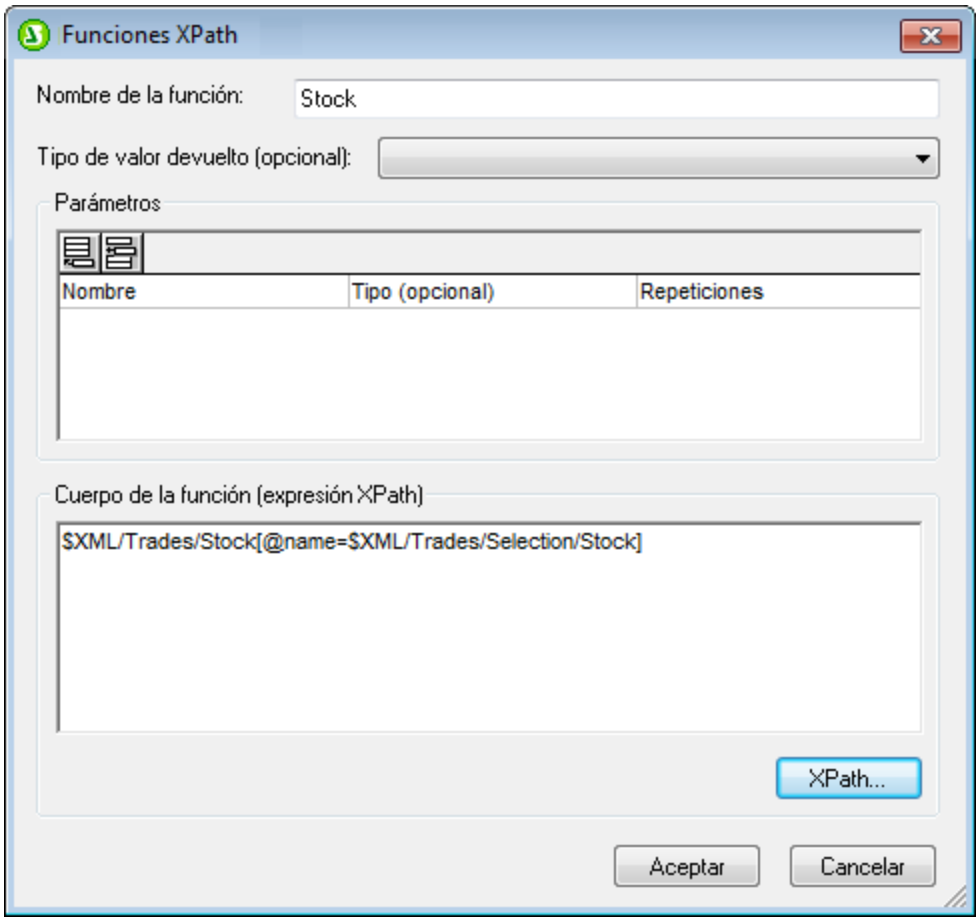
1. La expresión XPath comienza con la raíz de documento. La raíz de documento se especifica en el primer paso de la expresión XPath como `$XML`. Por ejemplo, la expresión XPath `$XML/Trades/Stock[1]` encuentra el primer elemento secundario `Stock` del elemento `/Trades`. StyleVision define la variable `$XML` (que encuentra la raíz de documento del esquema principal) a nivel global en todos los diseños SPS.
2. El nodo de contexto se puede pasar como parámetro. Para más información consulte el apartado Parámetros y nodos 488.

En algunos casos se devuelve un error:

- Cuando se define un parámetro pero no se utiliza en el cuerpo de la definición.
- Cuando el tipo de datos del valor devuelto por la función no coincide con el tipo devuelto definido por el usuario para la función.
- Cuando una función del SPS contiene un error, se genera un error XSLT para todo el diseño, aunque no se llame a la función que contiene el error. Sin embargo, no olvide que puede deshabilitar una función con solo desactivar su casilla en la lista **`Funciones XPath`**. Al deshabilitarla, la función no se incluye en el documento XSLT generado a partir del diseño. De este modo puede excluir expresiones XPath con errores de la hoja de estilos XSLT y evitar que se generen errores XSLT.

9.7.2 Reutilizar funciones para encontrar nodos

En el apartado anterior explicamos cómo generar una función XPath que encuentre un nodo concreto. La función `sps:Stock` que aparece en la imagen siguiente devuelve el elemento `/Trades/Stock` que tiene un atributo `name` cuyo valor coincide con el contenido del elemento `/Trades/Selection/Stock`.



Los pasos de la expresión `$XML/Trades/Stock[@name=$XML/Trades/Selection/Stock]` se puede dividir en varias funciones XPath. Por ejemplo:

- La función `sps:Stocks()`, cuya definición es: `$XML/Trades/Stock`
- La función `sps:SelectedStock()`, cuya definición es: `$XML/Trades/Selection/Stock`

© 2016-2022 Altova GmbH

Así podríamos escribir la expresión XPath en otras expresiones XPath como:

```
sps:Stocks()[@name=sps:SelectedStock()]
```

Cuando se crean así, las funciones XPath se pueden volver a utilizar en otras expresiones XPath del diseño SPS, lo cual permite crear expresiones XPath menos complicadas.

9.7.3 Parámetros en funciones XPath

A las funciones XPath definidas por el usuario se les pueden asignar parámetros. Los parámetros de la función se definen en el panel *Parámetros* del cuadro de diálogo "Funciones XPath" (*imagen siguiente*). Estos parámetros se pueden usar en la definición de la función XPath (en el panel *Cuerpo de la función*).

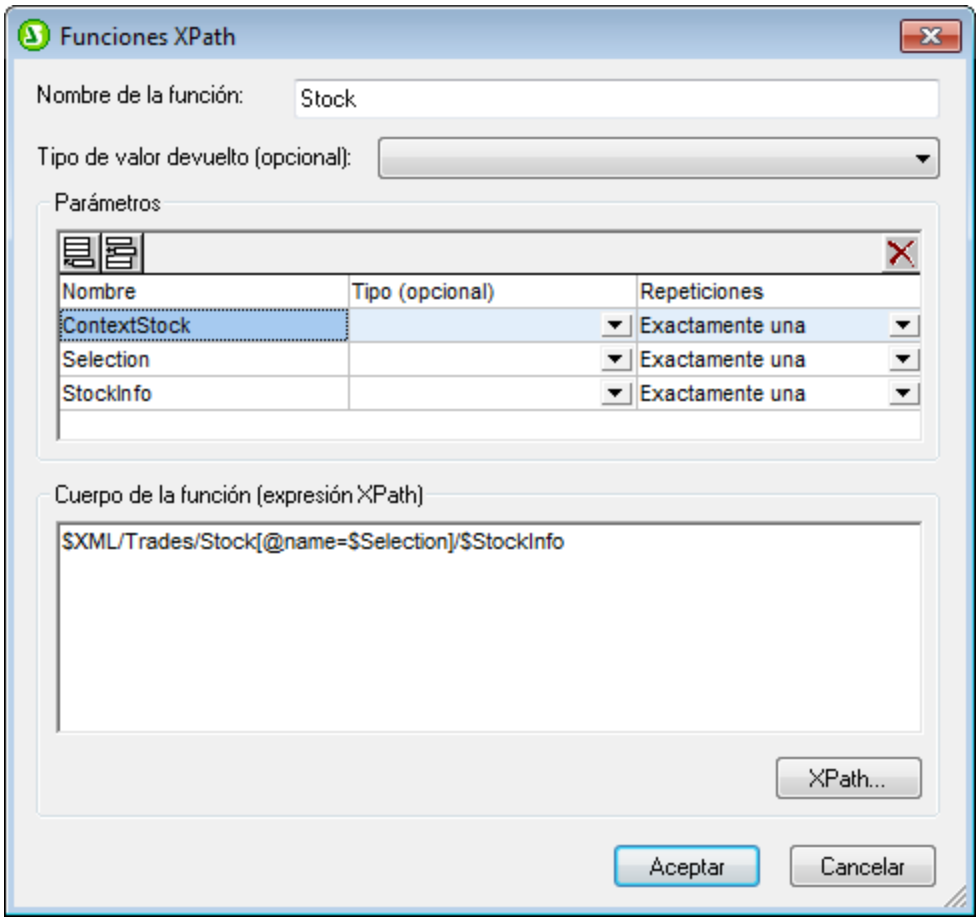
Funcionamiento de funciones XPath definidas por el usuario

Así funciona una función XPath:

1. En una llamada a función (por ejemplo, en un cálculo automático), el número de argumentos de la llamada a función debe coincidir con el número de parámetros definidos en la función (en el panel *Parámetros* del cuadro de diálogo "Funciones XPath", *imagen siguiente*). Además, el número de componentes suministrados por cada argumento (en la llamada a función) debe coincidir con el número dado en la columna *Repeticiones* del parámetro (*imagen siguiente*). Por último, si indicó una restricción de tipo de datos para el parámetro (en la columna *Tipo* del panel *Parámetros*), los valores dados por el argumento deben se de este tipo.
2. Los argumentos enviados a los parámetros de la función se utilizan en la función XPath (tal y como se defina en el panel *Cuerpo de la función*). El resultado obtenido al evaluar la expresión XPath se compara con la definición opcional de *Tipo devuelto* (*imagen siguiente*). Si el tipo de datos es el esperado, el resultado se usa en la expresión XPath desde la que se llamó a la función.

Orden de los parámetros

El orden de los parámetros de la función XPath definida por el usuario es importante porque cuando se llama a la función, los argumentos enviados en la llamada a función se asignan a los parámetros en el orden en el que se definieron en el panel *Parámetros* del cuadro de diálogo "Funciones XPath" (*imagen siguiente*).



De modo que si llamamos a la función XPath `sps:Stock` de la imagen anterior con esta expresión XPath:

```
sps:Stock($XML, Node1, Node2)
```

estos tres argumentos (`$XML, Node1, Node2`) se asignan en ese orden a los parámetros `$ContextStock`, `$Selection` y `$StockInfo` respectivamente.

Observe que los argumentos de la llamada a función están separados por comas. De esta forma, cada argumento se pasa al parámetro pertinente (de la lista del panel *Parámetros*).

Si quiere, puede cambiar el orden de los parámetros del panel *Parámetros* haciendo clic en los iconos **Anexar**, **Insertar** y **Eliminar**.

Tipo de datos de los parámetros

Si quiere, también puede definir el tipo de datos de los parámetros de la función XPath definida por el usuario. Si especifica un tipo de datos, el tipo de datos del argumento de entrada se compara con el tipo de datos del parámetro y, si los tipos no coinciden, se devuelve un error. Esta característica permite revisar los datos de entrada (de los argumentos de la llamada a función).

 © 2016-2022 Altova GmbH

Repeticiones

Los parámetros de la función XPath definida por el usuario se pueden entender como secuencias. La propiedad *Repeticiones* de un parámetro especifica cuántos elementos debe suministrar el argumento correspondiente de la llamada a función para dicho parámetro.

Tanto en las definiciones de función como en las llamadas a función se utilizan comas para separar los parámetros o los argumentos, así como para separar los componentes de una secuencia. Por tanto, es importante tener en cuenta en qué contexto se usan las comas: para separar parámetros/argumentos o para separar los componentes de una secuencia.

- Con los parámetros/argumentos, puede usar paréntesis para delimitar secuencias en la definición de función (parámetros) o en la llamada a función (argumentos).
- En las secuencias los paréntesis se pasan por alto.

A este respecto debe tener en cuenta estas reglas de uso:

- ***Uso de paréntesis en parámetros/argumentos:*** Varias funciones XPath toman como argumento una sola secuencia (p. ej. las funciones `avg()` y `count()`). Si esta secuencia está enumerada por medio de comas o de operadores de intervalo, la secuencia debe ir entre paréntesis para confirmar inequívocamente que se trata de una secuencia (y no de varias secuencias separadas por comas). Por ejemplo, en la función `avg((count($a), $b, $c))`, la función XPath 2.0 `avg()` toma la secuencia `(count($a),$b,$c)` como argumento. Como los componentes de la secuencia están enumerados, la secuencia debe ir entre paréntesis: `avg(`**`(count($a),$b,$c)`**`)`. Sin los paréntesis de la secuencia, la definición de la función `avg()` tendría tres parámetros y eso no sería válido porque la función `avg()` solo toma un argumento, compuesto por una sola secuencia.

- ***Ningún paréntesis en parámetros/argumentos:*** Igualmente, la función `count()` toma una sola secuencia como argumento. No obstante, en el ejemplo anterior la función `count($a)` toma una secuencia que no es una lista de componentes separados por comas, sino que la secuencia la recupera la variable/el parámetro `$a`. Por tanto, no es necesario poner el argumento entre paréntesis. Es decir, la expresión `count($a)` es correcta.

- ***Uso de paréntesis y comas en llamadas a función:*** En una llamada a función los paréntesis deben utilizarse correctamente para que cada argumento corresponda al parámetro correcto (del panel *Parámetros* del cuadro de diálogo "Funciones XPath"). Por ejemplo, si una función XPath definida por el usuario llamada `MyAverage()` se define con la expresión XPath 2.0: `avg((count($a),$b,$c))`, entonces esta llamada a función sería válida: `MyAverage((1,2,3),4,5)`. Los valores correspondientes a los tres parámetros `$a`, `$b` y `$c` serían la secuencia `(1,2,3)`, la secuencia única `4`, y la secuencia única `5`. Si quiere puede poner las secuencias únicas entre paréntesis. El valor devuelto por la función `MyAverage()` es `4`.

9.7.3.1 Parámetros y secuencias

Es importante mencionar la relación que existe entre parámetros y secuencias y el uso que se hace de ellos en las expresiones XPath. En esta documentación usamos los siguientes términos:

- Una ***secuencia*** está formada por componentes que son valores atómicos o nodos. Una coma sirve para construir una secuencia: poniendo una coma entre los componentes de una secuencia.

- La definición de una función XPath puede indicar que la función debe tomar ***parámetros***. Por ejemplo, en la expresión XPath 2.0 `count($a)`, la parte situada entre los paréntesis de la función es el parámetro de la función y debe ser una secuencia de componentes.
- En una llamada a función un ***argumento*** está formado por uno o varios componentes. Por ejemplo, la función `count(//Person)` tiene un argumento: `//Person`. Este argumento es válido porque devuelve una secuencia de nodos que equivale a la firma de la función `count()`. La firma de una función especifica el número de parámetros y el tipo de datos esperado de cada parámetro. También especifica qué devuelve la función y el tipo de datos del objeto devuelto.
- La función `substring('StyleVisionExamples', 6, 6)`, que devuelve la cadena `Vision`, tiene tres argumentos. Según la firma de la función `substring()`, esto es válido. Cuando una llamada a función tiene varios argumentos, los argumentos se separan con comas.

Paréntesis como delimitador de secuencias

Hay algo muy importante que debe tener en cuenta cuando construya expresiones XPath: *para delimitar secuencias que usen comas o el operador de intervalos* ***se utilizan paréntesis****. Como resultado, cada secuencia delimitada por paréntesis se lee como un solo parámetro (en las definiciones de función) o como un solo argumento (en las llamadas a función).*

Los paréntesis no son obligatorios en una expresión de ruta (p. ej. `//Person/@salary`) porque las expresiones de ruta se pueden leer inequívocamente como un solo parámetro o argumento. De hecho, una expresión de ruta es un parámetro/argumento formado por una secuencia.

Estos ejemplos ilustran el punto anterior:

- `avg(`**`(10, 20, 30)`**`)`

  La función XPath 2.0 `avg` toma una secuencia de componentes como único argumento. Como esta secuencia es una enumeración separada por comas, los paréntesis interiores son necesarios para delimitar la secuencia obligatoria. Sin los paréntesis interiores, la función tendría tres argumentos y, por tanto, no sería válida. Los paréntesis exteriores son los paréntesis de la función.

- `avg(`**`//Person/@salary`**`)`

  Esta expresión de ruta selecciona los nodos de atributo `salary` de todos los elementos `Person` y devuelve sus valores en forma de secuencia que se debe evaluar (cuyo promedio se debe calcular). En este caso no hacen falta los paréntesis porque la secuencia no se enumera. El argumento es una sola expresión de ruta. La expresión de ruta se evalúa y los valores devueltos se suministran a la función como componentes de una secuencia.

- `count(`**`(10 to 34)`**`)`

  Esta enumeración se consigue con el operador de intervalos. El operador de intervalos `'to'` genera una secuencia de componentes separados por comas (los enteros comprendidos entre 10 y 34) antes de que se lea el argumento. Como resultado, la función `count()` tiene una secuencia de 25 componentes separados por comas dentro de su argumento. Para poder leer esta secuencia como un único argumento es necesario añadir los paréntesis. Sin estos paréntesis, la llamada a función tendría 25 argumentos en vez de uno y la llamada no sería válida porque la función `count()` solo puede tener un argumento, según su firma.

- `count(`**`(10 to 34, 37)`**`)`

© 2016-2022 Altova GmbH

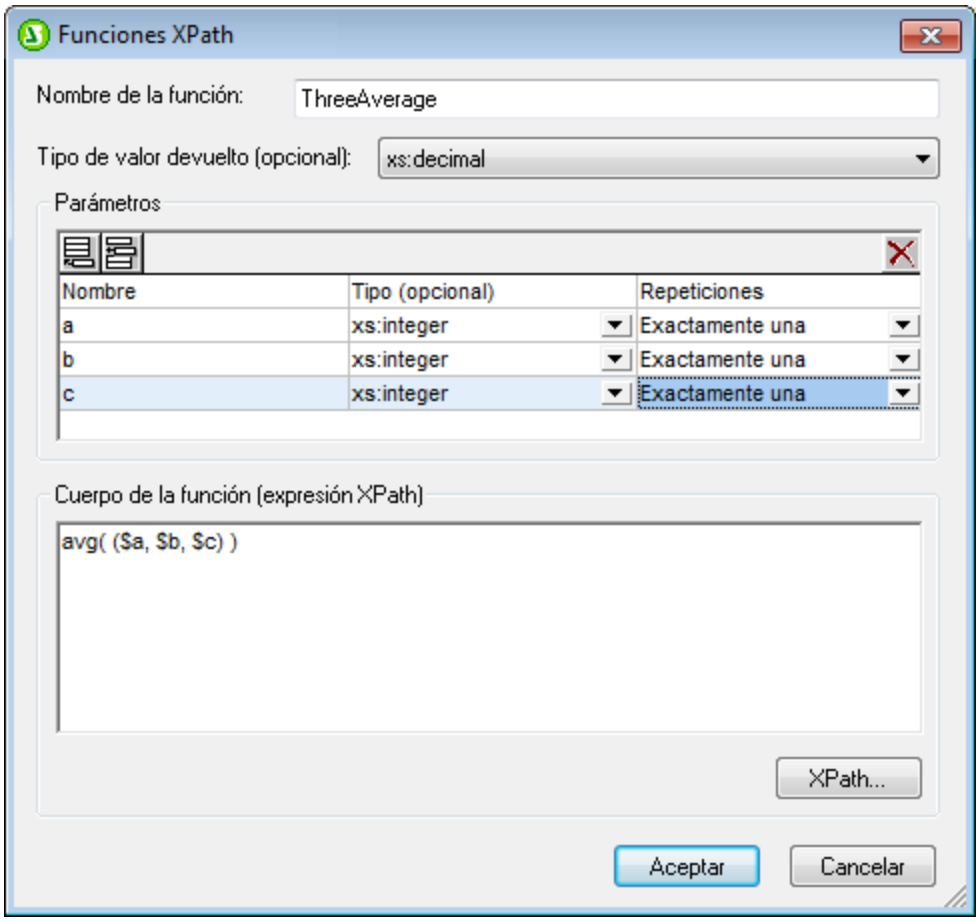
Los paréntesis interiores señalan un argumento de la llamada a función: una sola secuencia formada por 26 componentes.

- `count(//Person)`

  En este caso no hace falta poner el argumento entre paréntesis porque el argumento es una expresión de ruta que recopila los nodos `//Person` del documento XML y devuelve estos nodos como componentes de la secuencia que se debe contar.

Usar parámetros XPath en funciones XPath

Cuando use parámetros en la definición de una función XPath definida por el usuario, asegúrese de que (i) el número de argumentos de la llamada a esta función XPath es correcto y de que (ii) los argumentos dan como resultado el tipo de datos y las repeticiones indicadas en la definición de la función.



En el ejemplo de la imagen anterior se definieron tres parámetros, que se utilizan para definir la función XPath (en el panel *Cuerpo de la función*) .

Cada parámetro del panel *Parámetros* se considera una sola secuencia. El número de componentes permitidos dentro de cada secuencia se especifica con la propiedad *Repeticiones*. En la definición de la imagen anterior, por ejemplo, cada parámetro se definió como secuencia única (ver propiedad *Repeticiones*), es decir, una secuencia con exactamente un componente. Por tanto, cada argumento de la llamada a función

debe tener una secuencia formada por exactamente un componente. La propiedad *Tipo* especifica el tipo de datos de los componentes de la secuencia.

En el ejemplo anterior, en la definición de función XPath (en el panel *Cuerpo de la función*) cada parámetro ofrece un componente de la secuencia cuyo promedio debe calcularse. Como el conjunto de parámetros XPath constituye una secuencia, estos deben ir entre paréntesis para garantizar toda la secuencia se lea como parámetro de la función `avg()`. Si en tiempo de ejecución alguno de los argumentos de la llamada a función (equivalentes a los tres parámetros) no es una secuencia única, se devuelve un error.

A continuación damos algunos ejemplos de expresiones XPath que llaman a la función XPath `ThreeAverage()`. En la vista **Diseño** puede insertar un cálculo automático y darle las expresiones XPath que aparecen a continuación para ver los resultados. La función toma una secuencia de tres enteros y calcula el promedio.

- `sps:ThreeAverage(10,20,30)` devuelve `20`. Hay tres argumentos válidos en la llamada a función, que equivalen a los tres parámetros XPath.

- `sps:ThreeAverage( (10),(20),(30) )` devuelve `20`. Hay tres argumentos válidos en la llamada a función, que equivalen a los tres parámetros XPath. Cada argumento de entrada va entre paréntesis (lo cual es redundante porque cada secuencia es una secuencia única de todas maneras. Sin embargo, esta redundancia no constituye un error).

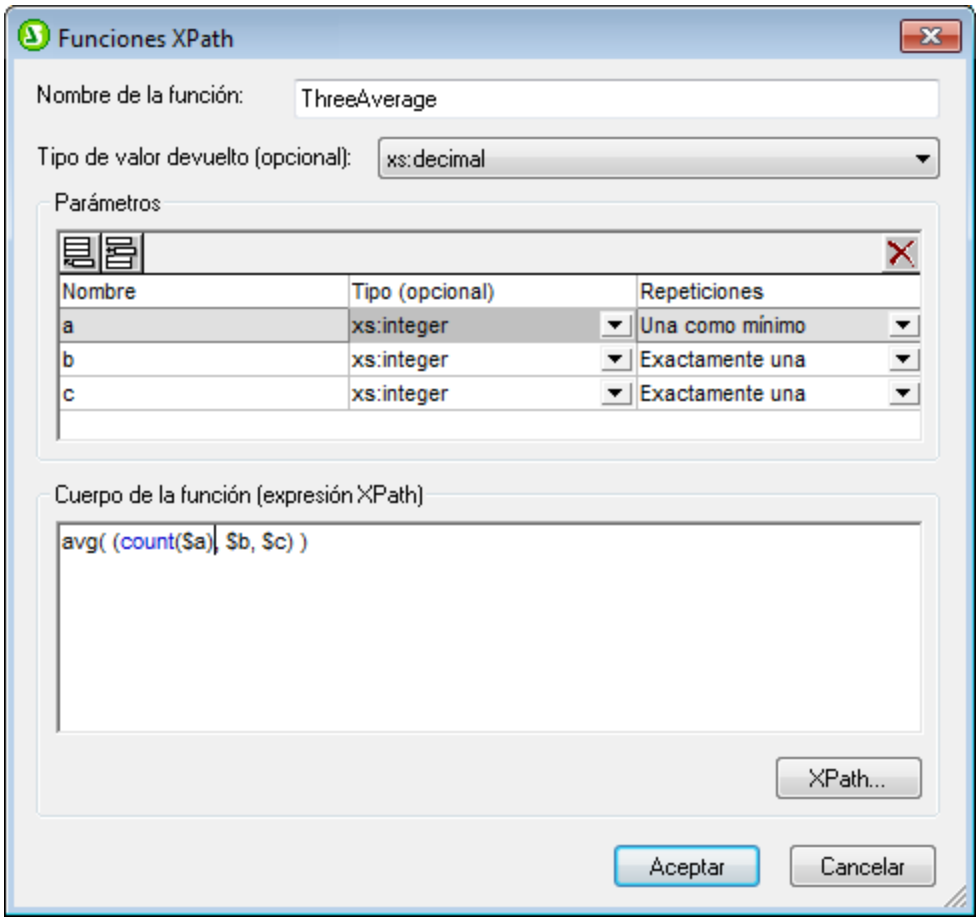
- `sps:ThreeAverage( (10),20,30 )` devuelve `20`. Hay tres argumentos válidos, que equivalen a los tres parámetros XPath. El primer argumento va entre paréntesis (lo cual es redundante pero no es un error).

- `sps:ThreeAverage( (10,20),(30),(40) )` devuelve un error porque el primer argumento no es válido al ser no ser una secuencia única tal y como exige la definición del primer parámetro `$a` (`'Exactamente una repetición'`).

- `sps:ThreeAverage( (10,20,30) )` devuelve un error porque solamente se suministra un argumento entre paréntesis. Además el argumento no es válido porque no es una secuencia única.

Si la propiedad *Repeticiones* de un parámetro es `Una como mínimo` (*imagen siguiente*), entonces ese parámetro se define como secuencia compuesta por varios componentes.

 © 2016-2022 Altova GmbH



En la definición de la imagen anterior, el primer parámetro se definió como una secuencia de varios componentes y los otros dos parámetros como secuencias únicas. La función definida cuenta el número de componentes suministrados por el primer parámetro, añade el resultado a la suma de los dos enteros dados por los otros dos parámetros y por último divide el resultado por tres para calcular el promedio. Observe que:

- El parámetro de la función `avg()` es una secuencia entre paréntesis. Esto especifica que la función `avg()` toma como parámetro una sola secuencia compuesta por tres componentes. La secuencia está formada por tres enteros: el primero es el suministrado por la función `count()` y el segundo y el tercero son los dados por los parámetros `b` y `c`.
- El argumento de la función `count()` no está entre paréntesis porque es inequívocamente una secuencia única.

A continuación damos dos ejemplos del uso de parámetros en llamadas a la función XPath `Average()` utilizada en el ejemplo.

- `sps:Average((1,2),3,4)` devuelve `3`. Hay tres argumentos de entrada válidos en la llamada a función, que equivalen a los tres parámetros XPath. El primer argumento está entre paréntesis. Cuando la función `count()` se ejecuta en este argumento, la función devuelve el valor `2`, que será el primer componente de la secuencia suministrada a la función `avg()`.
- `sps:Average( 4,4,4 )` devuelve `3`. Hay tres argumentos de entrada válidos. El primer argumento puede ser una secuencia de un componente. No hace falta usar paréntesis para separar los argumentos.

Otros aspectos de importancia

Además debería tener en cuenta estos aspectos:

- Si definió un parámetro con una repetición de `Una como mínimo`, entonces puede definir una función como `MyAverage()` con una expresión XPath como `avg(($a))`. Esta función acepta un argumento que es una secuencia formada por varios componentes. Puede llamar a esta función de esta forma: `sps:MyAverage((2,3,4))`, lo cual devuelve el valor `3`. El argumento de entrada debe ir entre paréntesis para garantizar que se lea como una sola secuencia y no como tres secuencias únicas.

- Si definió un parámetro `$a` con una repetición de `Una o ninguna`, entonces puede definir una función como `MyAverage()` con una expresión XPath como `avg(($a, $b, $c))`. Esta función acepta tres secuencias como argumento, con la posibilidad de que la primera secuencia esté vacía. Si la primera secuencia debe estar vacía, entonces debe enviar una secuencia vacía explícitamente como primer argumento de entrada. De lo contrario se emite un error. Si llamamos a la función así `sps:MyAverage(30,20,10)`, devuelve el valor `20`. Pero también se le puede llamar así `sps:MyAverage(((),20,10)`, lo cual devuelve el valor `15` (observe que la secuencia vacía cuenta como valor de entrada vacío. Para que el valor devuelto sea 10, el primer componente debería ser `0`). Sin embargo, la llamada a función `sps:MyAverage(20,10)` devuelve un error porque no se suministró una primera secuencia vacía y, por tanto, se entiende que el tercer argumento de entrada está ausente.

Ejemplos complejos

Como decíamos, una gran ventaja de las funciones XPath definidas por el usuario es que se pueden usar en expresiones XPath del diseño SPS. Otra ventaja es que permiten construir funciones XPath complejas personales, no disponibles en el grupo de funciones de XPath 2.0. Por ejemplo, puede construir una función factorial con una expresión XPath que tome una secuencia única como único parámetro. Si el parámetro `$num` es el número que se debe factorizar, entonces la expresión XPath que crearía la función sería:

```
if ($num < 2) then 1 else $num * sps:Factorial($num - 1)
```

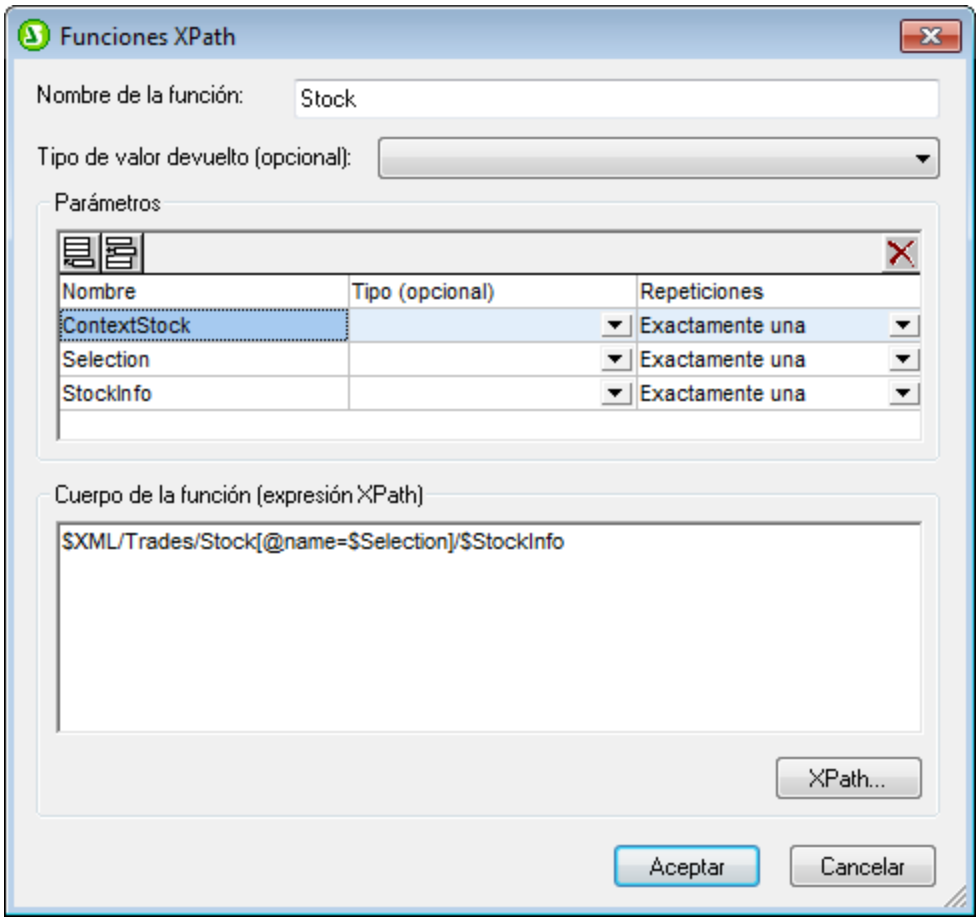
Si esta función se llamara `Factorial()`, entonces se podría obtener el factorial `6`, por ejemplo, llamando a la función con `sps:Factorial(6)`.

9.7.3.2 Parámetros y nodos

Cuando use parámetros en funciones XPath que encuentran nodos, debe tener en cuenta que la función no tiene nodo de contexto, independientemente de la posición desde la que se llame a la función en el diseño. El nodo de contexto se puede suministrar bien en la expresión XPath que se usa para definir la función (en el panel *Cuerpo de la función*) o bien en la expresión XPath que se usa para llamar a la función. En este último caso, el contexto se puede aportar mediante argumentos en la llamada a función.

Por ejemplo, imagine la función XPath definida por el usuario `Stock()`, que tiene definidos los tres parámetros que aparecen en la imagen siguiente.

 © 2016-2022 Altova GmbH



La definición del cuerpo de la función es `$ContextStock[@name=$Selection]/$StockInfo`, que usa los tres parámetros pero no incluye información sobre el nodo de contexto. La información sobre el nodo de contexto se puede dar en la expresión XPath que llama a la función. Por ejemplo:

```
sps:Stock( $XML/Trades/Stock, $XML/Trades/Selection/Stock, @name )
```

La llamada a función tiene tres argumentos, cuyos valores aportan la información de contexto o el localizador de nodos. También podría usar estas expresiones XPath como llamada a función, que dan el mismo resultado:

```
sps:Stock( /Trades/Stock, /Trades/Selection/Stock, @name )
sps:Stock( /Trades/Stock, //Selection/Stock, @name )
```

La variable `$XML`, que devuelve la raíz de documento, puede omitirse de las llamadas a función desde componentes de diseño porque el nodo de contexto se conoce en las expresiones XPath de los componentes de diseño.

Observe que en la llamada a función del ejemplo anterior hay tres argumentos de entrada que equivalen respectivamente a los tres parámetros definidos para la función XPath definida por el usuario:

- `$ContextStock = $XML/Trades/Stock` (el elemento `/Trades/Stock`)
- `$Selection = $XML/Trades/Selection/Stock` (el elemento `/Trades/Selection/Stock`)
- `$StockInfo = @name`

La expresión XPath de la definición de función es:

```
$ContextStock[@name=$Selection]/$StockInfo
```

Cuando se reemplazan los argumentos de entrada, la expresión XPath de la definición de la función se convierte en:

```
$XML/Trades/Stock[@name=$XML/Trades/Selection/Stock]/@name
```

Es importante tener en cuenta que lo que se pasa a la función son los conjuntos de nodos y no las cadenas de texto.

De este modo se pasa el nodo de contexto y los pasos de búsqueda a la función. A partir de ese momento la función se puede evaluar para encontrar y devolver los nodos necesarios.

© 2016-2022 Altova GmbH

9.8 Trabajar con fechas

Si el documento de origen contiene nodos que toman valores de fecha, puede usar los tipos de datos `xs:date` o `xs:dateTime` del esquema XML subyacente para aprovechar las potentes funciones de manipulación de datos de XPath 2.0/3.0 (*ver ejemplos* 491). StyleVision es compatible con los tipos de datos `xs:date` y `xs:dateTime` y ofrece:

1. Un selector de fechas 492 gráfico para ayudar al usuario de la vista Authentic a insertar fechas en el formato léxico correcto del tipo de datos del nodo. El selector de fechas solamente está disponible en la vista Authentic porque es un dispositivo de entrada de datos.
2. Un amplio abanico de opciones de formato de fechas 493 (disponibles a través de la característica Formato de valor 380).

Estas características de StyleVision se describen en los apartados de esta sección: Usar el selector de fechas 492 y Formato de fechas 493.

Nota: los datos de fecha y hora no se pueden manipular con XPath 1.0. No obstante, si usa XPath 1.0 en su diseño puede usar el selector de fechas 492 para garantizar la integridad de los datos y la característica *Formato de valor* para dar formato a las fechas 493.

Calcular fechas con XPath 2.0

Puede manipular fechas con ayuda de cálculos automáticos 293 basados en expresiones XPath 2.0. A continuación ofrecemos algunos ejemplos.

- Las funciones XPath 2.0 `current-date()` y `current-dateTime()` sirven para obtener la fecha y la fecha-hora actuales respectivamente.
- Puede realizar substracciones con valores de fecha. Por ejemplo: `current-date() - DueDate` devolvería un valor `xdt:dayTimeDuration` como `P24D`, por ejemplo, que indica una diferencia positiva de 24 días.
- Puede usar funciones XPath 2.0 para extraer unidades de tiempo de duraciones. Por ejemplo: `days-from-duration(xdt:dayTimeDuration('P24D'))` devolvería el entero `24`.

El siguiente ejemplo muestra una expresión XPath 2.0 dentro de un cálculo automático. La expresión calcula un interés anual del 4% sobre una cantidad no pagada por cada día de retraso y devuelve la suma de la cantidad principal más el interés acumulado:

```
if            (current-date() gt DueDate)
then   (round-half-to-even(InvoiceAmount +
                     (InvoiceAmount*0.04 div 360 *
                            days-from-duration((current-date() - DueDate))), 2))
else   InvoiceAmount
```

Este cálculo se podría hacer con XPath 2.0 si el elemento `DueDate` tuviera definido el tipo de datos `xs:date` y el contenido del elemento se insertara en su formato léxico correcto (es decir, `YYYY-MM-DD[±HH:MM]`), siendo el componente de uso horario (prefijo ±) opcional. Gracias al selector de fechas puede asegurarse de que el usuario de la vista Authentic inserte las fechas en el formato léxico correcto.

9.8.1 Usar el selector de fechas

El selector de fechas (*imagen siguiente*) es un calendario gráfico de la vista Authentic que sirve para insertar fechas en el formato léxico correcto para los nodos de tipo de datos `xs:date` y `xs:dateTime`.

El formato léxico de los datos es el del tipo de datos.

- Para el tipo de datos `xs:date` el formato de los datos insertados es `YYYY-MM-DD[±HH:MM]`, siendo el componente de zona horaria (prefijo ±) un componente opcional, de acuerdo con la especificación XML Schema. El valor del componente de zona horaria se puede elegir también en el selector de fechas.
- Para el tipo de datos `xs:dateTime` el formato de los datos insertados es `YYYY-MM-DDTHH:MM:SS[±HH:MM]`, siendo el componente de zona horaria (con el prefijo ±) un componente opcional, de acuerdo con la especificación XML Schema. El valor del componente de zona horaria se puede elegir también en el selector de fechas.

Insertar y eliminar un selector de fechas en el diseño

En el diseño SPS puede insertar un selector de fechas (i) para cualquier nodo que tenga el tipo de datos `xs:date` o `xs:dateTime` y (ii) cuando a partir de este nodo se crea contenido (marcador de posición `(contenido)`) o un campo de entrada. Hay dos formas de insertar el selector de fechas:

- De forma automática cuando se inserte un nodo de tipo `xs:date` o `xs:dateTime` en el diseño SPS. Para que el selector de fechas se inserte automáticamente con este tipo de nodos, primero debe activar la función **Agregar selector de fechas automáticamente** (haga clic en el comando **Authentic | Agregar selector de fechas automáticamente**). Cuando esta función está activada, el selector de fechas se inserta cada vez que se crea contenido o un campo de entrada a partir de un elemento de tipo `xs:date` o `xs:dateTime` o cada vez que un componente se convierte en uno de estos dos componentes.
- El selector de fechas también se puede insertar con el comando de menú **Insertar | Selector de fechas** cuando el cursor está dentro de un nodo de tipo `xs:date` o `xs:dateTime` del diseño SPS. Este comando está disponible tanto en el menú **Insertar** como en el menú contextual que aparecer al hacer clic con el botón derecho dentro de un nodo de tipo `xs:date` o `xs:dateTime`.

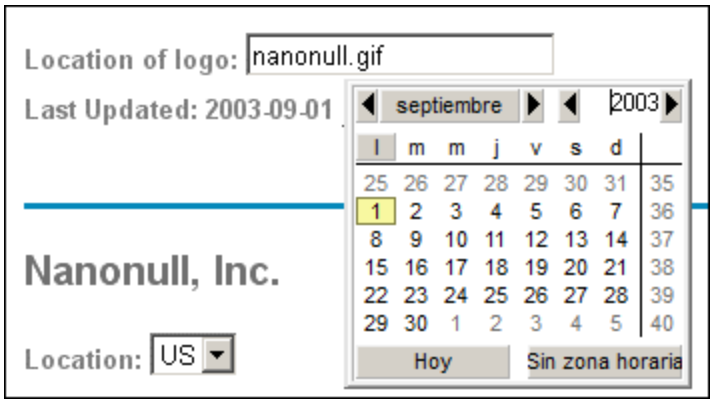
En la posición donde se insertó el selector de fechas aparece este icono . Para eliminar un selector de fechas, selecciónelo y pulse la tecla **Supr** o **Retroceso**.

Usar el selector de fechas en la vista Authentic

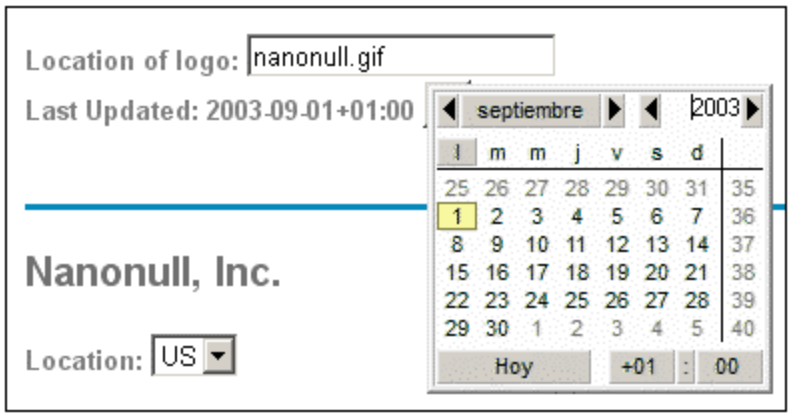
En la vista Authentic el selector de fechas aparece en forma de icono (*imagen siguiente*).

 © 2016-2022 Altova GmbH

Haga clic en el icono para abrir el selector de fechas propiamente dicho (*imagen siguiente*). Seleccione la nueva fecha y la fecha se introducirá en el formato léxico correcto correspondiente al tipo de datos del nodo en cuestión.



Para introducir una zona horaria, haga clic en el botón **Sin zona horaria** (*imagen anterior*) y seleccione una opción. La zona horaria se introduce en el formato léxico correcto correspondiente al tipo de datos del nodo (*imagen siguiente*).



9.8.2 Formato de fechas

En un documento XML las fechas se guardan en el formato específico del tipo de datos del nodo en cuestión. Por ejemplo, el valor de un nodo `xs:date` tiene el formato `YYYY-MM-DD[±HH:MM]`, mientras que el valor de un nodo `xs:dateTime` tiene el formato `YYYY-MM-DDTHH:MM:SS[±HH:MM]`. Estos formatos son la representación

léxica de los datos. En la vista Authentic y en los documentos de salida se muestra por defecto esta representación léxica de los datos. Sin embargo, puede usar la función *Formato de valor* del diseño SPS para que los datos aparezcan con otro formato en la vista Authentic y, en algunos casos, también en los documentos de salida.

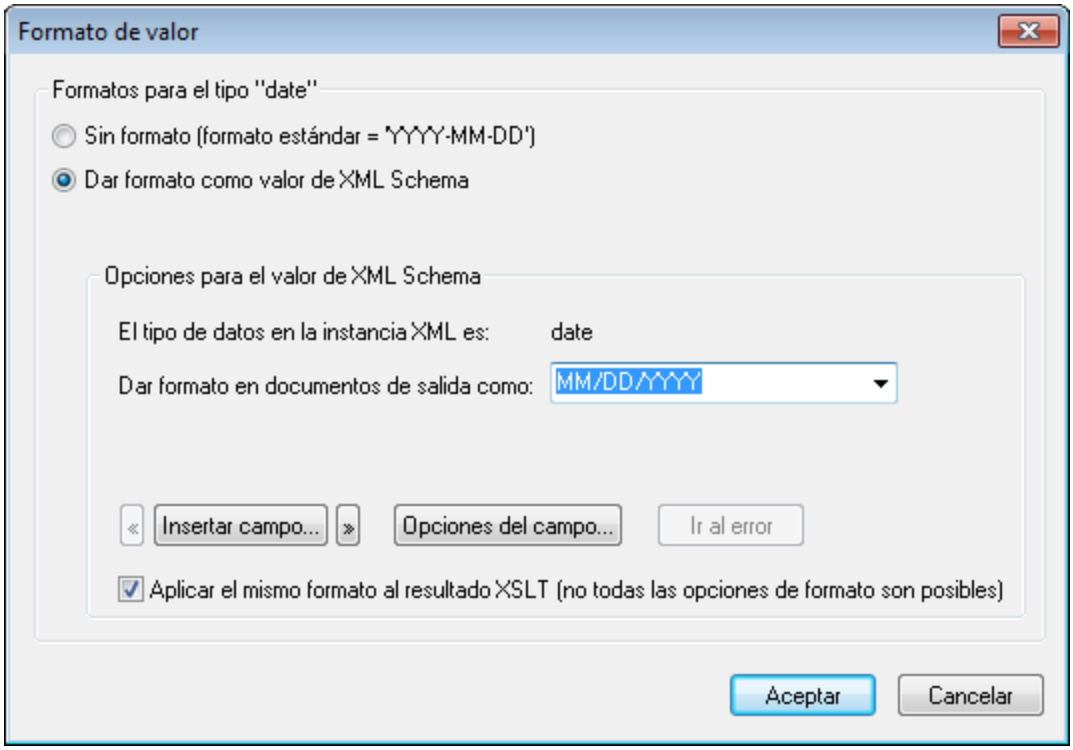
En el caso de valores de fecha puede usar la función *Formato de valor* para definir formatos personalizados para nodos y cálculos automáticos de este tipo de datos:

- `xs:date`
- `xs:dateTime`
- `xs:duration`
- `xs:gYear`
- `xs:gYearMonth`
- `xs:gMonth`
- `xs:gMonthDay`
- `xs:gDay`

Usar el formato de valor para dar formato a los nodos de fecha

Siga estos pasos para dar a un nodo de fecha un formato distinto a su formato léxico:

1. Seleccione el marcador de posición `(Cálculo automático)` o el campo de entrada del nodo. Recuerde que el formato de valor solamente se puede aplicar a nodos que han creado **contenido o un campo de entrada** en el diseño SPS.
2. En la ventana Propiedades seleccione el componente **`contenido`** y después expanda el grupo de propiedades *Cálculo automático*. Ahora haga clic en el botón **Edición** ... de la propiedad `formato de valor`. Esto abre el cuadro de diálogo "Formato de valor" (*imagen siguiente*).



 © 2016-2022 Altova GmbH

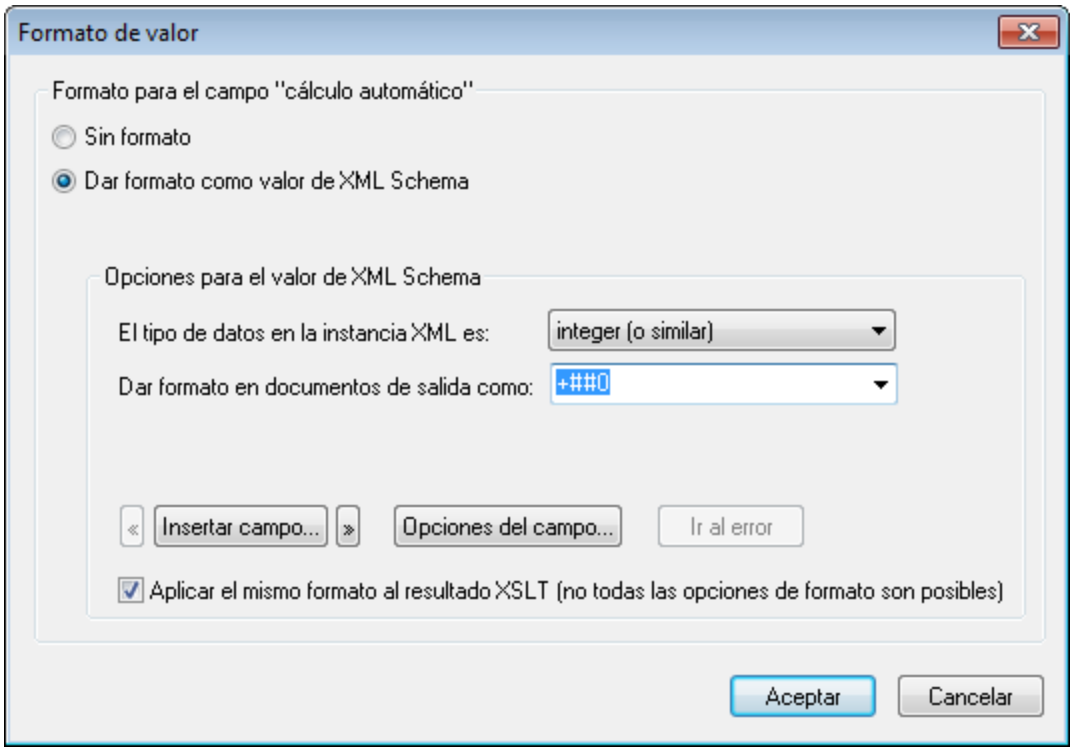
La opción predeterminada es la opción *Sin formato* (el formato léxico estándar del tipo de datos del nodo seleccionado).

3. Para definir un formato distinto seleccione el botón de opción *Dar formato*.
4. Ahora puede seleccionar un formato de fecha predefinido en la lista desplegable del cuadro combinado (imagen siguiente) o definir un formato personal en el campo de entrada del cuadro combinado. Para más información sobre la sintaxis que debe usar consulte el apartado Sintaxis 383.

Usar el formato de valor para dar formato a cálculos automáticos

Cuando un cálculo automático da como resultado un valor que está en un formato de fecha léxico, puede usar la característica *Formato de valor* para dar formato a la representación gráfica del resultado.

1. Seleccione el cálculo automático en el diseño.
2. En la ventana Propiedades seleccione el componente **`contenido`** y después expanda el grupo de propiedades *Cálculo automático*. Ahora haga clic en el botón **Edición** ... de la propiedad `formato de valor`. Esto abre el cuadro de diálogo "Formato de valor" (*imagen siguiente*).



La opción predeterminada es la opción *Sin formato* (el formato léxico estándar del tipo de datos del nodo seleccionado).

3. Para definir un formato distinto seleccione el botón de opción *Dar formato*.
4. En el panel *Opciones para el valor de XML Schema*, haga clic en el primer cuadro combinado (*El tipo de datos en la instancia XML es*) y seleccione `date`, que es el tipo de datos del resultado del cálculo automático. Después, en el segundo cuadro combinado (*Dar formato en documentos de salida como*) puede definir su propio formato para el resultado del cálculo automático. Para más información sobre la sintaxis que debe usar consulte el apartado Sintaxis 383.

Aplicar el formato de valor en el documento de salida

El formato de valor definido se aplica a la presentación de los datos en la vista Authentic pero algunas definiciones de formato de valor (no todas) también se aplican a los documentos de salida HTML, texto y RTF. No obstante, para ello es necesario marcar la casilla *Aplicar el mismo formato al resultado XSLT* situada al final del cuadro de diálogo "Formato de valor". Si no marca esta casilla o si la definición de formato de valor no es compatible con esto, solamente la vista Authentic usará el formato de valor definido para presentar los datos. En los documentos de salida HTML, texto y RTF el valor aparecerá en su formato léxico (si se trata de un nodo) o en el formato del resultado del cálculo automático (si se trata de un cálculo automático).

© 2016-2022 Altova GmbH

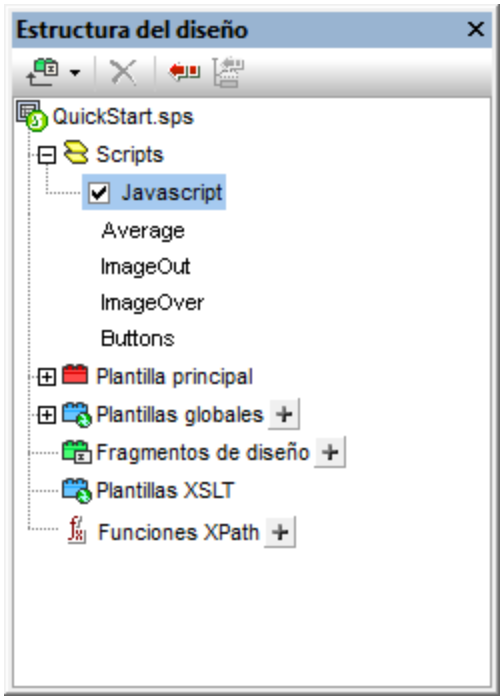
9.9 Usar scripts

En StyleVision puede definir funciones JavaScript para cada diseño SPS con ayuda del editor de JavaScript (al que puede acceder desde la pestaña de la vista **Diseño**). Las funciones definidas en el editor de JavaScript se almacenan en el encabezado del documento HTML y se les invoca desde el cuerpo del documento HTML. Puede usar este tipo de funciones para:

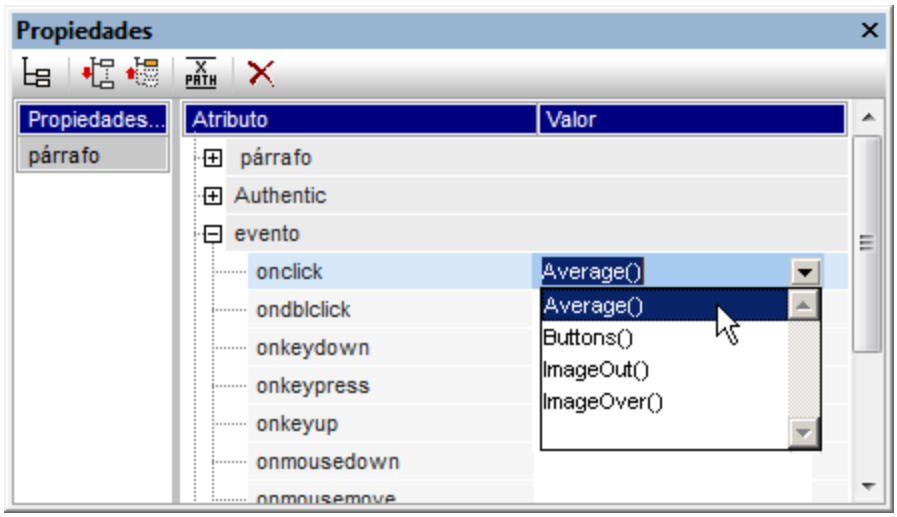
- Conseguir un resultado complejo mezclando varias instrucciones de scripts. Si este es su objetivo, recomendamos escribir todos los scripts en funciones distintas y en una posición (en el encabezado) y hacer referencia a las funciones en el documento de diseño.
- Usar un script concreto en varias posiciones del documento de diseño.

El apartado Definir funciones JavaScript 498 explica cómo crear las funciones en el editor de JavaScript de StyleVision.

Todas las funciones JavaScript definidas para un diseño SPS en el editor de JavaScript aparecen en la ventana Estructura del diseño, en la lista `Scripts`. Por ejemplo, en la imagen siguiente puede ver que en el diseño SPS activo se definieron cuatro funciones JavaScript: `Average`, `ImageOut`, `ImageOver` y `Buttons`.



Las funciones definidas en el editor de JavaScript están disponibles en la interfaz gráfica del usuario como llamadas a controladores de eventos. Si selecciona un componente del diseño, podrá asignar cualquiera de estas funciones a una propiedad de controlador de eventos del componente seleccionado (ventana Propiedades, grupo de propiedades *evento*) (*imagen siguiente*). Para aprender a asignar una función JavaScript a un controlador de eventos consulte el siguiente apartado Asignar funciones a controladores de eventos 499.



Nota: los scripts solamente afectan a los documentos de salida HTML. No son compatibles con la vista Authentic.

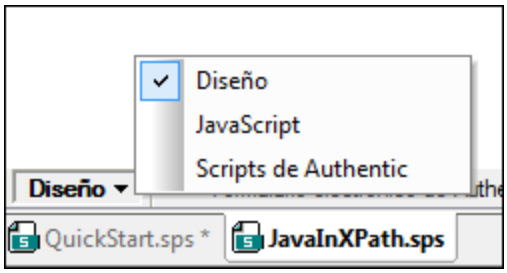
Scripts en archivos SPS modulares

Cuando se añade un módulo SPS a un archivo SPS 251, los scripts del módulo SPS importado están disponibles en el archivo SPS de destino y se pueden usar como controladores de eventos (desde la ventana Propiedades) en el archivo SPS de destino. Para más información consulte la sección Archivos SPS modulares 251.

9.9.1 Definir funciones JavaScript

Siga estos pasos para definir funciones JavaScript:

1. En la vista **Diseño** cambie al editor de JavaScript haciendo clic en la pestaña **Diseño** y seleccionando **JavaScript** en el menú que aparece (*imagen siguiente*).



2. En el editor de JavaScript escriba las definiciones de función (*imagen siguiente*).

 © 2016-2022 Altova GmbH

```
1   function DisplayTime()
2   {
3   now = new Date();
4   hours = now.getHours();
5   mins = now.getMinutes();
6   secs = now.getSeconds();
7   result = hours + ":" + mins + ":" + secs;
8   alert(result);
9   }
10
11  function ClearStatus()
12  {
13  window.status="";
14  }
```

En el ejemplo de la imagen anterior, por ejemplo, hay dos funciones JavaScript: `DisplayTime` y `ClearStatus`. Estas dos funciones se insertarán en el encabezado del archivo HTML de la siguiente forma:

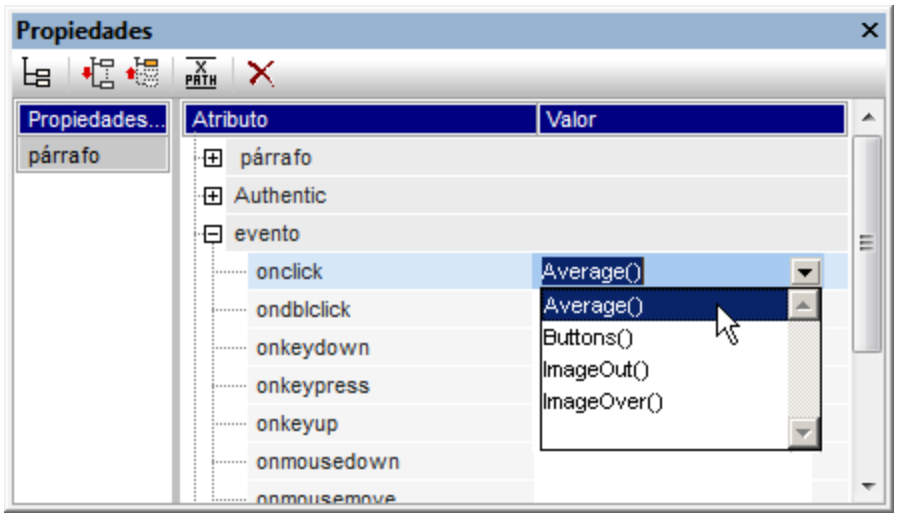
```
<script language="javascript">

<!-- function DisplayTime()
{
        now = new Date();
        hours = now.getHours();
        mins = now.getMinutes();
        secs = now.getSeconds();
        result = hours + "." + mins + "." + secs;
        alert(result)
}

function ClearStatus()
{
        window.status="";
}
-->

</script>
```

Ahora puede llamar a estas funciones desde cualquier posición del documento HTML. En StyleVision todas las funciones definidas en el editor se pueden asignar a una propiedad de controlador de eventos del grupo de propiedades *evento* (ventana Propiedades). Para más información consulte el apartado Asignar funciones a controladores de eventos 499.

9.9.2 Asignar funciones a controladores de eventos

En StyleVision puede asignar funciones JavaScript a controladores de eventos para aquellos eventos que aparezcan en la representación HTML de componentes del diseño SPS. Estos controladores de eventos se usarán en los documentos de salida HTML. El controlador de un evento (p. ej. `onclick`) se define asignando una función global como controlador de eventos. En la ventana Propiedades las funciones globales definidas en

el editor de JavaScript están disponibles como controladores de eventos (en las listas desplegables de los cuadros combinados de las propiedades) (*imagen siguiente*).



Siga estos pasos para asignar una función a un controlador de eventos:

1. Seleccione el componente del diseño SPS cuyo controlador de eventos desea definir. El componente puede ser un nodo o contenido de cualquier tipo, ya sea dinámico o estático.
2. En la ventana Propiedades expanda el grupo *evento*. Bajo este grupo de propiedades aparecen todos los eventos disponibles (*imagen anterior*).
3. En la columna *Valor* de cada evento haga clic en la punta de flecha para abrir la lista desplegable del cuadro combinado. La lista incluye todas las funciones definidas en el editor de JavaScript.
4. Seleccione la función que desea usar como controlador de eventos del evento.

La función JavaScript se ejecuta en el documento HTML de salida cuando se desencadena el evento del componente para el que se definió el controlador de eventos.

9.9.3 Archivos JavaScript externos

Hay dos maneras de acceder a archivos JavaScript externos desde un diseño SPS:

1. Creando un elemento definido por el usuario o un bloque XML definido por el usuario 500. Estos objetos de diseño pueden incluir un elemento `SCRIPT` que accede al archivo JavaScript externo. Recuerde que el elemento definido por el usuario o bloque XML definido por el usuario debe estar dentro del elemento `BODY` del diseño (y, por tanto, dentro del elemento `BODY` del documento HTML de salida, no en el elemento `HEAD`).
2. Creando un script en el editor de JavaScript 501 que accede al archivo externo. El script estará dentro del elemento `HEAD` del documento HTML de salida.

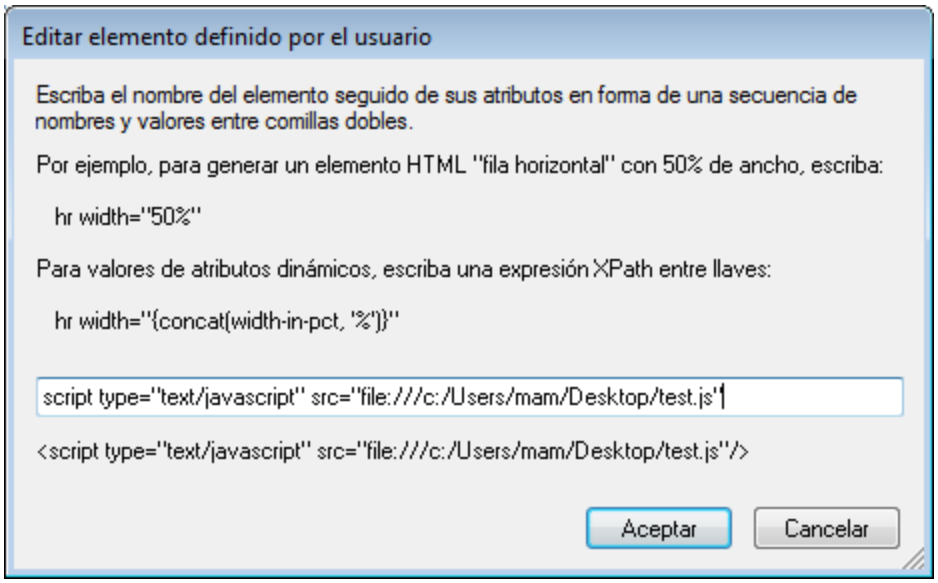
Elementos definidos por el usuario y bloques XML definidos por el usuario

Para acceder a archivos JavaScript externos puede usar elementos definidos por el usuario 151 y bloques XML definidos por el usuario 152. Con ayuda de estos objetos podrá colocar un elemento `SCRIPT` que acceda al archivo JavaScript externo en cualquier posición del elemento `BODY` del documento HTML de salida.

 © 2016-2022 Altova GmbH

Para insertar un elemento definido por el usuario[151]:

1. Ponga el cursor en la posición del diseño donde desea insertar el elemento `SCRIPT` que debe acceder al archivo JavaScript externo.
2. Seleccione el comando de menú **Insertar | Insertar elemento definido por el usuario | Elemento definido por el usuario**. Este comando también está disponible en el menú contextual que aparece al hacer clic con el botón derecho.



3. Se abre el cuadro de diálogo "Editar elemento definido por el usuario" (*imagen anterior*). Inserte el elemento `SCRIPT` tal y como muestra la imagen: use la URL del archivo JavaScript como valor del atributo `src` del elemento `SCRIPT`. Por ejemplo: `script type="text/javascript" src="file:///c:/Usuarios/mam/Desktop/test.js"`
4. Para terminar haga clic en **Aceptar**.

El mismo resultado puede obtenerse con un bloque XML definido por el usuario[152]. Para ello inserte un bloque XML definido por el usuario[152] en lugar de un elemento definido por el usuario[151] e inserte el elemento `SCRIPT` como bloque XML complejo, es decir, con etiquetas de apertura y cierre.

Editor de JavaScript

El editor de JavaScript[498] sirve para insertar scripts externos en el elemento `HEAD` del documento HTML de salida. Para ello debe insertar el fragmento de script que aparece a continuación en el editor de JavaScript, fuera de las definiciones de función (si las hay).

```
var script = document.createElement('script');
script.type = 'text/javascript';
script.src = 'file:///c:/Users/Desktop/test.js';
var head = document.getElementsByTagName('head')[0];
head.appendChild(script)
```

Al archivo JavaScript externo que localiza la URL de `script.src` se puede acceder desde el elemento `HEAD` del documento HTML de salida.

9.10 Importación HTML

En StyleVision puede crear estos documentos a partir de un documento HTML importado:

- Un documento SPS basado en el diseño y en la estructura del archivo HTML importado.
- Un esquema XML cuyos elementos o atributos se crean a partir de los componentes del documento HTML. En el esquema definido por el usuario también puede crear elementos y atributos adicionales sin relación alguna con el documento HTML.
- Un documento XML (i) cuya estructura está basada en el esquema XML creado y (ii) cuyo contenido procede del archivo HTML.
- Hojas de estilos XSLT basadas en el diseño activo en la vista **Diseño**.

Convertir HTML en XML paso a paso

El mecanismo de importación HTML, que permite crear archivos XML a partir del archivo HTML importado, se divide en estos pasos:

1. Primero se crea un archivo SPS nuevo a partir de un archivo HTML 502. Cuando se importa un archivo HTML en StyleVision, se crea un documento SPS nuevo. El documento HTML aparece en la vista **Diseño** con las etiquetas de marcado HTML. En la ventana Estructura del esquema se crea un esquema XML vacío con un elemento de documento llamado `UserRoot`. Este es el esquema en el que se basa el nuevo diseño SPS. El contenido y el marcado del documento HTML que aparecen en la vista **Diseño** se incluye en el diseño SPS como contenido estático.
2. Después se crea el esquema definido por el usuario y se diseña el SPS 504. El esquema se amplía (i) arrastrando componentes del documento HTML hasta la posición deseada de la Estructura del esquema y (ii) añadiendo nodos definidos por el usuario a la Estructura del esquema. En la vista **Diseño** el contenido HTML utilizado para generar nodos en la Estructura del esquema ahora aparece entre etiquetas de nodo de esquema. El contenido HTML que no tenga un nodo de esquema seguirá representado en la vista **Diseño** sin etiquetas de nodo.
3. El siguiente paso es dar formato a los nodos del documento de diseño, mejorar las reglas de procesamiento y añadir contenido estático. Estos cambios solamente se reflejarán en el diseño SPS y en la hoja de estilos XSLT generada. Los cambios no afectan ni al esquema generado ni al archivo XML.
4. Por último, cuando termine de diseñar la Estructura del esquema y el SPS, puede generar y guardar 508:

   - un esquema XML correspondiente a la Estructura del esquema que diseñó;
   - un archivo de datos XML cuya estructura esté basada en el esquema y en el contenido de los nodos de esquema que se crean con los marcadores de posición `(contenido)` del diseño SPS;
   - un diseño SPS (`.sps`) y hojas de estilos XSLT basadas en el diseño.

9.10.1 Crear archivo SPS a partir de un archivo HTML

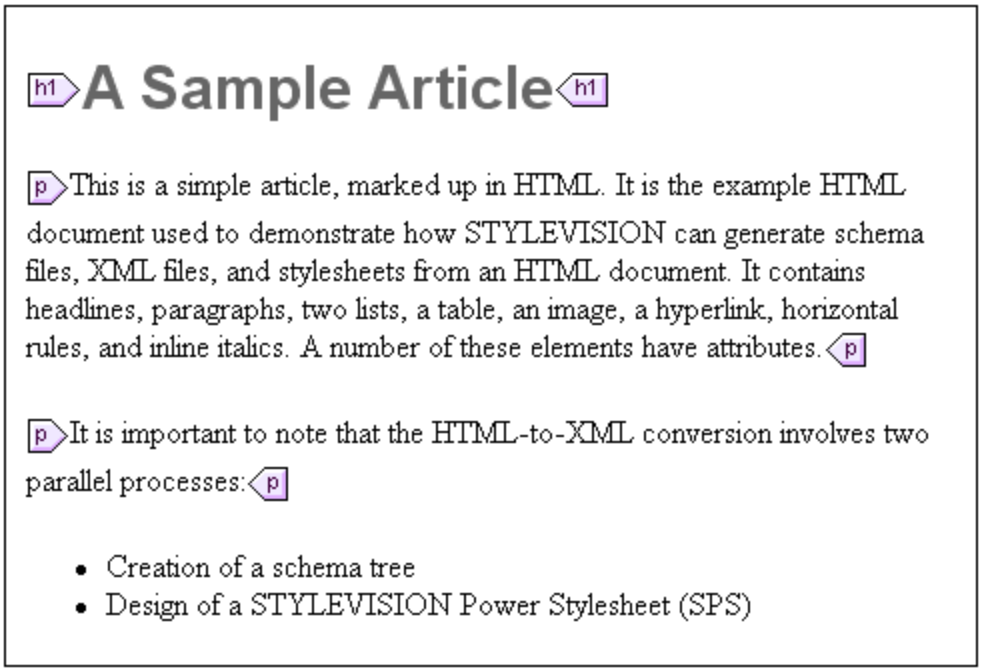
Siga estas instrucciones para crear un archivo SPS nuevo a partir de un documento HTML:

1. Seleccione el comando de menú **Archivo | Nuevo | Nuevo a partir de un archivo HTML**.
2. Aparece el cuadro de diálogo "Abrir". Busque el archivo HTML que desea importar, selecciónelo y haga clic en **Abrir**.

 © 2016-2022 Altova GmbH

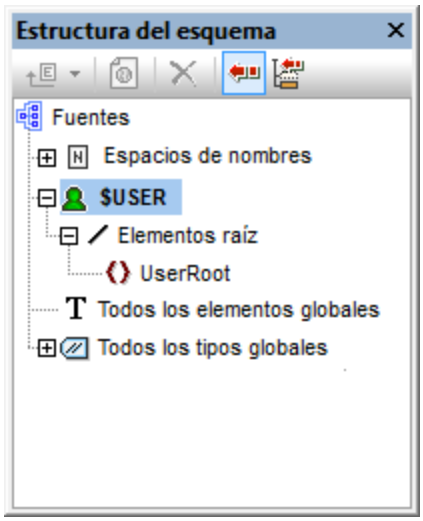
3. StyleVision le pregunta si quiere convertir las rutas de acceso relativas del archivo HTML en rutas de acceso absolutas. Decídase por una de las dos opciones y haga clic en **Sí** o **No** según corresponda.

Se crea un documento SPS, que aparece en la vista **Diseño**. El documento SPS está marcado con los formatos HTML predefinidos de StyleVision (*imagen siguiente*).



Observe que el documento HTML aparece dentro de la plantilla principal y que no hay una plantilla global.

En la ventana Estructura del esquema se crea un esquema definido por el usuario (*imagen siguiente*), cuyo elemento raíz (elemento de documento) se llama `UserRoot`.



Observe que la lista **`Todos los elementos globales`** está vacía (es decir, no hay elementos globales).

Estructura y diseño del documento SPS

El documento SPS contiene una sola plantilla, la plantilla principal, que se aplica al nodo de documento de un documento XML interno temporal. Este documento XML tiene la estructura del esquema definida por el usuario en la ventana Estructura del esquema. Llegados a este punto:

- en la vista **Diseño** los componentes del documento HTML aparecen dentro de la plantilla principal del SPS en forma de componentes estáticos.
- en la vista **Authentic** la representación de estos componentes HTML será una representación no editable con contenido no XML.
- las **hojas de estilos XSLT** contienen estos componentes HTML como elementos de resultado literal.
- el **esquema** solo tiene el elemento de documento `UserRoot` y, por tanto, el documento XML interno temporal solo contiene el elemento de documento `UserRoot` sin nodos secundarios.

Ahora puede seleccionar componentes HTML en la vista **Diseño** y convertirlos en elementos/atributos del esquema definido por el usuario. Hay dos maneras de hacer esto:

1. **Convirtiendo** el componente seleccionado en un elemento o atributo: En la vista **Diseño** se inserta el marcador de posición `(contenido)` entre las etiquetas del nodo. En el esquema se crea un elemento o atributo. En el documento XML se crea el nodo y la selección se convierte en el contenido de texto del nodo recién creado. El contenido del nodo creado en el documento XML se insertará de forma dinámica en los documentos de salida que se generarán con el SPS.
2. **Incluyendo** el componente seleccionado dentro de un elemento o atributo. En la vista **Diseño** se inserta la selección entre las etiquetas del nodo pero no se inserta el marcador de posición `(contenido)`. Esto significa que la selección aparece en el diseño SPS como contenido estático. En el esquema se crea un elemento o atributo. En el documento XML se crea el nodo pero está vacío. El texto estático que está dentro de las etiquetas de nodo de esquema aparecerá en los documentos de salida pero no se generará contenido dinámico para este nodo a no ser que se inserte el marcador de posición `(contenido)`.

Por tanto, el marcador de posición `(contenido)` tiene un papel importante porque indica en qué posiciones del diseño se mostrarán datos dinámicos del documento XML (en los documentos de salida) y en qué posiciones se podrán editar los datos del documento XML (en la vista Authentic).

9.10.2 Crear el esquema y el diseño SPS

El esquema definido por el usuario se crea arrastrando componentes desde la vista **Diseño** hasta el esquema en la ventana Estructura del esquema. Para esto es necesario arrastrar los componentes uno por uno. El componente seleccionado se coloca en un nodo de la Estructura del esquema y se crea como nodo secundario o del mismo nivel. Al soltar el componente sobre la Estructura del esquema puede elegir qué tipo de nodo se crea (un elemento o un atributo) y si la selección se debe convertir en el nodo nuevo o si, por el contrario, se inserta dentro del nodo nuevo.

Componentes que se pueden seleccionar para crear nodos en el esquema

Estos son los componentes de la vista **Diseño** con los que puede crear nodos en el esquema:

- Nodos del documento HTML.
- Cadenas de texto situadas dentro de un nodo.
- Cadenas de texto adyacentes que forman parte de nodos distintos.

 © 2016-2022 Altova GmbH

- Imágenes.
- Hipervínculos.
- Tablas.
- Listas.
- Cualquier combinación de estos componentes.

Este apartado explica paso a paso cómo crear nodos en el esquema a partir de los componentes de la vista **Diseño**. El uso de tablas y listas se describe por separado en el apartado Crear elementos/atributos a partir de tablas y listas 506.

Para empezar haga clic en el componente HTML en la vista **Diseño** o resalte la cadena de texto que desea usar. Si quiere puede seleccionar varios componentes (haciendo clic y arrastrando el ratón por encima de los componentes). No olvide que StyleVision amplía la selección al principio y al final seleccionando elementos de nivel superior hasta que el primer y el último elemento seleccionados pertenezcan al mismo elemento primario.

Posición en la Estructura del esquema

Al arrastrar y pasar la selección por encima de un nodo de la Estructura del esquema, el puntero del ratón cambia y se convierte en uno de estos iconos:

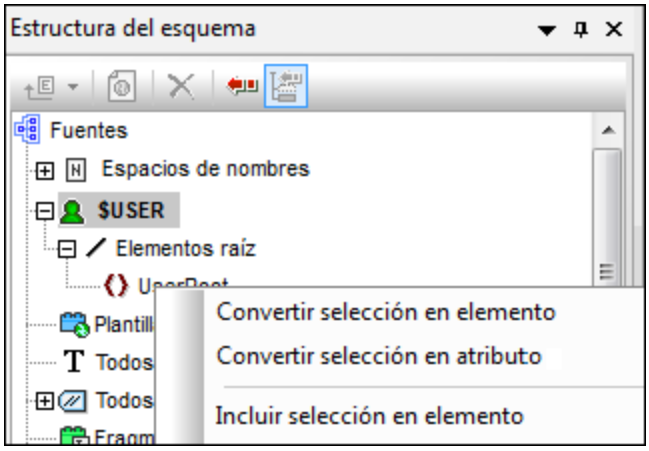
- **Crear como elemento del mismo nivel** *:* si suelta la selección cuando aparece este icono, la selección crea un nodo del mismo nivel que el nodo sobre el que se suelta la selección.
- **Crear como elemento secundario** *:* si suelta la selección cuando aparece este icono, la selección crea un nodo secundario del nodo sobre el que se suelta la selección.

Por tanto, es importante pensar sobre qué nodo quiere soltar la selección y si quiere crear un secundario o un elemento del mismo nivel del nodo.

Cómo crear el nodo

Al soltar la selección (*ver más arriba*) aparece un menú contextual (*imagen siguiente*) con dos opciones principales:

- Crear un elemento o atributo a partir del nodo
- Incluir la selección en el nodo



Es necesario tener en cuenta estos aspectos:

- Cuando se crea un nodo (elemento o atributo) a partir de la selección, en la vista **Diseño** la selección se sustituye con las etiquetas de nodo y el marcador de posición `(contenido)`. En el diseño y en los documentos de salida el contenido de texto de la selección se elimina del contenido estático. En los documentos de salida el texto de la selección aparece como contenido dinámico del nodo del documento XML.
- Si convierte un nodo HTML en un nodo XML, las etiquetas de nodo XML se insertan dentro de las etiquetas de nodo HTML.
- Cuando se incluye la selección (incluidos los nodos HTML) dentro de un nodo XML, las etiquetas de nodo XML se insertan antes y después de la selección. En el diseño y en los documentos de salida el contenido de texto de la selección se conserva como texto estático.
- Las etiquetas de nodo insertadas se insertan con la ruta necesaria (es decir, con las etiquetas del nodo antecesor que establecen una ruta relativa al nodo contenedor). La ruta puede ser absoluta o relativa, dependiendo del contexto del nodo en el diseño.
- Para aprender a crear nodos a partir de tablas y listas consulte el apartado Crear elementos/atributos a partir de tablas y listas 506.

Agregar y eliminar nodos en el esquema

Si quiere puede seguir añadiendo nodos al esquema definido por el usuario que no estén basados en componentes HTML del diseño. Para ello haga clic con el botón derecho en un nodo y seleccione el comando pertinente en el menú contextual. Si lo prefiere, puede usar los iconos de la barra de herramientas de la ventana Estructura del esquema.

Para eliminar un nodo, selecciónelo y después use el menú contextual o el icono de la barra de herramientas. Sin embargo, tenga en cuenta que al eliminar un nodo algunas rutas del diseño pueden dejar de ser válidas.

Modificar el diseño

Para modificar la estructura del diseño arrastre los componentes por la vista hasta su nueva posición e inserte componentes estáticos y dinámicos. También puede modificar los estilos con las funciones de estilo de StyleVision.

9.10.3 Crear elementos/atributos a partir de tablas y listas

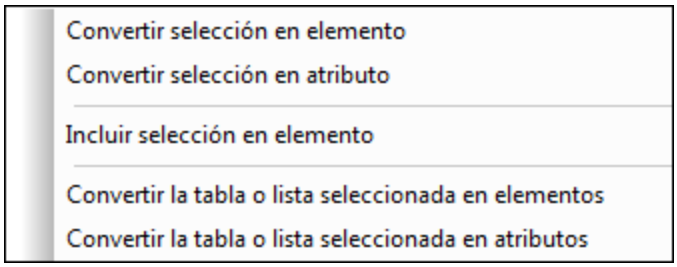
Las listas y tablas del documento HTML se pueden convertir en nodos de elemento y atributo del esquema XML a fin de conservar la estructura de lista o de tabla en el esquema.

Convertir una tabla en elementos o atributos

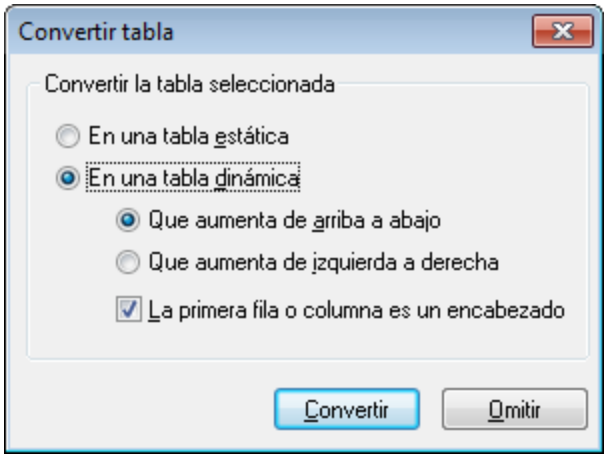
Siga estos pasos para convertir una tabla en nodos del esquema:

1. Seleccione la tabla HTML (basta con resaltar parte del texto de la tabla).
2. Arrastre la selección hasta el nodo de la Estructura del esquema donde quiere crear un nodo secundario o del mismo nivel.
3. Cuando aparezca el icono **Crear como elemento del mismo nivel** o **Crear como secundario**, suelte la selección.
4. Aparece un menú contextual (*imagen siguiente*) donde puede seleccionar el comando **Convertir la tabla o lista seleccionada en elementos** o **Convertir la tabla o lista seleccionada en atributos**.

 © 2016-2022 Altova GmbH



5. Tras hacer clic en el comando se abre el cuadro de diálogo "Convertir tabla" (*imagen siguiente*). Seleccione si la tabla se convierte en una tabla estática o dinámica.



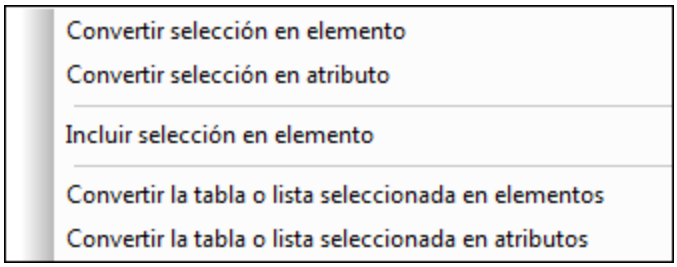
Si selecciona la opción **tabla estática**, por cada celda de la tabla se creará un nodo de esquema. En la vista **Diseño** cada nodo se inserta con el marcador de posición `(contenido)`. Los datos de las celdas de la tabla se copian en el documento XML interno temporal (y en el documento XML generado). La opción **tabla dinámica** solo está disponible si todas las filas de la tabla tienen la misma estructura. En la vista **Diseño** las filas de la tabla dinámica se representan con una sola fila (porque todas las filas tienen la misma estructura). Los datos de la tabla se copian al archivo XML. La tabla dinámica puede aumentar de arriba a abajo (las filas se organizan en vertical) o de izquierda a derecha (las filas se convierten en columnas y se expanden de izquierda a derecha. Si marca la casilla *La primera fila o columna es un encabezado,* entonces (i) se incluye una fila de encabezado con los encabezados de columna como texto estático y (ii) los nodos del esquema reciben como nombre el texto del encabezado. Si la primera fila/columna no se marca como encabezado, entonces no se incluye una fila de encabezado en la tabla.

6. Cuando termine de configurar la tabla haga clic en **Convertir**.

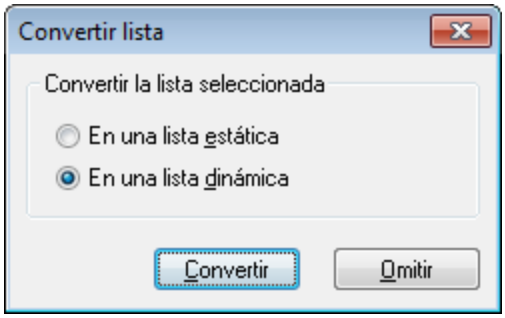
Convertir una lista en elementos o atributos

Siga estos pasos para convertir una lista en nodos del esquema:

1. Seleccione la lista HTML (basta con resaltar parte del texto de la lista).
2. Arrastre la selección hasta el nodo de la Estructura del esquema donde quiere crear un nodo secundario o del mismo nivel.
3. Cuando aparezca el icono **Crear como elemento del mismo nivel** o **Crear como secundario**, suelte la selección sobre el nodo.
4. Aparece un menú contextual (*imagen siguiente*) donde puede seleccionar el comando **Convertir la tabla o lista seleccionada en elementos** o **Convertir la tabla o lista seleccionada en atributos**.



5. Tras hacer clic en el comando se abre el cuadro de diálogo "Convertir lista" (*imagen siguiente*) donde puede seleccionar si se crea una lista estática o dinámica.



Si selecciona la opción **lista estática**, por cada elemento de la lista se crea un nodo de esquema. En la vista **Diseño** cada nodo se inserta con el texto del elemento de lista HTML como contenido estático. Si selecciona la opción lista dinámica, los elementos de la lista se representan con un solo nodo en la vista **Diseño**. El elemento de lista incluye también el marcador de posición `(contenido)`.

6. Elija una opción y haga clic en **Convertir** para terminar.

9.10.4 Generar resultados

Cuando termine de diseñar el SPS puede generar estos archivos de salida con el comando **Archivo | Guardar archivos generados**:

- El esquema definido por el usuario (es decir, el esquema que creó en la ventana Estructura del esquema).
- El archivo XML definido por el usuario (es decir, un documento XML basado en el esquema creado que contiene los datos importados del archivo HTML).
- Hojas de estilos para documentos de salida HTML, texto y RTF.
- Documentos de salida HTML, texto y RTF.

 © 2016-2022 Altova GmbH

9.11 Interfaces ASPX para aplicaciones web

Si necesita crear un diseño SPS para un informe HTML de datos XML o de BD para Internet, siga estos pasos en StyleVision:

1. Si los datos de origen están en una BD, cuando termine de diseñar el SPS en StyleVision genere un archivo XML a partir de la BD. (Si los datos de origen están en un archivo XML entonces puede saltarse este paso.)
2. Genere también un archivo XSLT-para-HTML.
3. Transforme el archivo XML con ayuda del archivo XSLT-para-HTML generado.
4. Ponga el archivo HTML resultante en el servidor.

En las aplicaciones web el archivo HTML podría estar obsoleto si los datos de origen (BD o XML) se modificaron. Para actualizar el archivo HTML en el servidor web con los datos nuevos sería necesario (i) volver a generar el archivo XML a partir de la BD, (ii) transformar el archivo XML nuevo con ayuda del archivo XSLT-para-HTML generado y (iii) poner el resultado HTML en el servidor.

StyleVision ofrece una práctica solución para actualizar páginas web HTML rápidamente. Se trata de una característica que genera una aplicación ASPX automáticamente. Es decir, StyleVision genera todos los archivos necesarios de la aplicación ASPX (los archivos ASPX, XSLT y de código). Estos archivos se pueden poner en el servidor junto con el archivo de BD o el archivo XML de origen y con el archivo XSLT-para-HTML generado. Cada vez que se actualiza el archivo ASPX (el archivo de la interfaz web) ocurre esto: (i) si son datos de BD, se genera un archivo XML nuevo a partir de la BD (este paso se omite si son datos XML), (ii) el archivo XML se transforma con el archivo XSLT-para-HTML que está en servidor y (iii) el resultado de la transformación aparece en la página de la interfaz web. De este modo la página de la interfaz web mostrará rápidamente la versión más reciente de la BD o de los datos XML.

Generar archivos para una solución ASPXC

Tras crear el archivo SPS basado en una BD o en un archivo XML, siga estos pasos para crear una solución ASPX:

1. Abra el archivo SPS en StyleVision y haga clic en el comando de menú **Archivo | Diseño web | Generar aplicación web ASPX**. Los archivos de aplicaciones ASPX se crean en la carpeta indicada por el usuario. La carpeta en la que se genera la aplicación ASPX contendrá, entre otros, estos archivos:

   - `Readme.doc`
   - `NombreArchivoSPS.aspx`
   - `NombreArchivoSPS.xslt`
   - `NombreArchivoSPS.cs`

2. Ponga el archivo de BD o XML en el servidor, en la misma carpeta que la aplicación ASPX. El archivo `.aspx` es el punto de entrada de la aplicación. Si se actualiza este archivo, se actualizan también los datos XML o de BD representados en la página de la interfaz web.

Nota: para que la transformación XSLT se ejecute correctamente debe instalar la aplicación RaptorXML de Altova. Si tiene problemas con la transformación, consulte el archivo `ReadMe.doc` que incluye información sobre cómo configurar RaptorXML.

Funcionamiento

La carpeta en la que se genera la aplicación ASPX contendrá, entre otros, estos archivos:

- `Readme.doc`
- `NombreArchivoSPS.aspx`
- `NombreArchivoSPS.xslt`
- `NombreArchivoSPS.cs`

`NombreArchivoSPS.aspx` es la URL del documento de salida. `NombreArchivoSPS.aspx` ejecuta el código C# almacenado en el archivo `NombreArchivoSPS.cs`. Este código C# lee el contenido XML (de archivos o de una BD, según corresponda) y se lo envía a RaptorXML junto con el archivo `NombreArchivoSPS.xslt`. (RaptorXML contiene el motor de transformación XSLT de Altova y se puede descargar del sitio web de Altova). RaptorXML transforma el contenido XML con ayuda del archivo XSLT suministrado. El resultado es un documento HTML que la aplicación web muestra en el explorador. Cuando cambia el contenido XML (porque se realicen cambios en la BD, por ejemplo) y se navegue a la página `NombreArchivoSPS.aspx` (o se actualice la página en el explorador web), se recuperan automáticamente los datos más recientes de la BD o del archivo XML y se presenta un documento actualizado en el explorador.

9.11.1 Ejemplo: host local en Windows 7

A continuación explicamos cómo configurar su host local para que sirva una aplicación ASPX. Para más información consulte el archivo `Readme.doc` de la carpeta de la aplicación ASPX. Esta carpeta y archivo se generan al seleccionar el comando **Archivo | Diseño web | Generar aplicación web ASPX**.

Instalar **RaptorXML**

Compruebe que tiene instalada la versión más reciente de RaptorXML. Esta aplicación contiene el motor de transformación de Altova y se usará para transformar el archivo XML (o el archivo XML generado a partir de la BD).

Activar Internet Information Services (el servidor web de Microsoft)

Si Microsoft Internet Information Services (IIS) no está activado, a continuación tiene las instrucciones para activarlo. El paso 5 muestra cómo probar si IIS está activo.

1. Haga clic en **Panel de control | Programas y características | Activar o desactivar las características de Windows**.
2. Marque la casilla *Internet Information Services*. La casilla de tres estados estará parcialmente marcada.
3. Ahora marque la casilla *Internet Information Services* | *Servicios World Wide Web* | *Características de desarrollo de aplicaciones* | *ASP.NET*.
4. Haga clic en **Aceptar**. Cuando termine el proceso, tendrá la carpeta `C:/inetpub/wwwroot`. Esta es la carpeta raíz del servidor web.
5. Pruebe a abrir `localhost` en su explorador web. Debería aparecer la pantalla de bienvenida de Internet Information Services.

© 2016-2022 Altova GmbH

Generar la aplicación ASPX en StyleVision

1. Recomendamos guardar la BD y el archivo SPS en la misma carpeta.
2. Cuando termine de diseñar el documento SPS, haga clic en el comando de menú **Archivo | Diseño web | Generar aplicación web ASPX**.
3. En el cuadro de diálogo que aparece cree una carpeta dentro de `C:/inetpub/wwwroot` y selecciónela (p. ej. `C:/inetpub/wwwroot/Test1`.)
4. Haga clic en **Aceptar** y StyleVision genera estos archivos en la carpeta elegida: `<NombreArchivo>.aspx`, `<NombreArchivo>_AltovaDataBaseExtractor.cs` y `Web.config` y la carpeta `App_Code` con varios archivos más.

Nota: para poder generar los archivos en `C:/inetpub/wwwroot` debe ejecutar StyleVision como administrador. Para ello haga clic con el botón derecho en el ejecutable de StyleVision o en su acceso directo y seleccione el comando **Ejecutar como administrador**.

Informar a ASPX de la aplicación generada

Para que ASPX conozca la aplicación generada con StyleVision:

1. Haga clic en **Panel de control** | **Herramientas administrativas** | **Administrador de Internet Information Services (IIS)**.
2. En el panel *Conexiones* expanda el árbol para ver la carpeta (p. ej. `Test1`). El icono de la carpeta es una carpeta amarilla estándar.
3. En el menú contextual de la carpeta, ejecute el comando **Convertir en aplicación** y haga clic en **Aceptar** en el diálogo. El icono de la carpeta ahora tiene forma de esfera.
4. En el panel *Conexiones* expanda el árbol para ver la entrada *Grupo de aplicaciones*. Seleccione esta entrada.
5. En el menú contextual de *DefaultAppPool* (disponible en el panel principal al seleccionar *Grupo de aplicaciones* en el panel *Conexiones*), seleccione el comando **Configuración avanzada**.
6. Para la propiedad *Identidad*, seleccione *Cuenta personalizada* e introduzca su usuario y contraseña de Windows.
7. Para la propiedad *Habilitar aplicaciones de 32 bits* escriba el valor `True`. (Para que puedan tener acceso los controladores de BD). Este paso solo es necesario si su versión de Windows es Windows 7 de 64 bits.

Ejecutar la aplicación-.

Escriba la dirección `localhost/Test1/<NombreArchivo>.aspx` en su explorador web (suponiendo que `Test1` sea el nombre de la carpeta donde se guardó la aplicación ASPX). El HTML transformado aparecerá en el explorador. Al actualizar esta página ASPX aparecerán los datos más recientes de la BD o del archivo XML.

Nota: si el explorador se queda colgado, compruebe que RaptorXML tiene asignada una licencia correctamente.

9.12 Archivo PXF: contenedor para el archivo SPS y archivos relacionados

Los diseños SPS que utilizan XSLT 2.0 o XSLT 3.0 se pueden guardar como formularios XML portátiles (formato de archivo PXF). El formato PXF es un formato de archivo especial desarrollado por Altova para empaquetar diseños SPS junto con archivos relacionados (como esquemas, archivo XML de trabajo, imágenes, archivos XSLT de transformación, etc.). La ventaja del formato de archivo PXF es que permite distribuir con un solo archivo todos los ficheros necesarios para editar un documento en la vista Authentic y para generar documentos de salida desde esta vista.

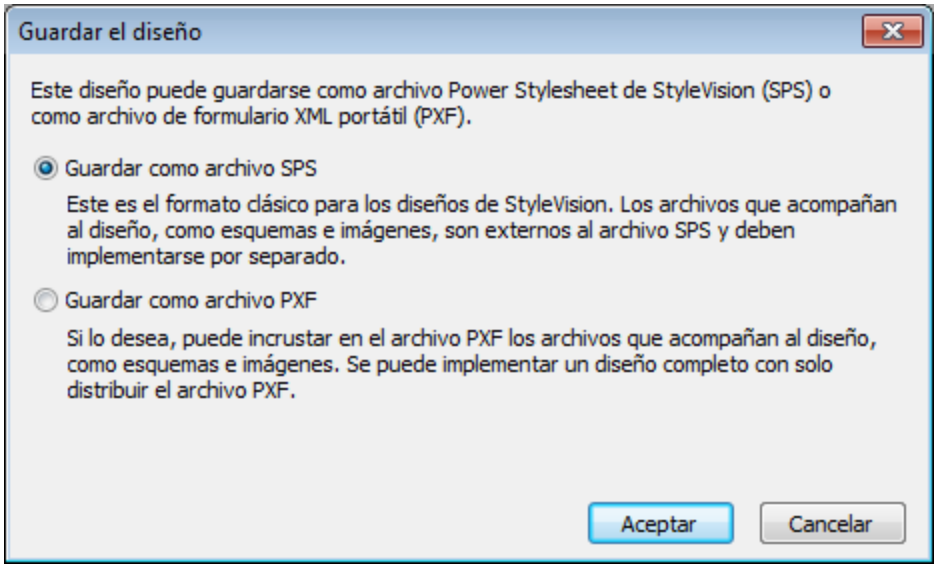
Esta sección describe el uso de los archivos PXF y se divide en tres apartados:

- Crear un archivo PXF 512
- Editar un archivo PXF 515
- Implementar un archivo PXF 516

Nota: solo se pueden guardar en formato PXF los archivos SPS diseñados con XSLT 2.0 XSLT 3.0.

9.12.1 Crear un archivo PXF

Para crear un archivo PXF a partir del diseño SPS y sus archivos relacionados abra el diseño SPS en StyleVision y seleccione el comando **Archivo | Guardar como**. Esto abre el cuadro de diálogo "Guardar el diseño" (*imagen siguiente*).



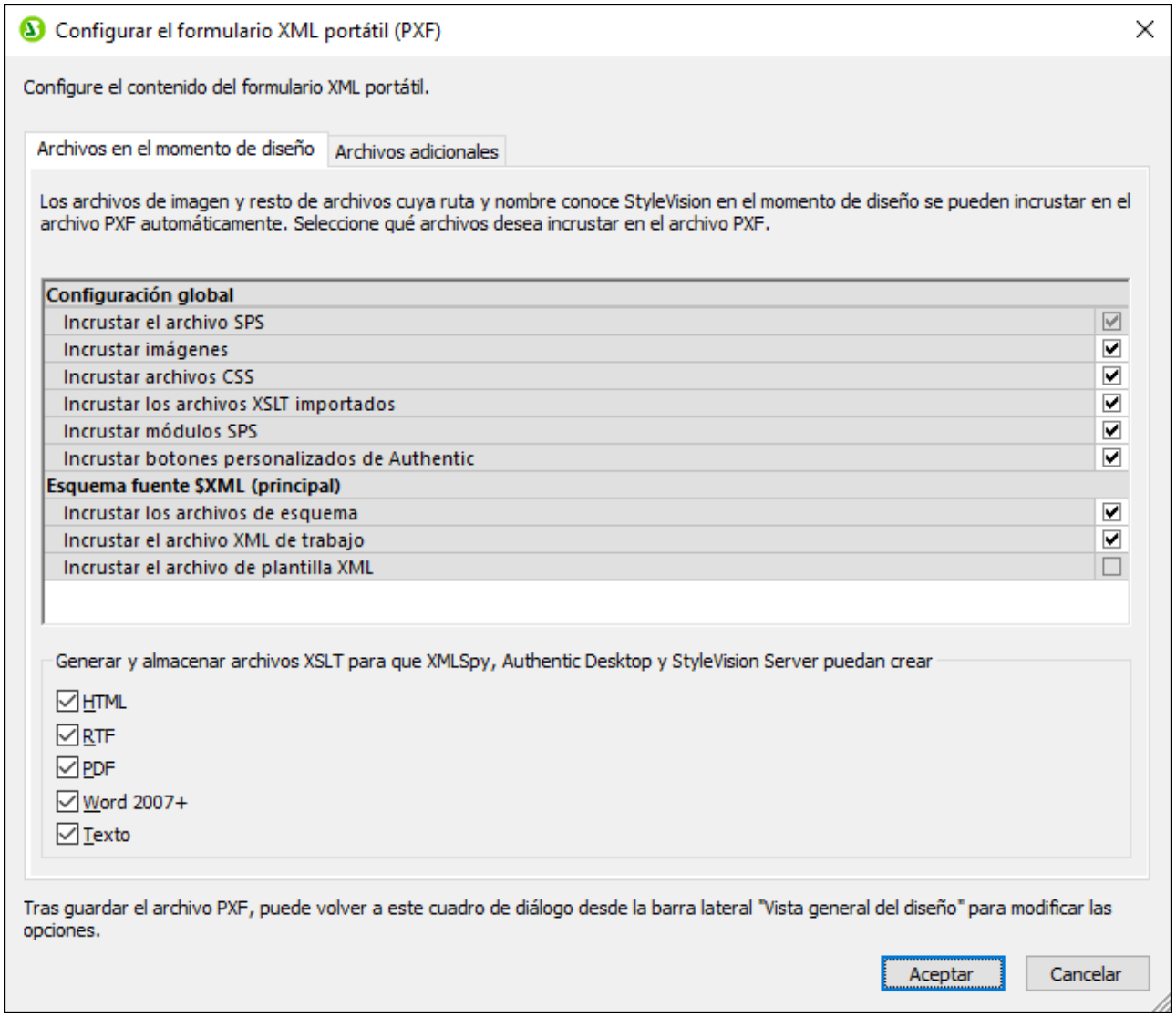
El formato estándar para los diseños de StyleVision es el **formato SPS** pero en este apartado nos ocupamos del formato PXF. Para ver cómo se guarda un archivo en formato SPS consulte la Referencia del usuario 778.

Guardar como PXF

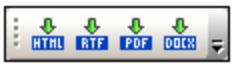
Si selecciona la opción *Guardar como archivo PXF* se abre el cuadro de diálogo "Guardar" de Windows, que

 © 2016-2022 Altova GmbH

funciona igual que con el comando **Guardar diseño** 773 solo que incluye un paso más: seleccionar los ficheros que se deben incluir en el archivo PXF. Tras indicar el nombre de archivo del PXF, aparece el cuadro de diálogo "Configurar el formulario XML portátil (PXF)" (*imagen siguiente*) donde puede seleccionar qué ficheros se incrustan en el PXF.



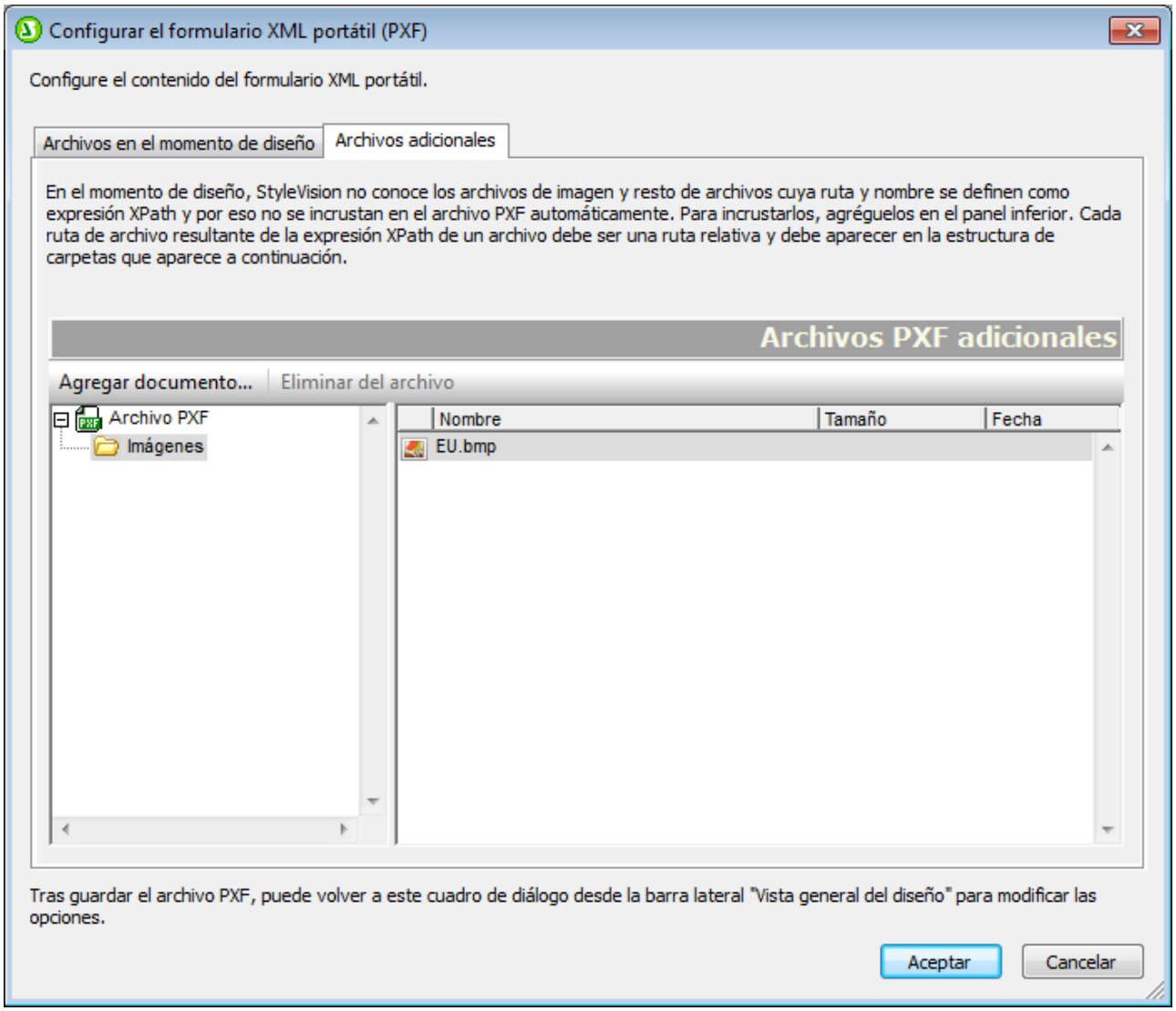
Bajo la entrada `Configuración global` de la pestaña *Archivos en el momento de diseño* puede seleccionar qué archivos relacionados con el diseño se incrustan y cuáles se omiten. También puede seleccionar los archivos XSLT generados a partir del diseño. En el panel inferior puede seleccionar para qué formatos desea generar e incrustar archivos XSLT. Si incrusta ficheros XSLT en el archivo PXF y abre el archivo PXF en la vista Authentic de un producto de Altova, se habilita el botón de la barra herramientas de Authentic que sirve para generar y ver documentos de salida (*imagen siguiente*).



Nota: si en el PXF incrusta archivos XSLT para formatos de salida que solo son compatibles con una edición superior de StyleVision (Enterprise es superior a Professional y Professional es superior a Basic) y el

PXF se abre en una edición inferior, cuando guarde el PXF los ficheros XSLT para formatos de salida incompatibles con la edición inferior no se guardarán. Aparece un aviso preguntando si quiere guardar el archivo PXF de todos modos.

En la pestaña *Archivos adicionales* (*imagen siguiente*) puede incluir más ficheros que no estén relacionados con el diseño pero que desee incrustar en el PXF de todos modos. Por ejemplo: imágenes a las que hace referencia el diseño a través de una URL generada con una expresión XPath. En el ejemplo siguiente el usuario seleccionó el archivo de imagen `EU.bmp` de la carpeta `Imágenes` para incrustarlo en el PXF.



Para incluir ficheros adicionales en el archivo PXF, haga clic en el botón **Agregar documento...** y busque el archivo. El cuadro de diálogo "Abrir" se abre automáticamente por la carpeta donde está el diseño SPS. Puede seleccionar ficheros de esta carpeta o de carpetas descendientes. Tras añadir el fichero al archivo XPF, el cuadro de diálogo muestra el fichero y la estructura de carpetas donde se encuentra. Por ejemplo, en la imagen anterior puede ver que el archivo `EU.bmp` está en una carpeta llamada `Imágenes`, que a su vez está en la carpeta del archivo SPS.

StyleVision devuelve un error si selecciona un archivo de una carpeta situada en un nivel superior a la carpeta

 © 2016-2022 Altova GmbH

donde está el diseño SPS.

En el diseño SPS las referencias a archivos adicionales deben hacerse con una ruta relativa y debe usar la estructura de carpetas que aparece en la pestaña *Archivos adicionales* (*imagen anterior*). Por ejemplo, para hacer referencia a la imagen `EU.bmp` de nuestro ejemplo debe usar la ruta relativa `Imágenes/EU.bmp`.

Nota: para poder guardar archivos PXF debe marcar la opción *Incrustar imágenes para RTF y Word 2007+* (**Archivo | Propiedades | Imágenes**).

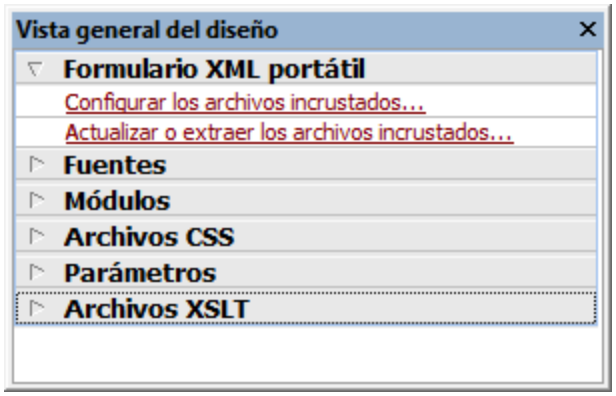
9.12.2 Editar un archivo PXF

Para abrir un archivo PXF en StyleVision haga clic en el comando **Archivo | Abrir** 768. Estos son los cambios que se pueden hacer en los archivos PXF:

- Puede editar la configuración del archivo PXF.
- Puede editar el contenido de algunos ficheros (como el SPS y el XML) en StyleVision y otros ficheros (como las imágenes y los archivos CSS) en aplicaciones externas. Sin embargo, todos los ficheros que componen el archivo PXF deben actualizarse de forma explícita en StyleVision.

Punto de entrada para editar el PXF

El punto de entrada para editar la configuración del PXF y actualizarlo es la entrada **Formario XML portátil** de la ventana Vista general del diseño (*imagen siguiente*).

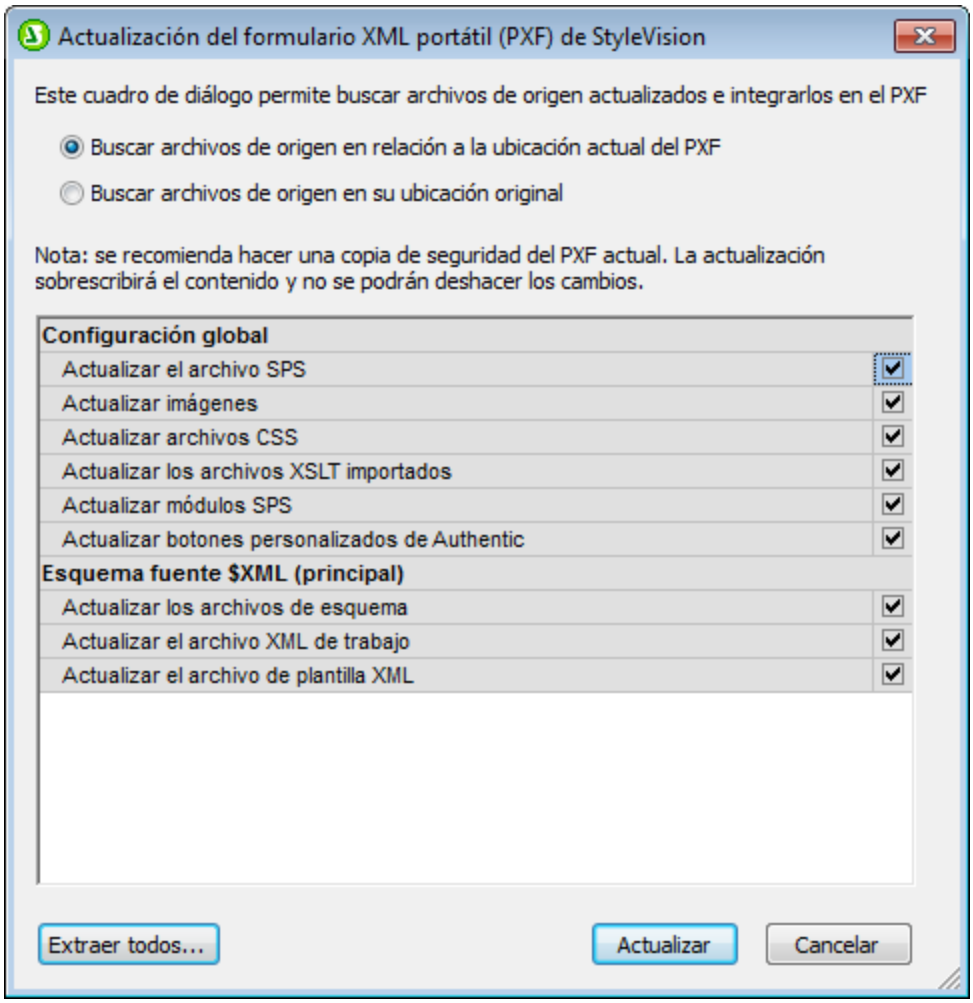


Configurar los archivos incrustados

Haga clic en el vínculo `Configurar los archivos incrustados...` de la ventana Vista general del diseño (*imagen anterior*) para abrir el cuadro de diálogo "Configurar el formulario XML portátil". Las opciones de configuración son las descritas en el apartado Crear un archivo PXF 512.

Actualizar los archivos incrustados

Haga clic en el vínculo `Actualizar o extraer los archivos incrustados...` de la ventana Vista general del diseño (*imagen anterior*) para abrir el cuadro de diálogo "Actualización del formulario XML portátil" (*imagen siguiente*).



Primero seleccione si los archivos de origen se deben buscar en relación a la ubicación actual del PXF o en su ubicación original. Después marque qué archivos desea actualizar y haga clic en **Actualizar**. Se crea un archivo PXF nuevo que sobrescribe el archivo PXF existente. Por tanto, recomendamos hacer una copia de seguridad del PXF antes de actualizarlo.

9.12.3 Implementar un archivo PXF

Una vez creado, el archivo PXF se puede transportar, descargar, copiar y guardar como cualquier otro archivo de datos. Como el archivo PXF puede contener todos los ficheros necesarios para editar un archivo XML en la vista Authentic y generar documentos de salida, el PXF es el único archivo que necesita el usuario de Authentic para realizar su trabajo.

Los archivos PXF se pueden abrir en la vista Authentic [27] de los productos de Altova. Aquí damos algunos ejemplos de cómo usar un archivo PXF con Altova XMLSpy, por ejemplo:

- El archivo PXF se abre con el comando **Archivo | Abrir**. El fichero XML incrustado se presenta en la vista Authentic gracias al SPS incrustado. El usuario edita el archivo XML en la vista Authentic y

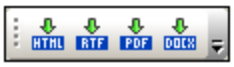
 © 2016-2022 Altova GmbH

guarda los cambios con **Archivo | Guardar**, que actualiza el archivo PXF (el XML incrustado se actualiza).

- El archivo PXF no contiene ningún XML incrustado y se abre con el comando **Archivo | Abrir**. En la vista Authentic se abre un archivo de plantilla XML basado en el diseño SPS. El comando **Archivo | Guardar** incrusta este archivo en el PXF.

- En XMLSpy puede asociar un archivo XML a un archivo PXF para que el SPS incrustado del archivo PXF se use para ver el XML en la vista Authentic del archivo. Esta asociación entre el archivo PXF y el archivo XML se crea con el comando **Authentic | Asignar una hoja de estilos de StyleVision**. Los cambios realizados se guardan en el archivo XML, no en el archivo PXF.

- Si el archivo PXF tiene incrustado un XSLT para uno de los formatos de salida, el usuario de la vista Authentic podrá generar documentos de salida en esos formatos. Esto se hace con el botón correspondiente de la barra de herramientas (*imagen siguiente*). Los botones de la barra de herramienta se habilitan en función del XSLT que tiene incrustado el archivo PXF. Por ejemplo, si el PXF tiene incrustadas hojas de estilos XSLT para los formatos HTML y PDF, solamente se habilitan esos botones de la barra de herramientas (mientras que los botones de RTF y DocX se deshabilitan).



Nota: si el archivo PXF está en un servidor web y se usa desde un complemento de Authentic para exploradores web, compruebe que el servidor no bloquea el archivo. Para ello añada este tipo MIME para las extensiones de archivo PXF (`.pxf`) : `application/x-zip-compressed` (desde el panel de administración de IIS, por ejemplo).

10 Archivos SPS y bases de datos

Los archivos SPS basados en bases de datos (aquellos cuyo esquema principal es una BD) se pueden usar con varios objetivos:

- Para editar la BD en la vista Authentic 27.
- Para generar un esquema XML con una estructura basada en la BD (si la BD no contiene un esquema. Las únicas BD que contienen esquemas son las BD XML como IBM DB2 versión 9).
- Para generar un archivo XML con datos de la BD (si los datos de la BD no están en formato XML todavía).
- Para diseñar y generar hojas de estilos XSLT para documentos de salida HTML, texto y RTF.
- Para generar informes de BD (basados en el diseño SPS) en formato HTML, texto y RTF. En StyleVision puede obtener una vista previa de estos informes.

Cuando la fuente de un esquema secundario de un SPS es una BD, los datos de la BD se pueden incluir en el documento de diseño pero la BD propiamente dicha no se puede editar en lavista Authentic 27. Lo que se puede editar es el documento XML o la BD asociados al esquema principal.

Procedimiento general

Esta sección describe el procedimiento general para trabajar con BD en StyleVision. Después del primer apartado 521, que resume el funcionamiento de las BD en StyleVision 521, los demás apartados describen todos los pasos del trabajo con BD. Tenga en cuenta que se distingue entre dos grandes tipos de BD: BD XML y BD no XML. El término *base de datos* tiene dos usos: un uso general, en referencia a todas las bases de datos y un uso específico, en referencia a las BD no XML. Para hacer referencia a bases de datos XML usamos siempre el término BD XML. Esto debe tenerse en cuenta porque el método de selección de los datos de BD para el SPS no es el mismo en estos dos tipos de BD.

- Conectarse a la BD 523: cómo conectarse a BD no XML, incluida la BD IBM DB2 (versiones anteriores a la versión 9).
- Seleccionar datos de la BD 609: cómo seleccionar datos XML y de esquema para el SPS a partir de la estructura de tabla de la BD (tanto BD XML como BD no XML).
- Esquema de BD y archivos XML de BD 620: cuando se cargan tablas de BD (de BD no XML), StyleVision genera y funciona con un esquema XML y un archivo XML basados en la estructura de la BD y en los datos de la BD respectivamente. En el caso de las BD XML, StyleVision no genera el esquema XML ni el archivo XML sino que hace referencia a los datos XML de la BD y al esquema disponible.
- Filtros de BD: filtrar datos de la BD 623: puede filtrar los datos de la BD que se cargan en el archivo XML temporal.
- Características de diseño para BD 628: los diseños SPS ofrecen funciones especiales para trabajar con BD, como controles de BD y consultas de BD.
- Generar archivos de salida 632: StyleVision puede generar informes de BD en gran cantidad de formatos de archivo.

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 StyleVision, compruebe que tiene acceso a los controladores de BD de 64 bits de la BD a la que quiere conectarse.

 © 2016-2022 Altova GmbH

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 SQLite desde la vista Authentic, un cuadro de mensaje le informará de esta limitación conocida.
Sybase ASE 15, 16	
Teradata 16	

Sitio web de Altova: Informes de BD

© 2016-2022 Altova GmbH

10.1 Bases de datos en StyleVision

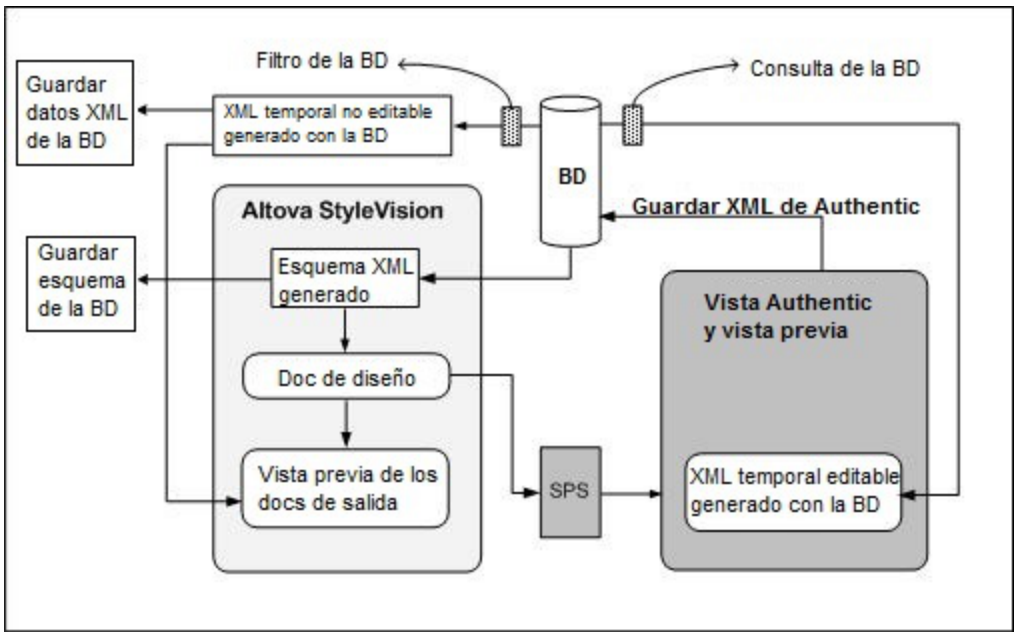
En StyleVision puede crear archivos SPS basados en una BD. Con estos archivos SPS puede:

- editar bases de datos en la vista Authentic y
- generar informes a partir de la base de datos.

Una vez creado el diseño SPS, puede ver los informes en StyleVision y generarlos en formatos de salida HTML y RTF. También puede guardar los archivos relacionados con la BD que StyleVision genera:

- un esquema XML basado en la estructura de la BD (no compatible con las BD XML, que ya tienen un esquema)
- un archivo XML con la estructura definida en el esquema generado y con el contenido de la BD (no compatible con las BD XML, donde los datos ya están disponibles en formato XML)
- una hoja de estilos XSLT para documentos de salida HTML (basada en el diseño SPS)
- una hoja de estilos XSLT para documentos de salida en formato texto (basada en el diseño SPS)
- una hoja de estilos XSLT para documentos de salida RTF (basada en el diseño SPS)
- el documento de salida HTML
- el documento de salida en formato texto
- el documento de salida RTF

El archivo XML guardado se puede procesar con la hoja de estilos XSLT pertinente. Esto permite mayor flexibilidad a la hora de generar informes.



Nota: StyleVision genera el esquema XML y los archivos XML a partir de bases de datos no XML y su estructura y contenido no se puede alterar en la vista Authentic. Esto se debe a que la estructura de estos archivos responde a la estructura de la BD no XML. La edición de la BD y la creación de informes a partir de ella depende de la estructura XML única que StyleVision genera a partir de la BD.

Funcionamiento general de los diseños SPS basados en BD

A continuación aparecen los pasos necesarios para crear y usar diseños SPS basados en una BD y para editar la BD y crear informes de BD en formato HTML, texto y RTF.

- Conectarse a la BD 523 con StyleVision. Durante el proceso de conexión puede especificar qué tablas de datos de la BD deben filtrarse en el esquema XML.
- Cuando se establece la conexión StyleVision genera un esquema XML temporal 620 basado en la estructura de la BD y el esquema aparece en la ventana Estructura del esquema de StyleVision. Si se trata de una BD XML, se hace referencia al esquema XML ya existente (bien en la BD o en otra ubicación).
- StyleVision también genera archivos XML temporales internos 620. Uno de ellos es un archivo XML no editable (*ver diagrama anterior*) y se usa para las vistas previas y como fuente del archivo de datos XML generado. El otro es un archivo XML editable que se puede ver en la vista Authentic (*ver diagrama anterior*). Cuando se guardan los cambios realizados en este archivo en la vista Authentic (con el comando **Archivo | Guardar datos XML de Authentic**), los cambios se escriben en la BD original. El archivo XML no editable se actualiza cada vez que se accede a una vista previa o cuando se guardan los datos XML.
- En StyleVision puede definir filtros de nivel superior 623 para limitar los datos importados en el archivo XML no editable (es decir, para las vistas previas y los documentos de salida).
- Puede usar consultas de la BD 628 dentro de la vista Authentic para limitar la lista de registros que se presentan en la vista Authentic. La consulta solo se usa durante la edición.
- Si realiza y guarda cambios en la BD, la próxima vez que abra una vista previa, el archivo XML no editable se actualiza con el nuevo contenido de la BD y la vista previa presenta un archivo actualizado.
- Los archivos SPS basados en BD se diseñan igual que los demás SPS: arrastrando y colocando nodos en la vista **Diseño**, insertando componentes estáticos, asignando propiedades y estilos, etc. Todos estos mecanismos se describen en la presente documentación.

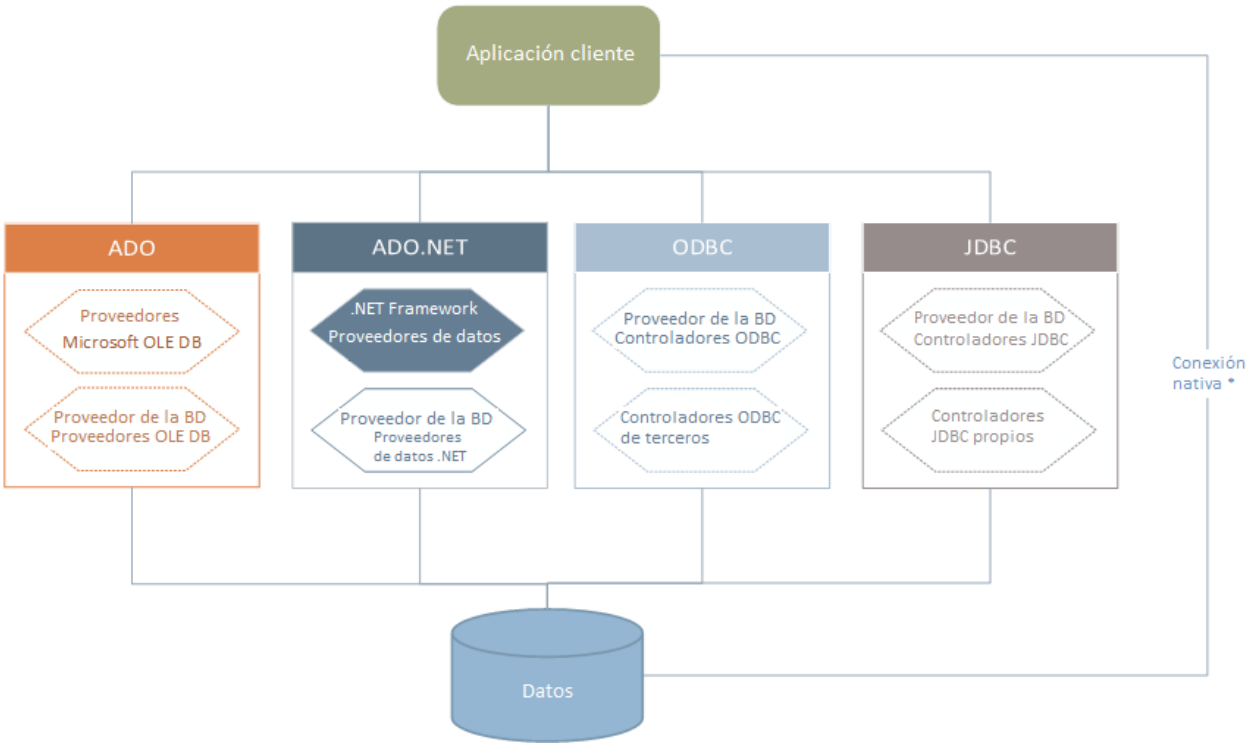
© 2016-2022 Altova GmbH

10.2 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 StyleVision 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, StyleVision 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 StyleVision (*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 StyleVision 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)

- 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* 541 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

10.2.1 Iniciar el asistente para la conexión de base de datos

StyleVision 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 526.

Para iniciar el asistente para la conexión de BD (*imagen siguiente*) siga estos pasos:

- En el menú haga clic en **Agregar** y después en **Nuevo desde BD**.

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 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**.

 © 2016-2022 Altova GmbH

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 553.

También puede consultar estos apartados:

- Conexiones ADO 529
- Conexiones ADO.NET 534
- Conexiones ODBC 541
- Conexiones JDBC 544

10.2.2 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* 541).
- 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* 543).
- 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* 541).
- 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* 544).
- 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/)
	ODBC	Controlador Firebird ODBC (https://www.firebirdsql.org/en/odbc-driver/)
IBM DB2	ADO	Proveedor IBM OLE DB para DB2

 © 2016-2022 Altova GmbH

Base de datos	Interfaz	Controladores
	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/)
	JDBC	Conector/JDBC (https://dev.mysql.com/downloads/connector/j/)
	ODBC	Conector/ODBC (https://dev.mysql.com/downloads/connector/odbc/)

Base de datos	Interfaz	Controladores
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)

© 2016-2022 Altova GmbH

10.2.3 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[534] 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:

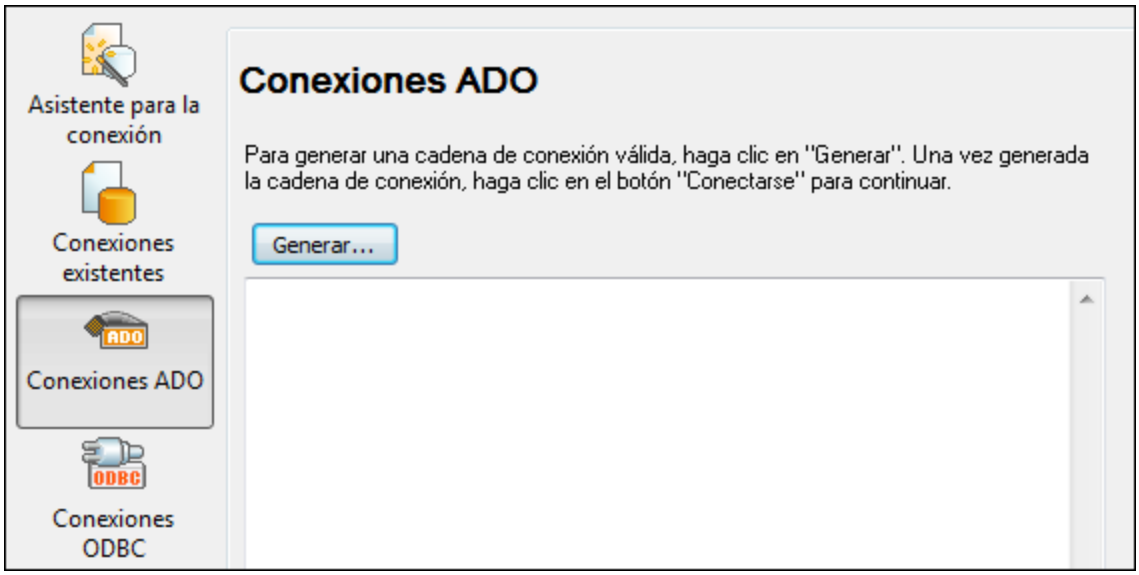
- *Controlador* de Microsoft OLE DB para SQL Server (MSOLEDBSQL)
- *Proveedor* de Microsoft OLE DB para SQL Server (SQLOLEDB)
- Cliente nativo de SQL Server (SQLNCLI)

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 StyleVision (32 bits o 64 bits). Los proveedores SQLOLEDB y SQLNCLI se consideran obsoletos, por lo que no se recomiendan.

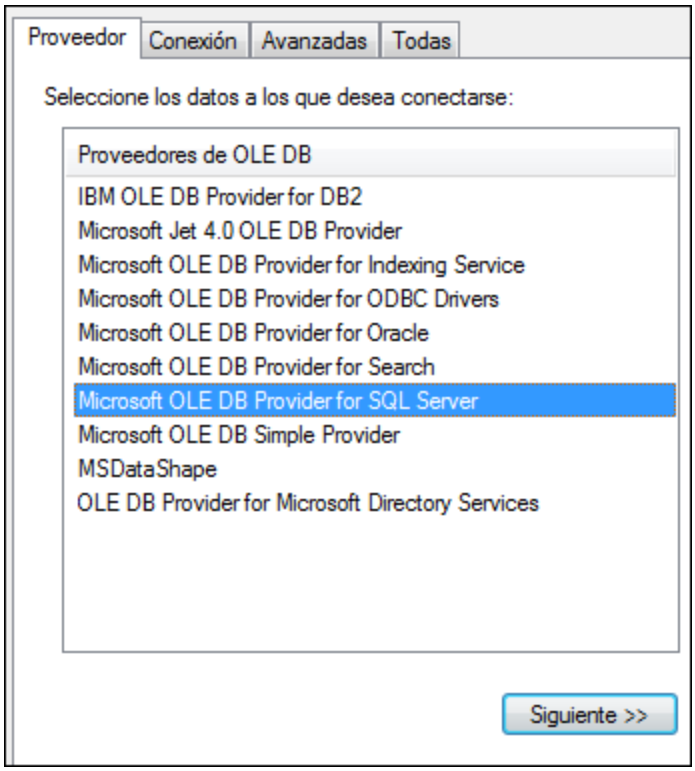
Es un problema conocido que el proveedor de BD para SQL Server **Microsoft OLE DB Provider for SQL Server (SQLOLEDB)** tiene dificultades para enlazar parámetros de consultas complejas como las expresiones comunes de tabla (CTE) e instrucciones SELECT anidadas.

Para configurar una conexión ADO:

1. Inicie el asistente para la conexión de base de datos[524].
2. Haga clic en **Conexiones ADO**.



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.

 © 2016-2022 Altova GmbH

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 526). Otra opción es configurar una conexión ADO.NET, ODBC o JDBC. 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 541.

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) 576. 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) 574.

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 532
- Configurar las propiedades de vínculo de datos de Microsoft Access 533

10.2.3.1 Conectarse a una base de datos Microsoft Access

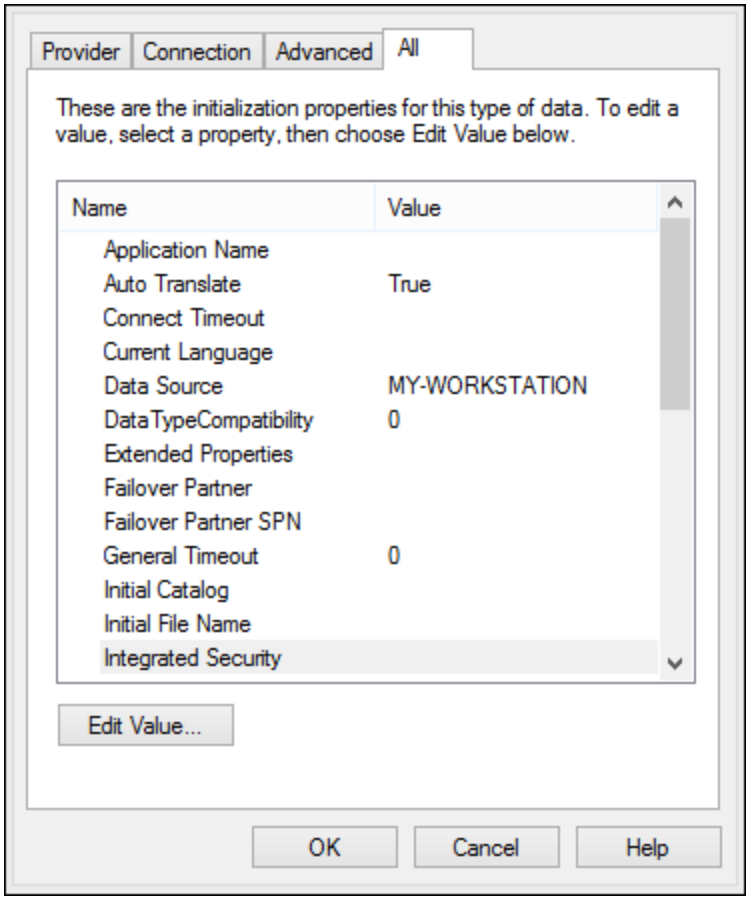
Siga estas instrucciones si desea conectarse a una base de datos Microsoft Access que no requiere contraseña. Si la base de datos está protegida con contraseña, entonces consulte el apartado Conectarse a Microsoft Access (ADO) 574.

Para conectarse a una BD Microsoft Access:

1. Inicie el asistente para la conexión de base de datos 524.
2. Seleccione el botón de opción *Microsoft Access (ADO)* y haga clic en **Siguiente**.
3. Busque el archivo de BD o introduzca su ruta de acceso (relativa o absoluta).
4. Haga clic en **Conectarse**.

10.2.3.2 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 529), 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

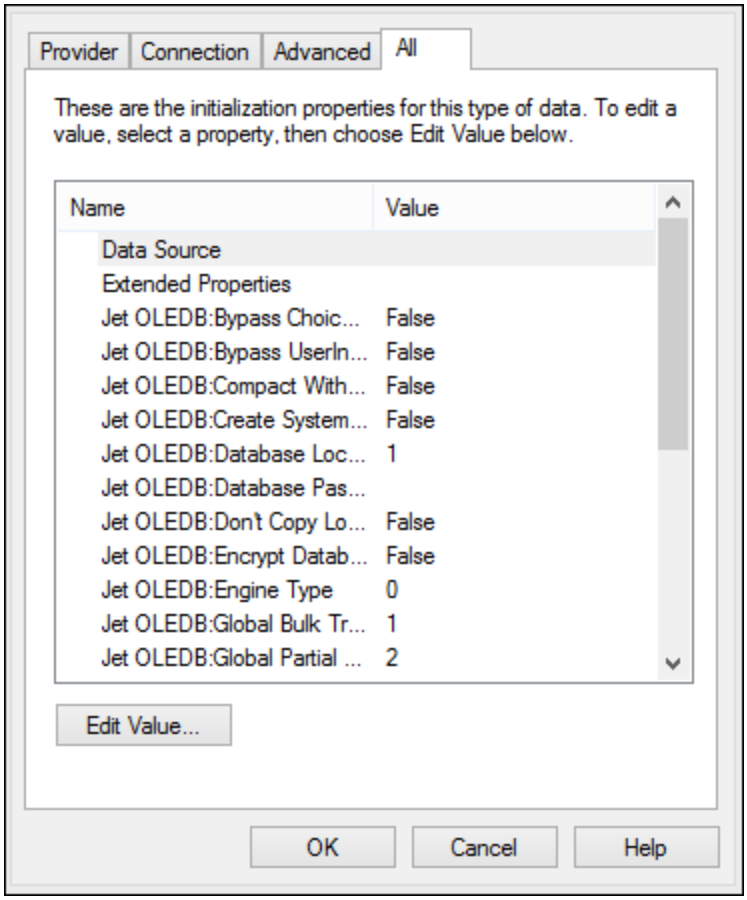
Propiedad	Notas
Seguridad integrada	Si seleccionó el proveedor de datos **SQL Server Native Client** en la pestaña *Proveedor*, esta propiedad debe ser un carácter de espacio en

 © 2016-2022 Altova GmbH

Propiedad	Notas
	blanco.
Almacenar información de seguridad	Esta propiedad debe tener el valor **True**.

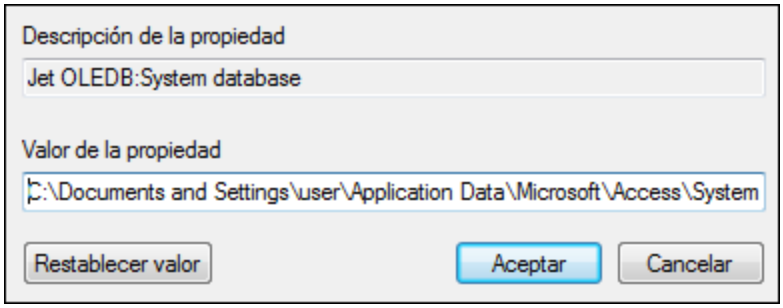
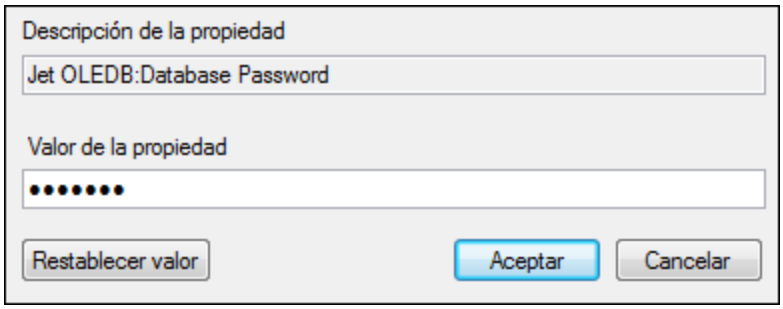
10.2.3.3 Configurar las propiedades de vínculo de datos de Microsoft Access

Cuando se conecte a una base de datos Microsoft Access por ADO (consulte Conexiones ADO 529), puede que deba configurar 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
Origen de datos	Esta propiedad almacena la ruta de acceso del archivo de BD Microsoft Access. Para evitar problemas de conexión, recomendamos

Propiedad	Notas
	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. 
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. 

10.2.4 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 StyleVision 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.

 © 2016-2022 Altova GmbH

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 StyleVision 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 541.

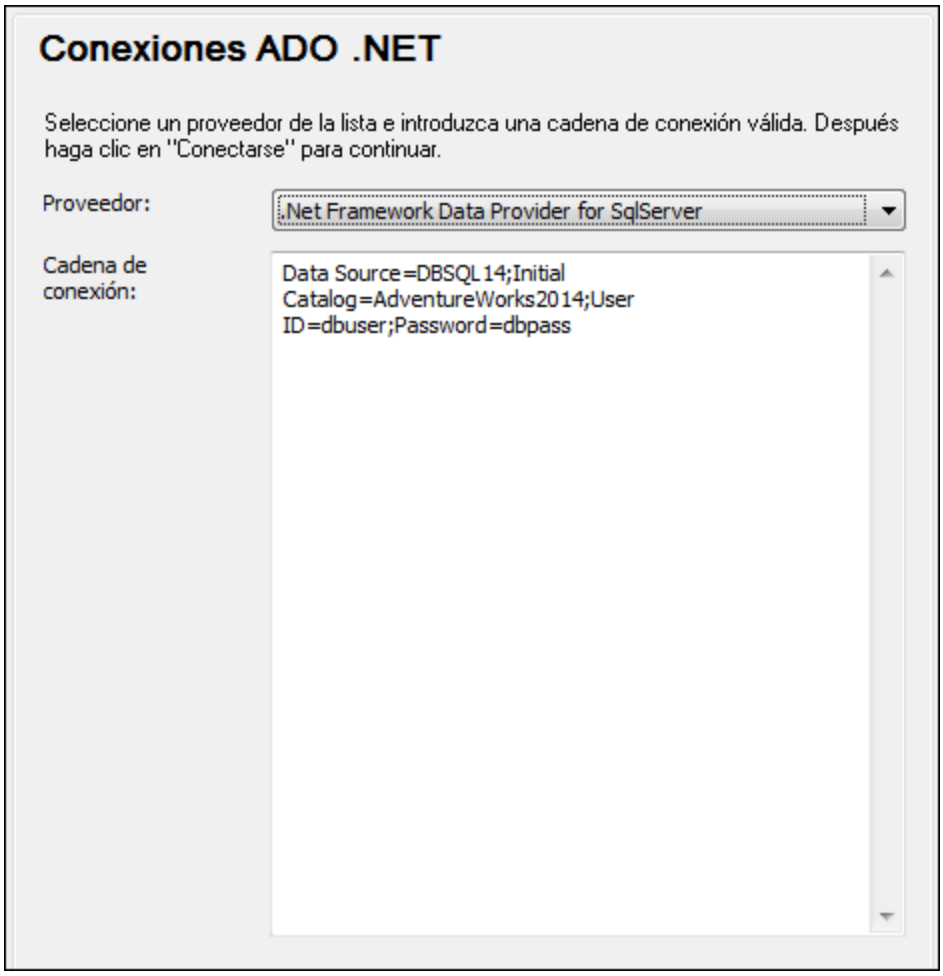
Para configurar una conexión ADO.NET:

1. Inicie el asistente para la conexión de base de datos 524.
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* 536).

> 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 539.



5. Haga clic en **Conectarse** para terminar.

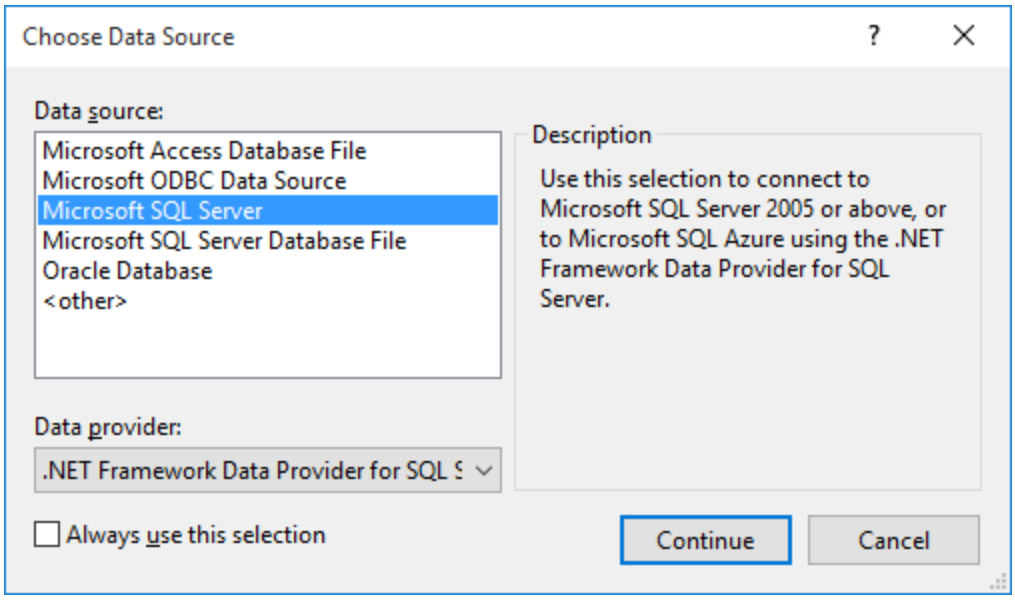
10.2.4.1 Crear una cadena de conexión en Visual Studio

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.

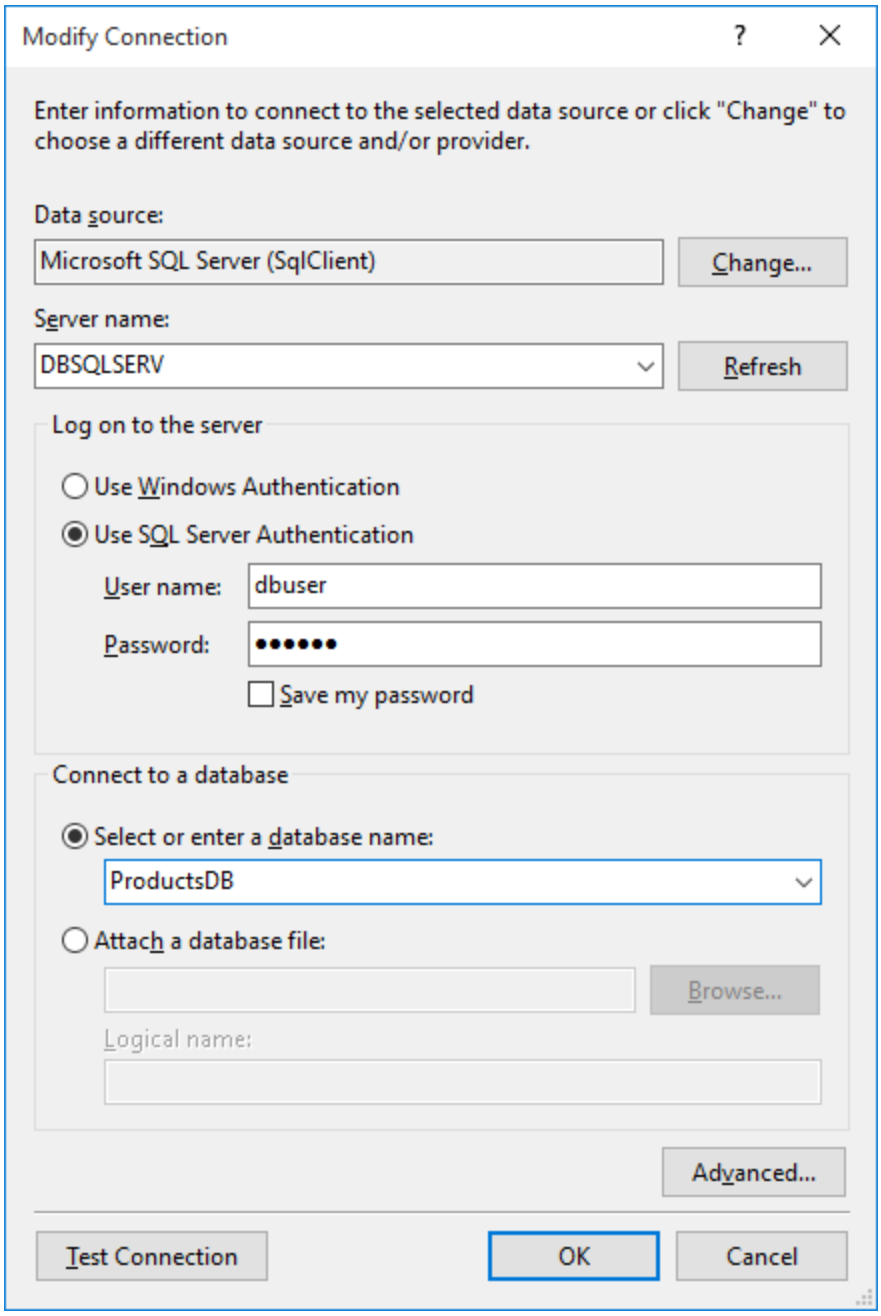
Para crear una cadena de conexión en Visual Studio:

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 StyleVision.

10.2.4.2 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* 534). 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) 589).

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

10.2.4.3 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 StyleVision 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.

10.2.5 Conexiones ODBC

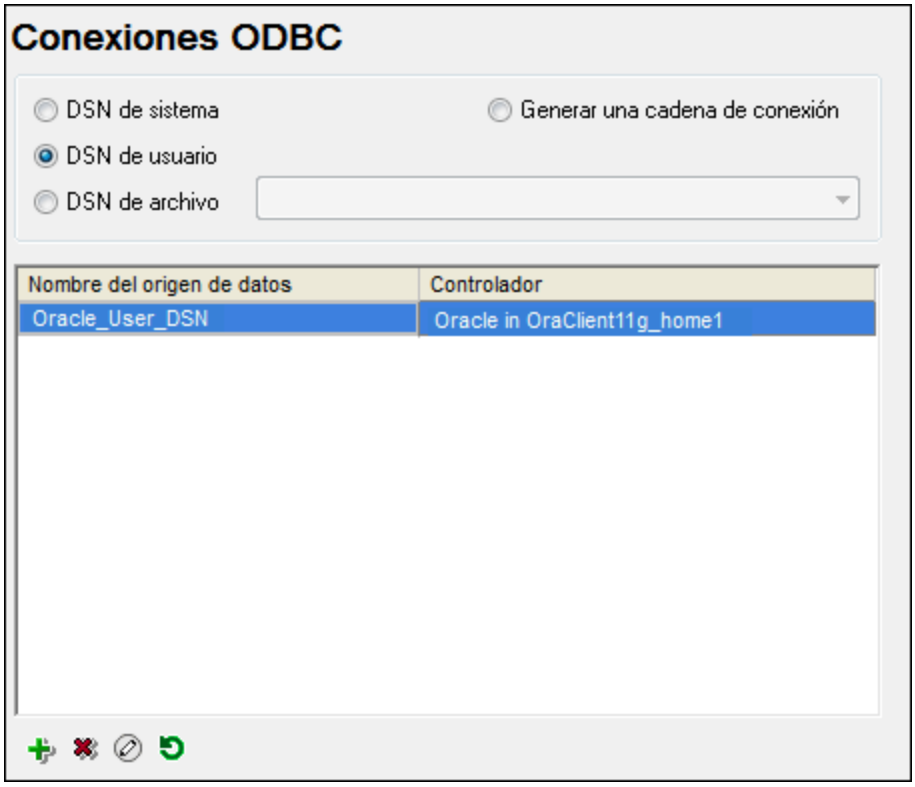
ODBC (Open Database Connectivity) es una tecnología de acceso de datos muy popular con la que se puede conectar a bases de datos desde StyleVision. Puede utilizarse como método principal de conexión con bases de datos o como alternativa a las conexiones OLE DB o JDBC.

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 StyleVision. 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 StyleVision 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 543).

Para conectarse usando un DSN nuevo:

1. Inicie el asistente para la conexión de base de datos 524.
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 524.
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 524.
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:

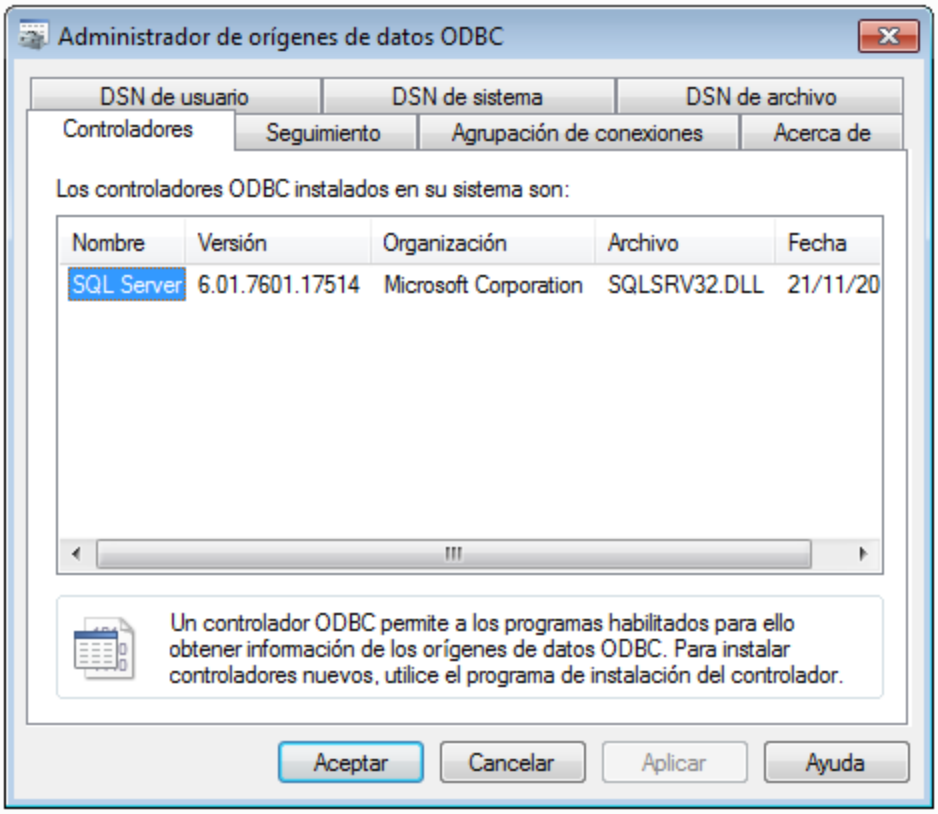
1. Inicie el asistente para la conexión de base de datos 524.
2. Haga clic en el botón **Conexiones ODBC**.
3. Seleccione *Generar una cadena de conexión*.
4. Pegue la cadena de conexión en el cuadro de texto y haga clic en **Conectarse**.

10.2.5.1 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 526). Cuando el controlador esté disponible en el sistema podrá crear conexiones ODBC con él.

10.2.6 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. StyleVision identifica la ruta de

 © 2016-2022 Altova GmbH

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 StyleVision (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* 553).
- 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 547.

Conectarse a un servidor SQL a través de JDBC con credenciales de Windows

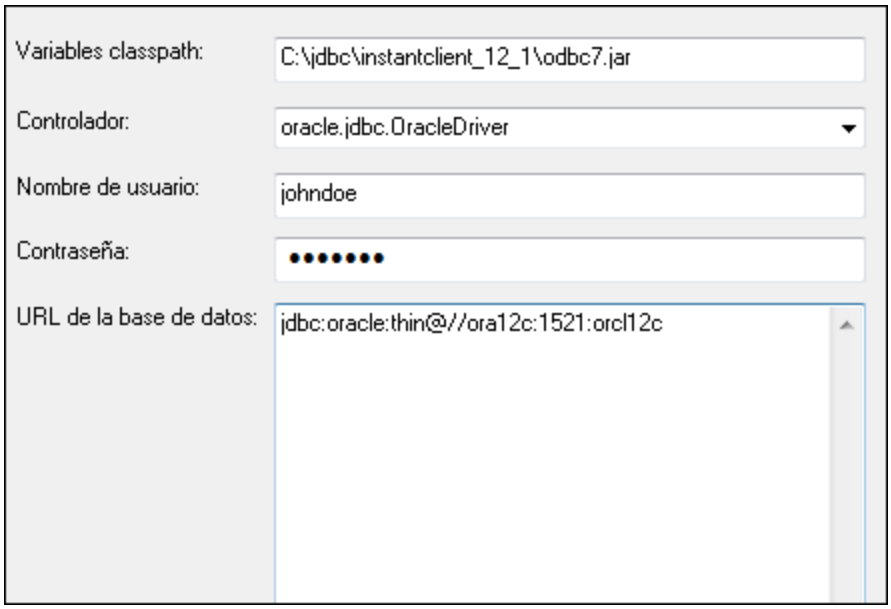
Si se conecta a un servidor SQL mediante JDBC con credenciales de Windows (seguridad integrada), tenga en cuenta lo siguiente:

- 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 524.
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* 547), 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**.

10.2.6.1 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.

10.2.7 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.Además, si tiene pensado implementar archivos para ejecutarlos en servidores macOS o Linux, tampoco será necesario instalar controladores en el servidor de destino.

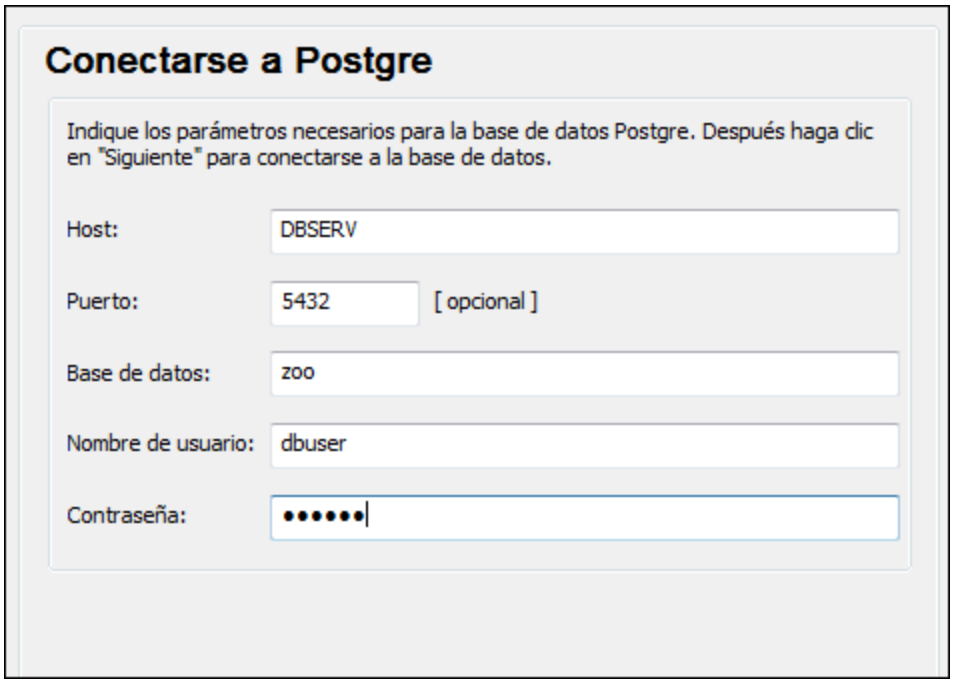
Si prefiere establecer la conexión por medio de un controlador no nativo, consulte estos temas de la documentación:

- Configurar una conexión JDBC 544
- Conectarse a PostgreSQL (ODBC) 595

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 524.
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**).

10.2.8 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 StyleVision, 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.

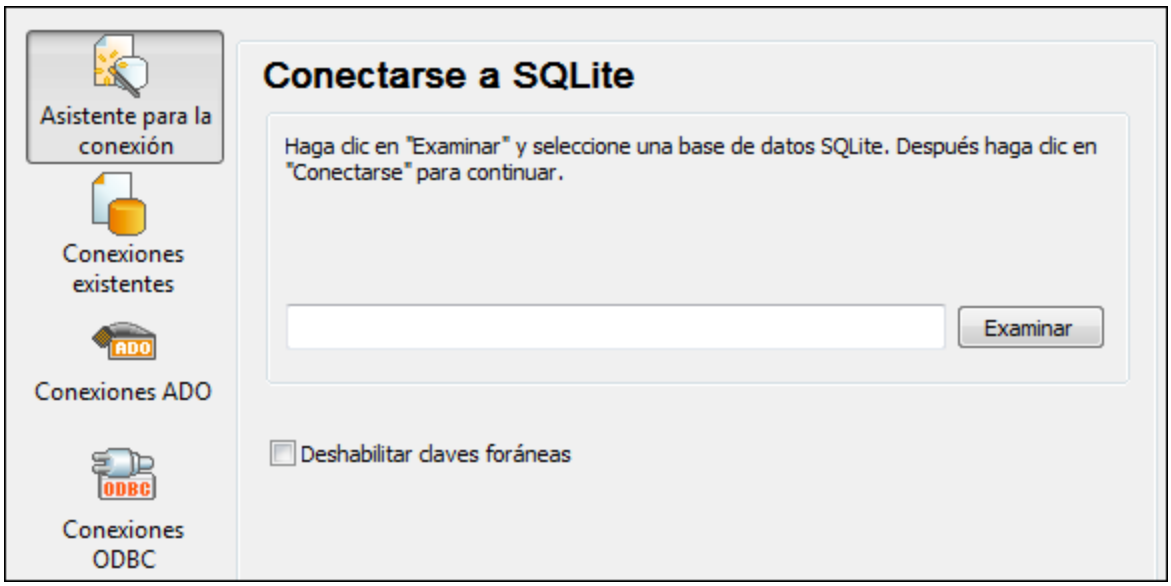
Importante

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.

10.2.8.1 Conectarse a una base de datos SQLite

Para conectarse a una base de datos SQLite:

1. Inicie el asistente para la conexión de base de datos 524
2. Seleccione el botón de opción *SQLite* y después haga clic en **Siguiente**.
3. 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.



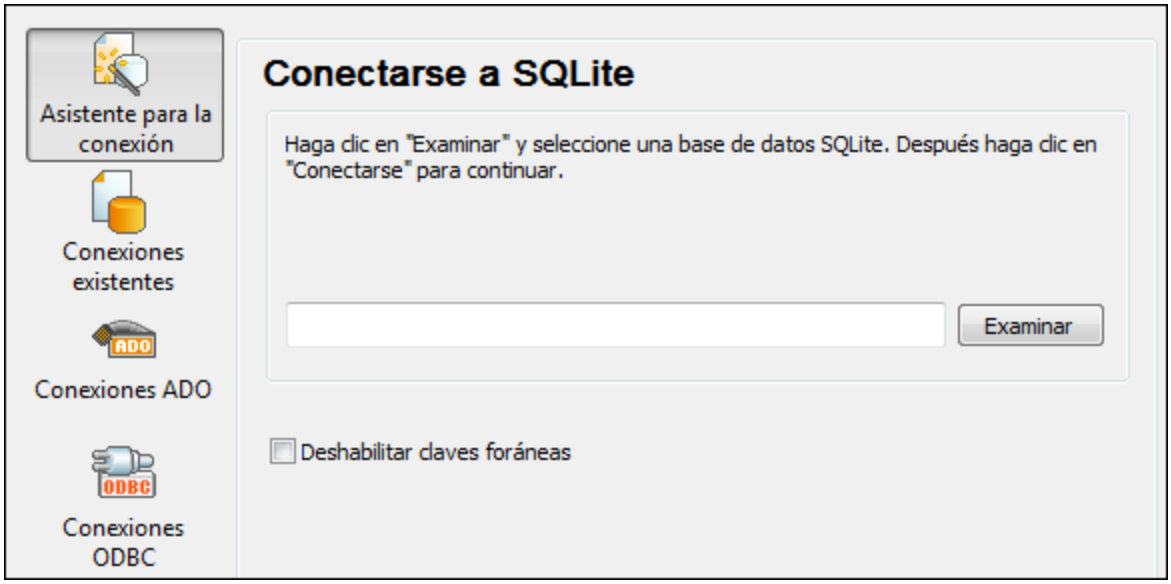
4. Si quiere, ahora puede marcar la casilla *Deshabilitar claves foráneas* (*véase Restricciones de clave foránea* 551).
5. Ahora haga clic en **Conectarse**.

10.2.8.2 Restricciones de clave foránea

Cuando se conecte con una base de datos SQLite desde StyleVision 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.)

10.2.9 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:

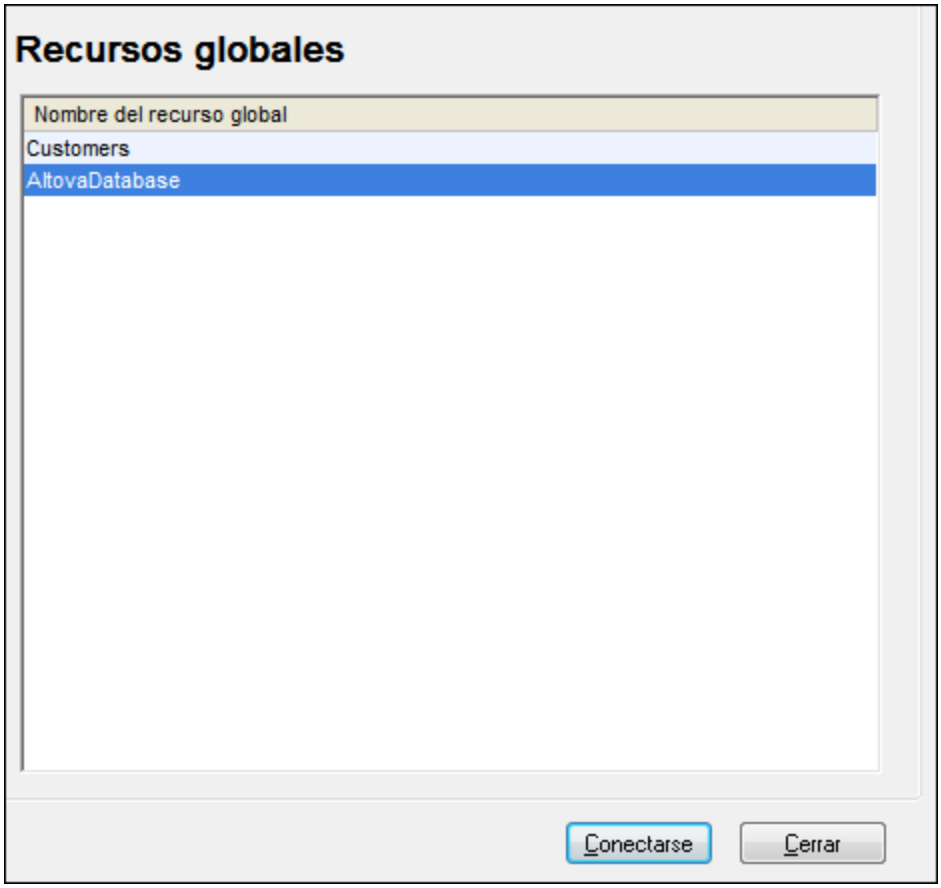
1. En el menú **Herramientas** de StyleVision, haga clic en **Recursos globales**.
2. Haga clic en **Agregar** y después en Base de datos.
3. Escriba un nombre para el recurso global en el campo *Alias del recurso*.
4. Haga clic en **Elegir base de datos**. Aparece el asistente para la conexión de base de datos 524.
5. Use el asistente para agregar una conexión de base de datos como hemos explicado.

 © 2016-2022 Altova GmbH

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*).



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.

10.2.10 Ejemplos de conexión a bases de datos

Esta sección incluye ejemplos de cómo conectarse a una base de datos desde StyleVision con ADO, ODBC o JDBC. Para ver los ejemplos de ADO.NET consulte el apartado Ejemplo: cadenas de conexión ADO.NET 539. Para ver cómo establecer una conexión nativa a PostgreSQL y SQLite (véanse respectivamente Conexiones PostgreSQL 549 y Conexiones SQLite 550).

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.

10.2.10.1 Firebird (JDBC)

Este ejemplo explica cómo conectarse a una base de datos Firebird por JDBC.

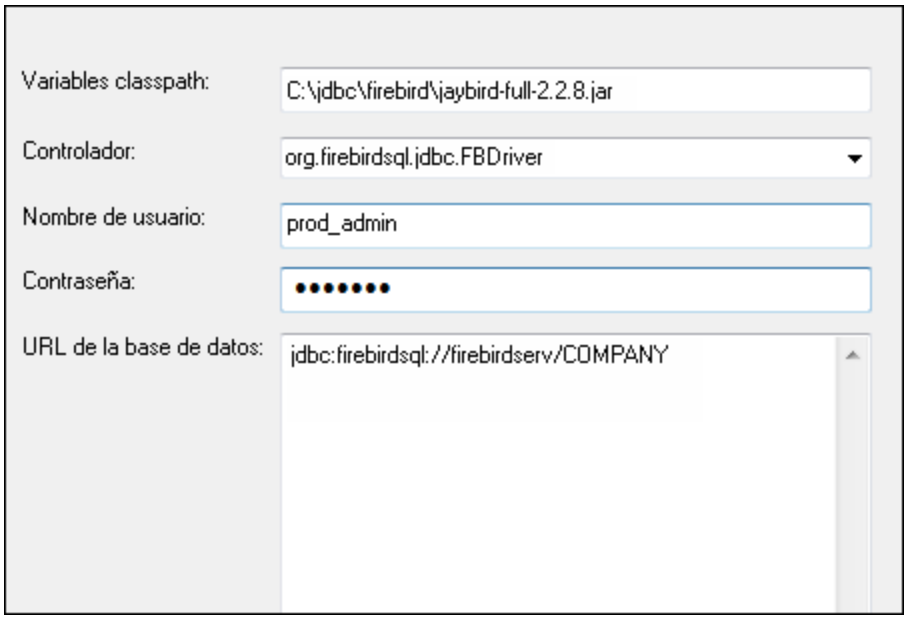
Requisitos:

- 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. StyleVision 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 StyleVision (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 524.
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* 547).
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.

© 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:firebirdsql://<host>[:<puerto>]/<ruta o alias de la BD>
```

7. Haga clic en **Conectarse**.

10.2.10.2 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.

Nota:

- La plataforma del controlador ODBC de Firebird y del cliente (de 32 o 64 bits) debe coincidir con la plataforma de StyleVision.
- La versión del cliente Firebird debe coincidir con la versión del servidor Firebird al que desea conectarse.

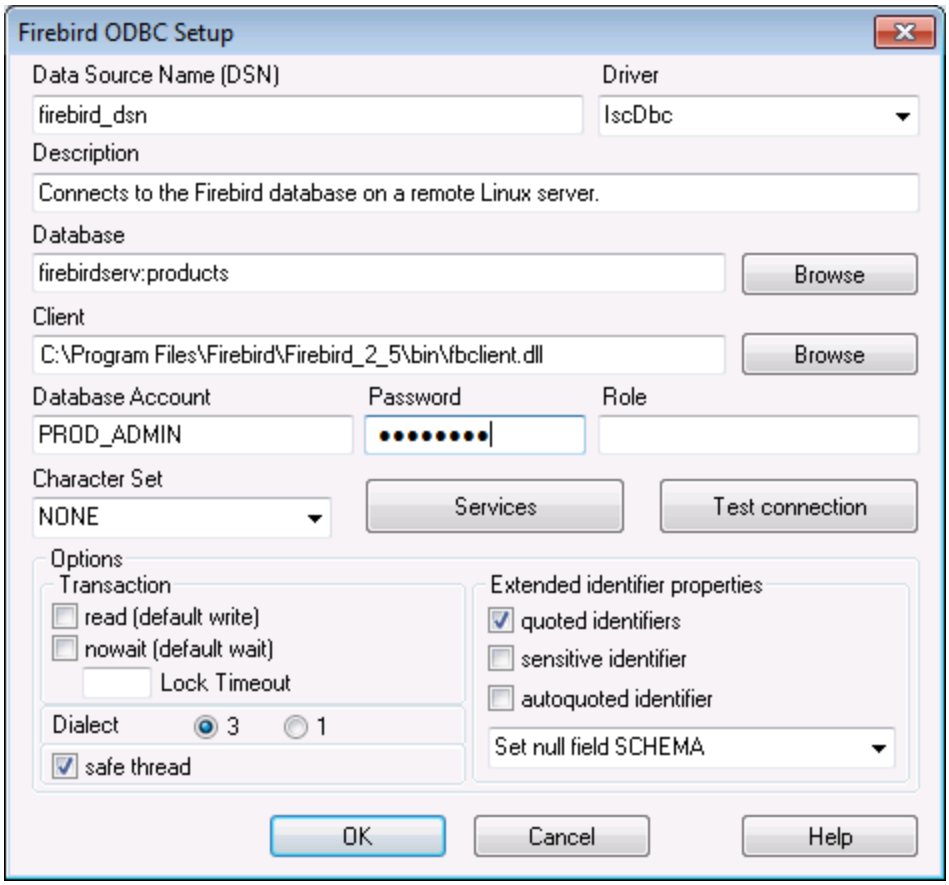
- Disponer de los datos de conexión: nombre de host o dirección IP, ruta de acceso (o alias) de la base de datos en el servidor, nombre de usuario y contraseña.

Para conectarse a Firebird por ODBC:

1. Inicie el asistente para la conexión de base de datos 524.
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.

© 2016-2022 Altova GmbH



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`

	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**.

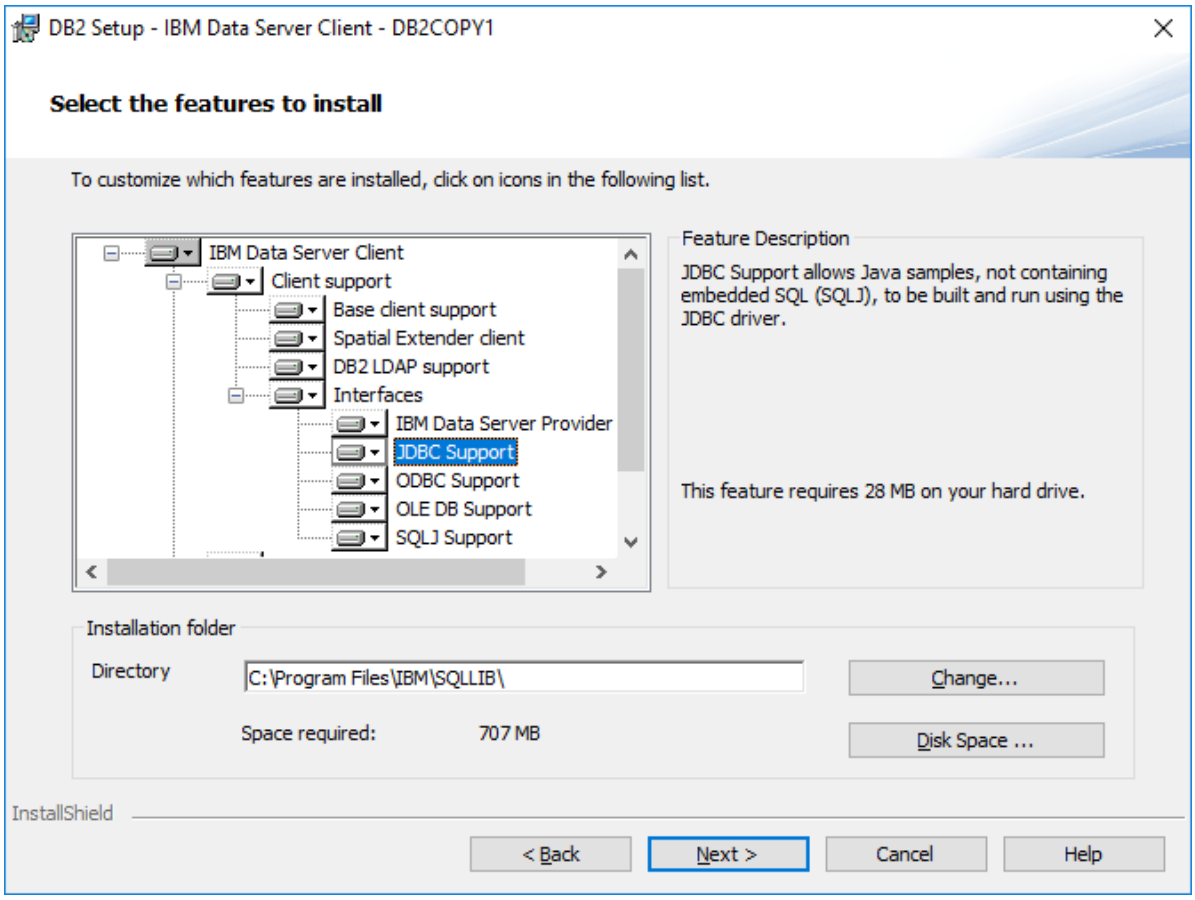
10.2.10.3 IBM DB2 (JDBC)

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. StyleVision 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 StyleVision (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 StyleVision.
- 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.

© 2016-2022 Altova GmbH

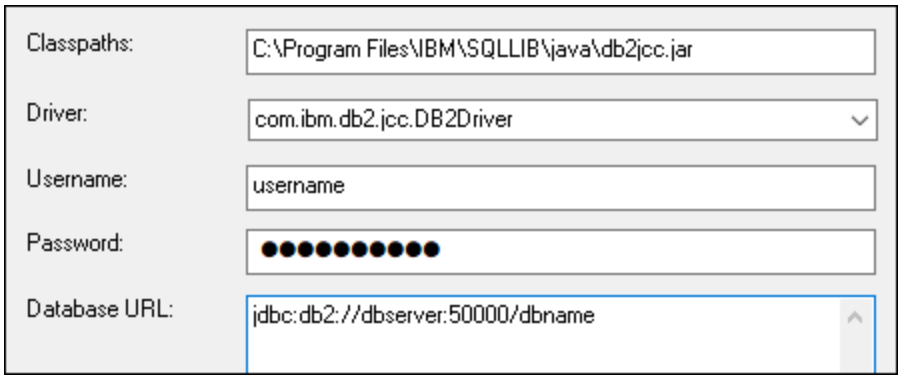


Si no ha modificado la ruta predeterminada de la instalación, una vez esta haya finalizado los archivos .jar requeridos están en el directorio **C:\Program Files\IBM\SQLLIB\java**.

- 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 524.
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* 547).
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*).



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**.

10.2.10.4 IBM DB2 (ODBC)

Este ejemplo explica cómo conectarse a una base de datos IBM DB2 por ODBC.

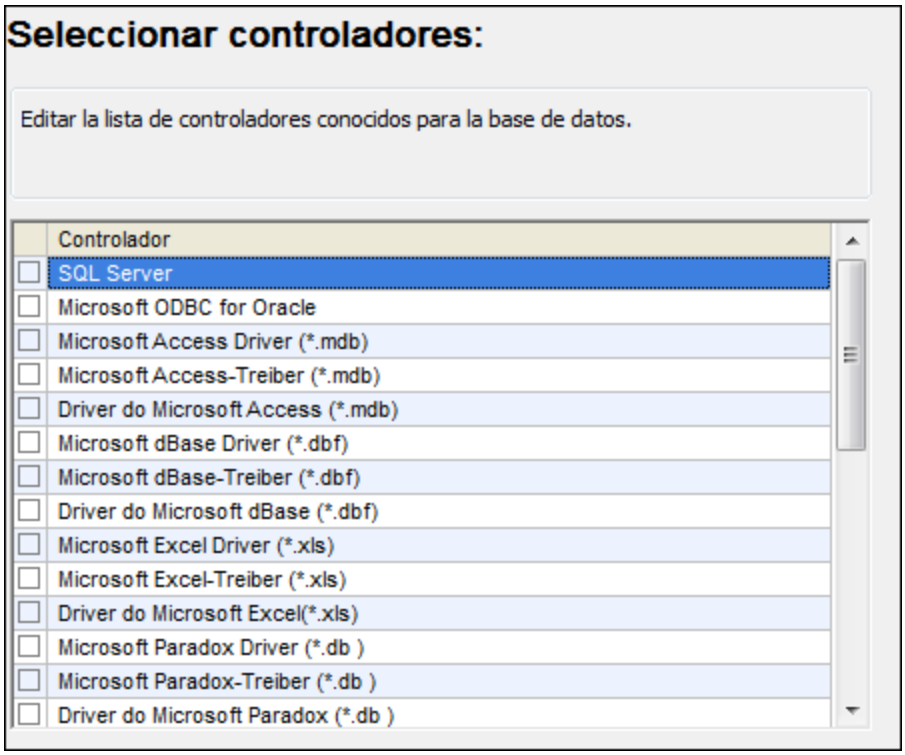
Requisitos:

- 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 543).
- 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 524 y seleccione la opción *IBM DB2 (ODBC/JDBC)*.
2. Haga clic en **Siguiente**.

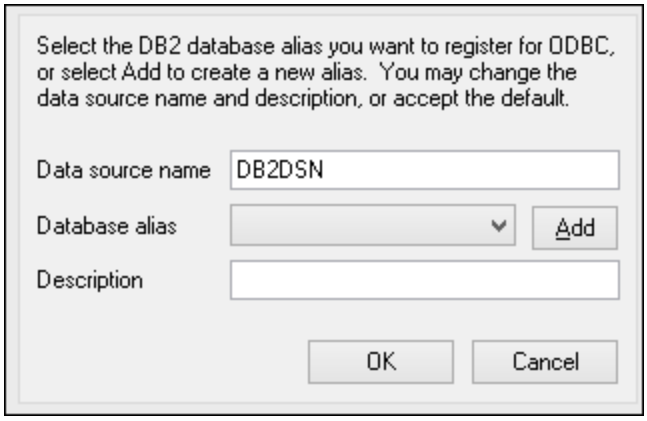
3. Seleccione la opción *ODBC* y haga clic en **Siguiente**. Si necesita editar la lista de controladores conocidos para la base de datos, seleccione los controladores que corresponden a IBM DB2 (ver Requisitos 560) y haga clic en **Siguiente**.



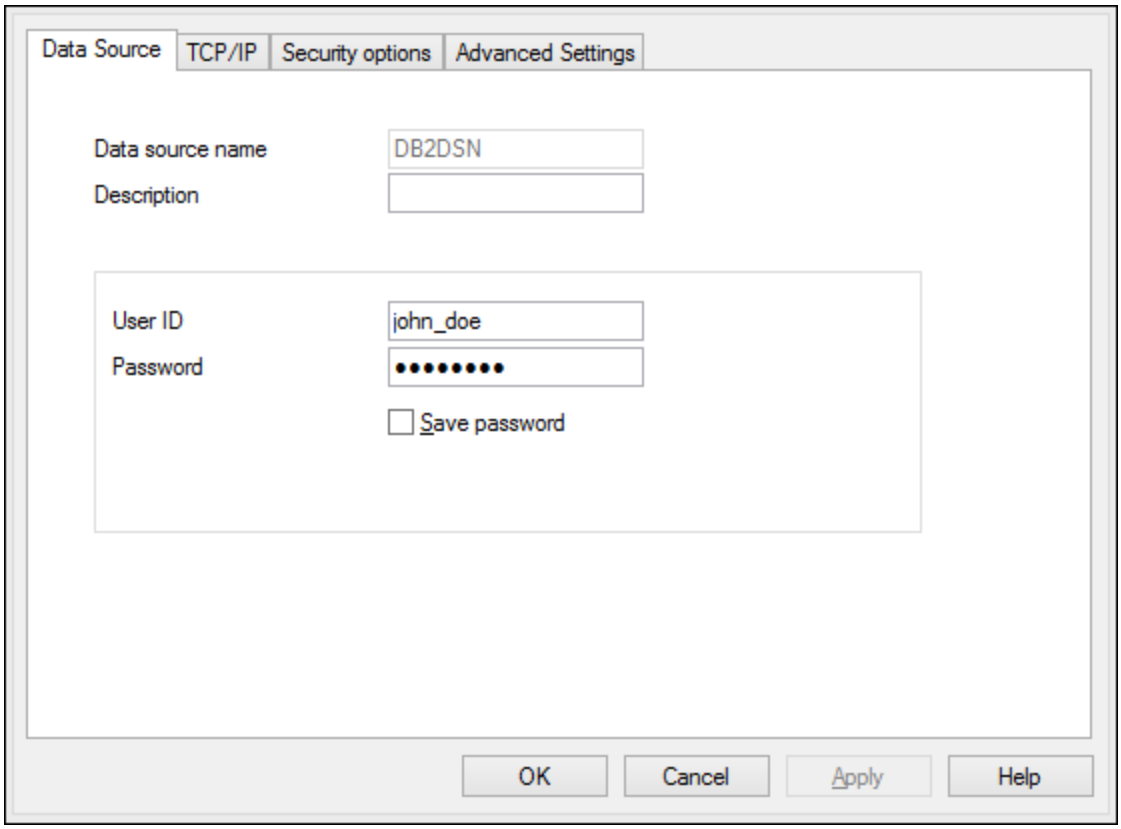
4. Seleccione el controlador de la lista y haga clic en **Conectarse**. (Para editar la lista de controladores disponibles haga clic en **Editar controladores** y active/desactive los controladores que desea agregar o eliminar)

© 2016-2022 Altova GmbH

5. Introduzca el DSN (**DB2DSN**) y haga clic en **Agregar**.

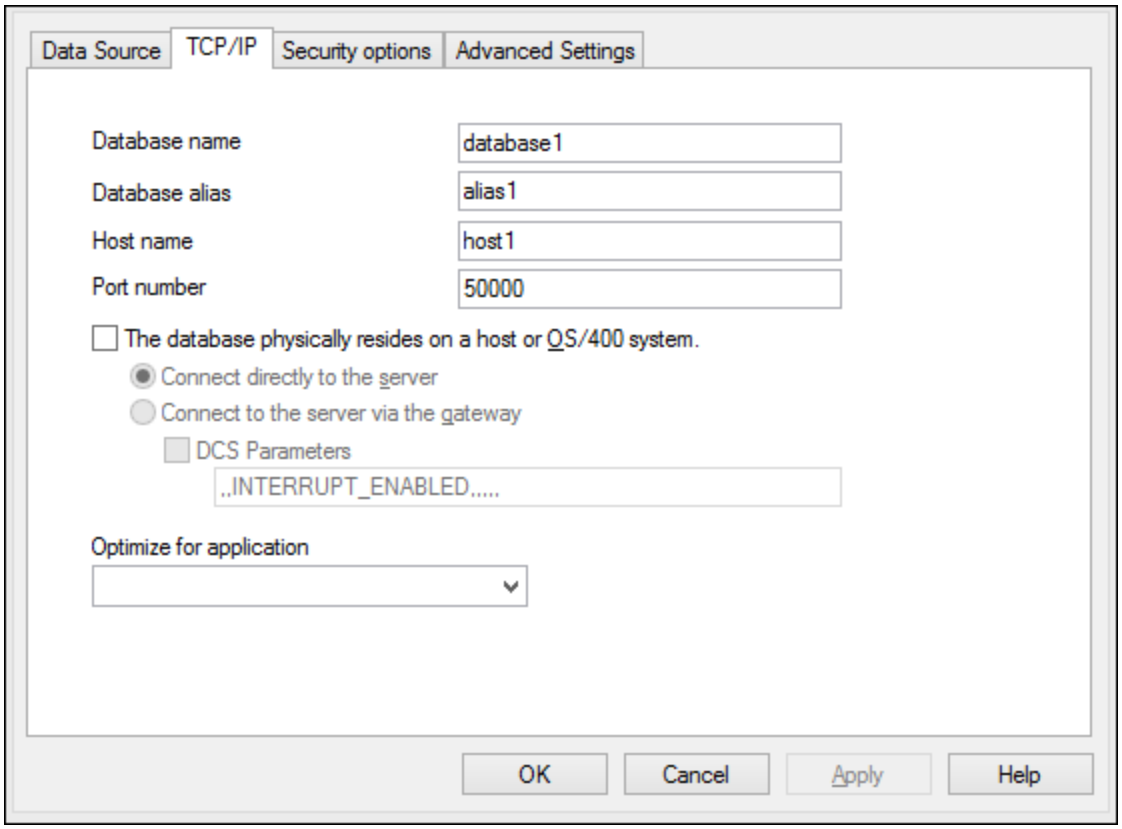


6. En la pestaña *Origen de datos* introduzca el nombre de usuario y la contraseña de la base de datos.

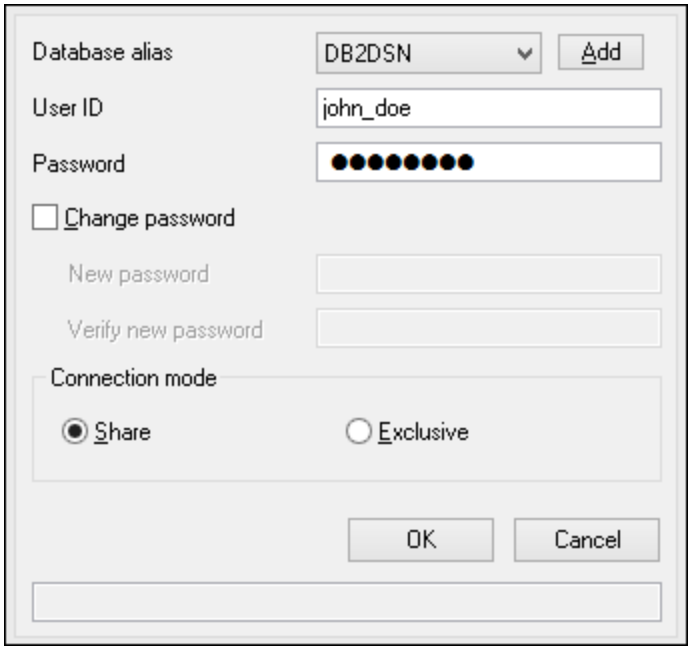


7. En la pestaña **TCP/IP** introduzca el nombre de la base de datos, un nombre para el alias, el nombre de host y el número de puerto. Después haga clic en **Aceptar**.

© 2016-2022 Altova GmbH



8. Vuelva a introducir el nombre de usuario y la contraseña y haga clic en **Aceptar**.



10.2.10.5 IBM DB2 para i (JDBC)

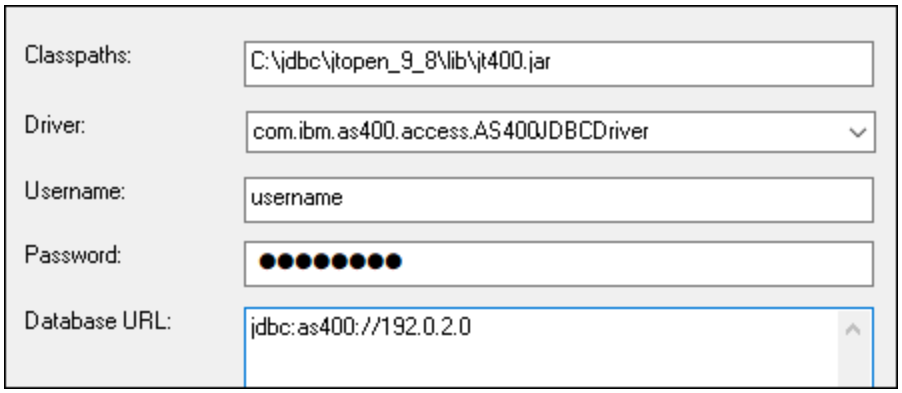
Este ejemplo explica cómo conectarse a una base de datos IBM DB2 for i por JDBC.

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. StyleVision 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 StyleVision (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 StyleVision.
- 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 524.
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* 547).
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.

 © 2016-2022 Altova GmbH

```
jdbc:as400://host
```

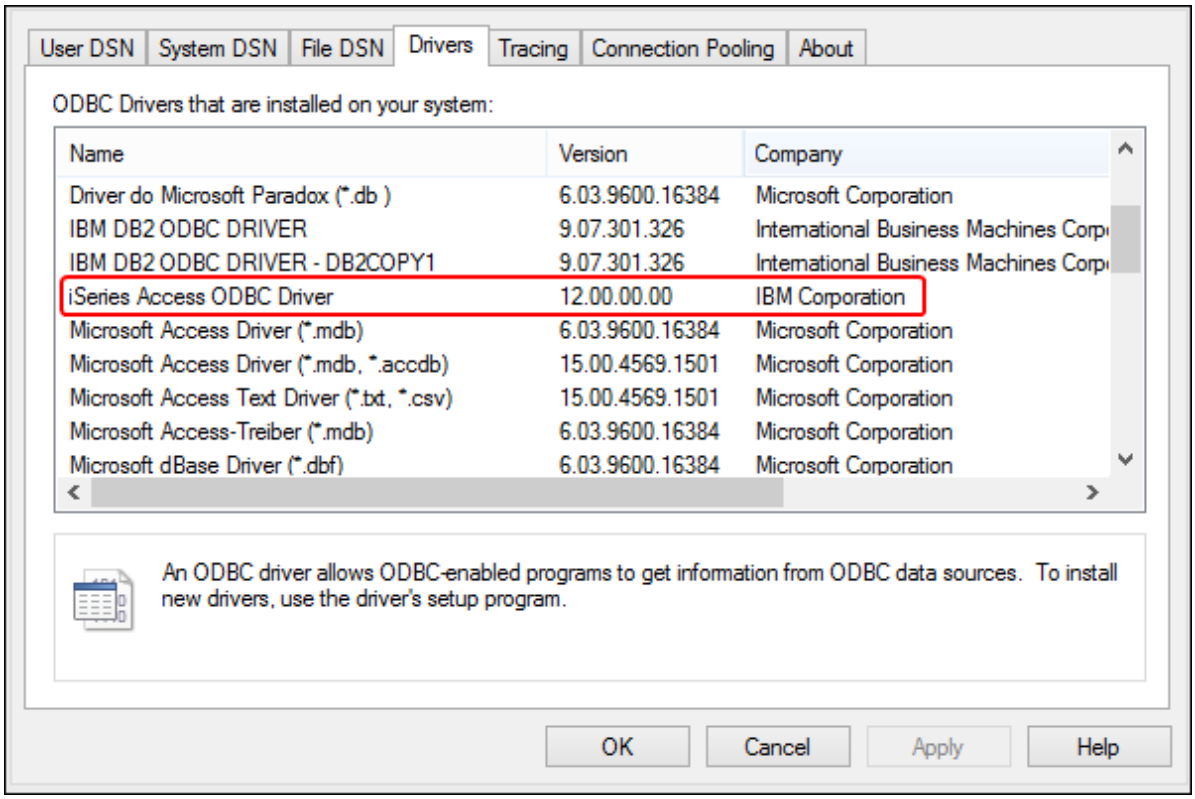
7. Haga clic en **Conectarse**.

10.2.10.6 IBM DB2 para i (ODBC)

Este ejemplo explica cómo conectarse a una base de datos IBM DB2 for i por ODBC.

Requisitos:

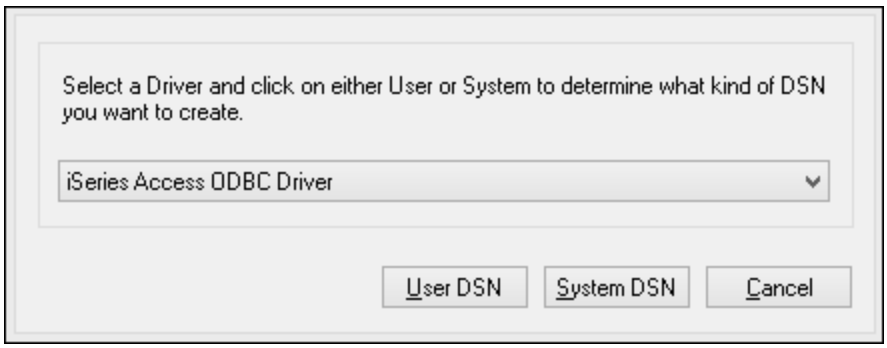
- *IBM System i Access for Windows* está instalado en el sistema (para este ejemplo usamos *IBM System i Access for Windows V6R1M0*). Compruebe que el controlador ODBC está en el equipo (consulte Ver los controladores ODBC disponibles 543).



- 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.

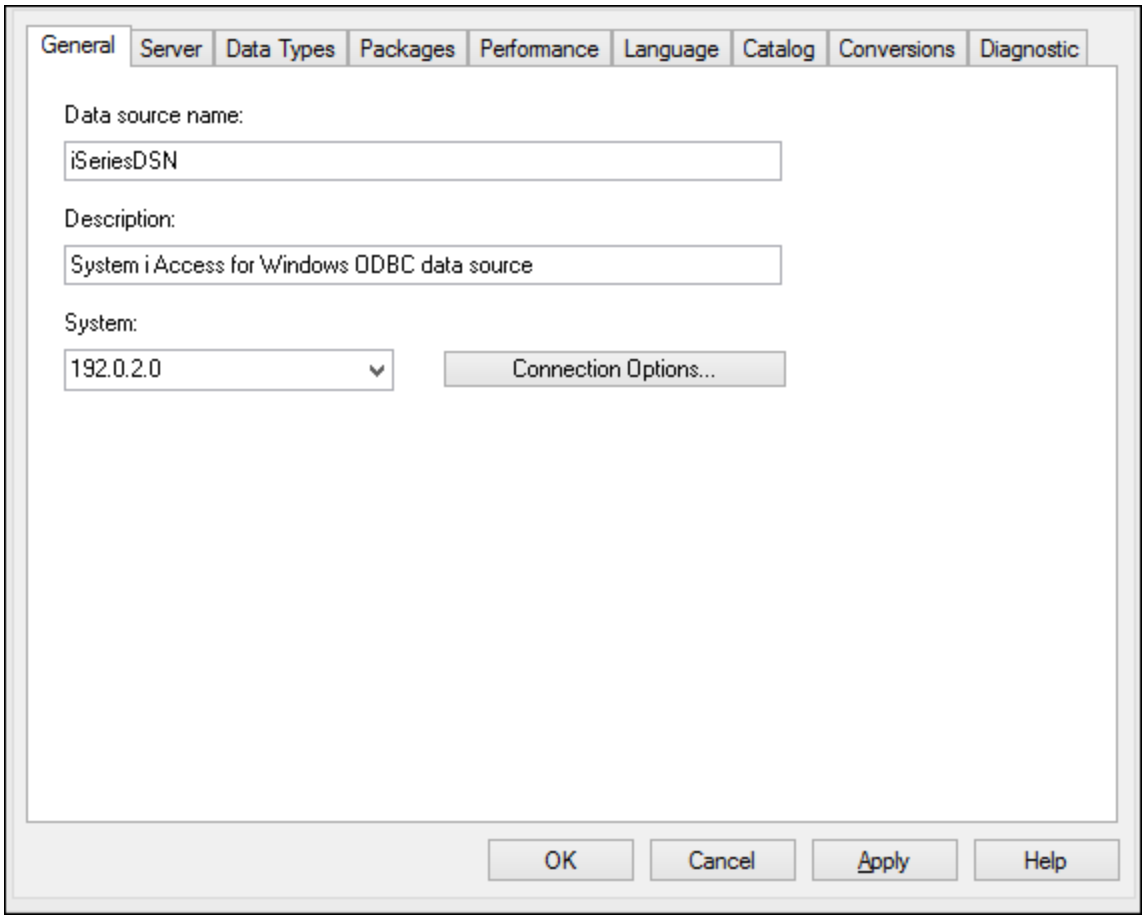
Para conectarse a IBM DB2 for i:

1. Inicie el asistente para la conexión de base de datos 524.
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**).

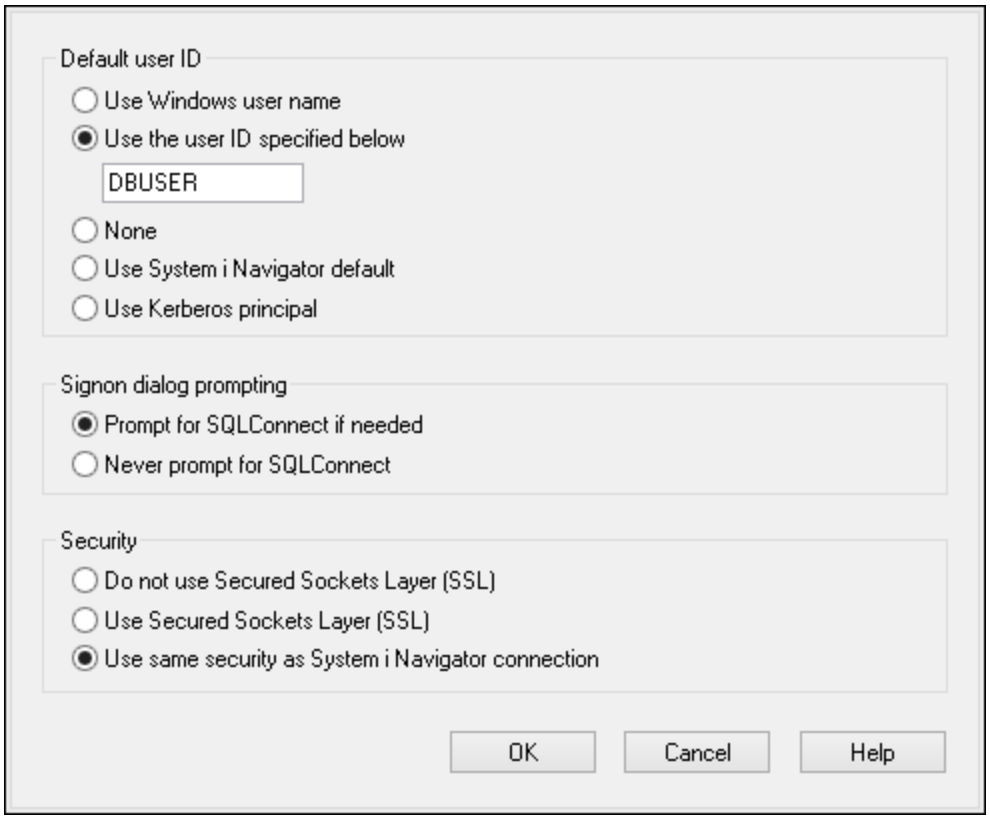


6. Introduzca el DSN y seleccione la conexión del cuadro combinado *Sistema*. En este ejemplo el DSN es **iSeriesDSN** y el sistema es **192.0.2.0**.

© 2016-2022 Altova GmbH



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**).



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**.

10.2.10.7 IBM Informix (JDBC)

Este ejemplo explica cómo conectarse a una base de datos IBM por JDBC.

Requisitos:

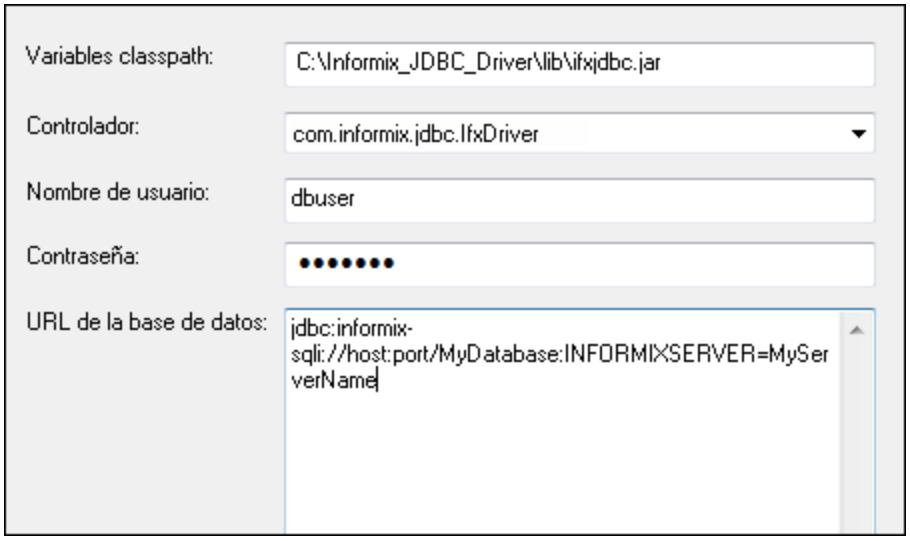
- Debe tener instalado Java Runtime Environment (JRE) o Java Development Kit (JDK). Este último puede ser el JDK de Oracle o uno de código abierto, como Oracle OpenJDK. StyleVision identifica la ruta de acceso al equipo virtual Java (JVM) a partir de estas ubicaciones, en este orden: a) la ruta personal 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 StyleVision (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*.

 © 2016-2022 Altova GmbH

- 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 524.
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* 547).
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**.

10.2.10.8 MariaDB (ODBC)

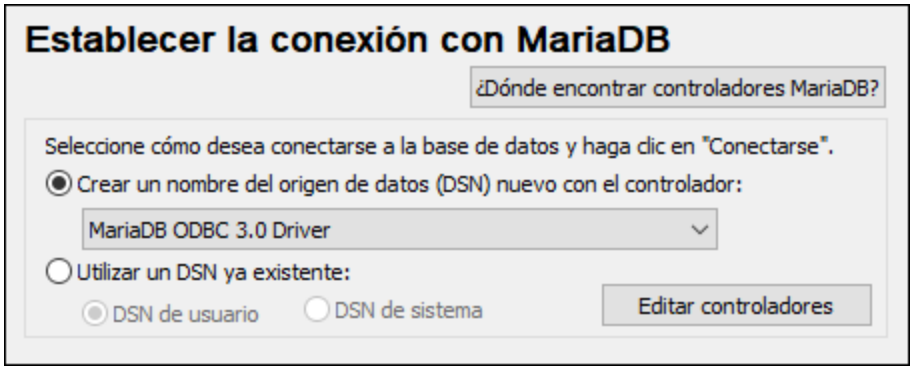
Este ejemplo explica cómo conectarse a un servidor de base de datos MariaDB por ODBC.

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.

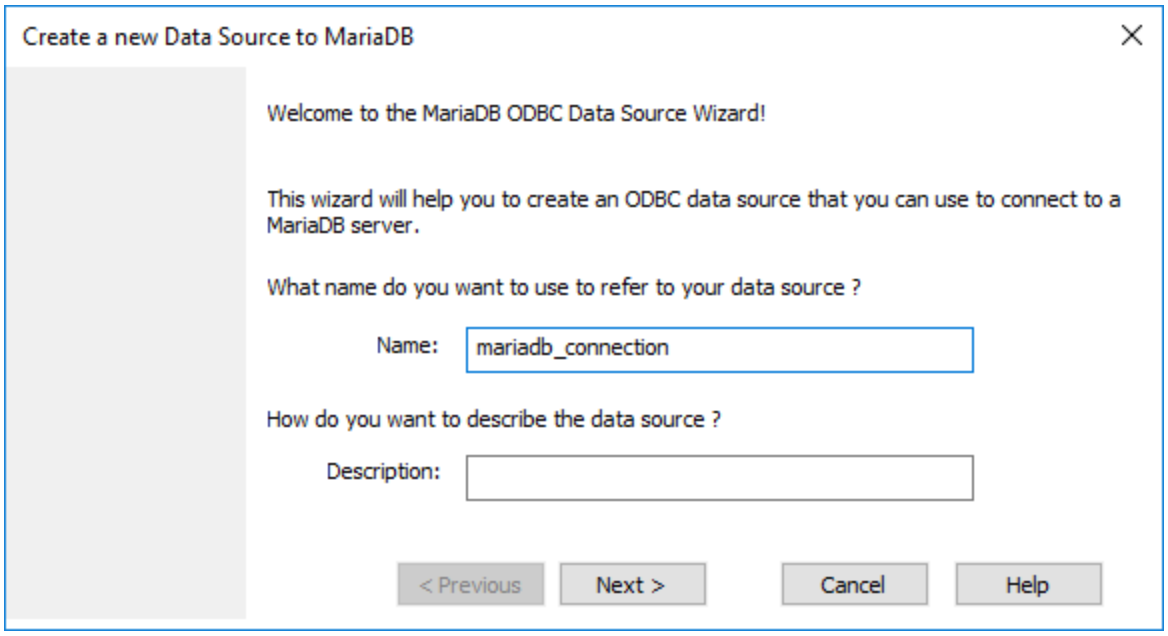
Para conectarse a MariaDB por ODBC:

1. Inicie el asistente para la conexión de base de datos 524.
2. Seleccione *MariaDB (ODBC)* y después haga clic en **Siguiente**.

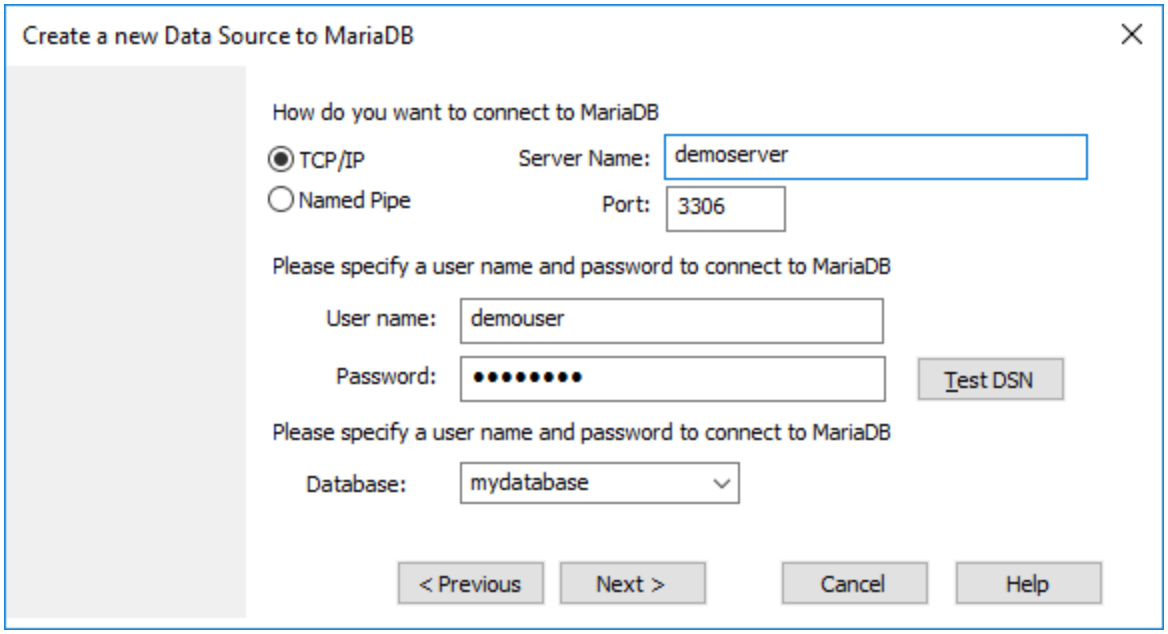


3. Seleccione *Crear un nombre del origen de datos (DSN) nuevo con el controlador* y elija el controlador **MariaDB ODBC 3.0 Driver**. Si este controlador no aparece en la lista, haga clic en **Editar controladores** y seleccione cualquier controlador MariaDB (la lista contiene todos los controladores ODBC que están instalados en el sistema operativo).
4. Haga clic en **Conectarse**.

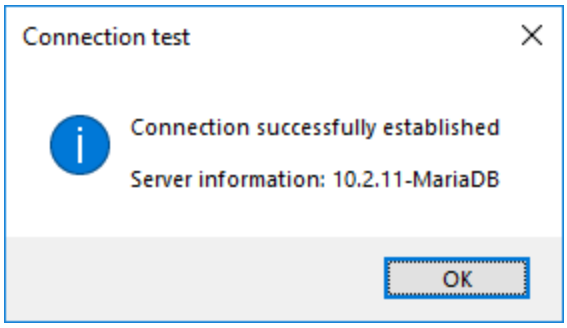
© 2016-2022 Altova GmbH



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:



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.

10.2.10.9 Microsoft Access (ADO)

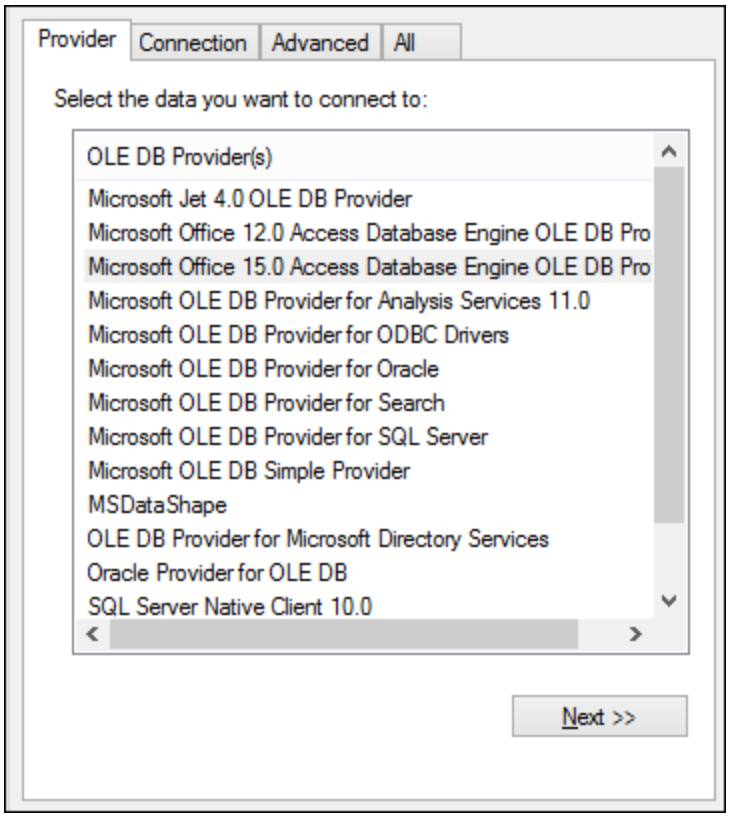
Lo más sencillo para conectarse a una base de datos Microsoft Access es seguir las instrucciones del asistente para la conexión de base de datos y buscar el archivo de base de datos, como se muestra en el apartado Conectarse a una base de datos Microsoft Access 531. También puede configurar explícitamente 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.

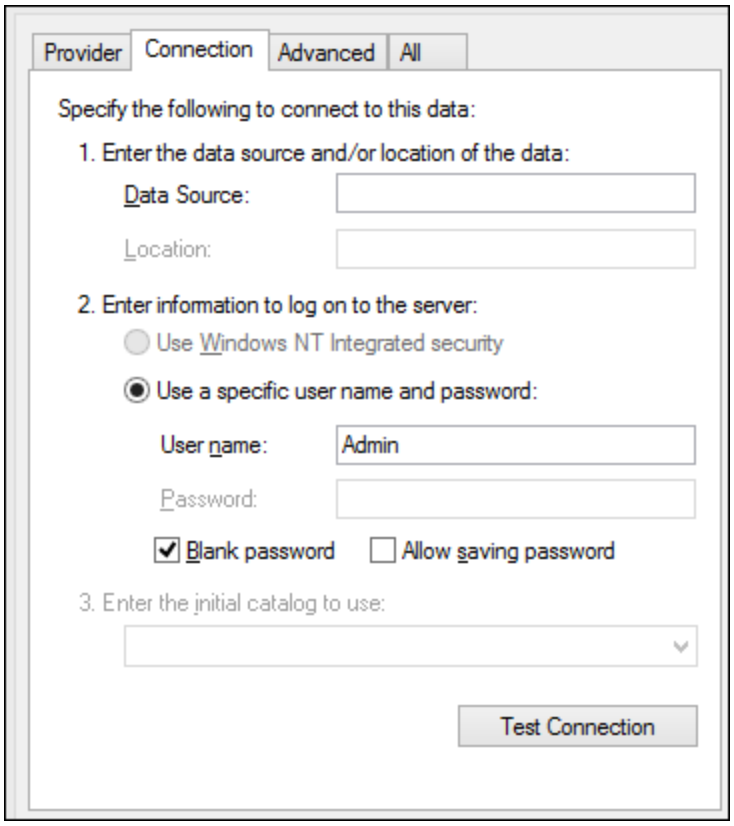
Para conectarse a una base de datos Microsoft Access protegida con contraseña:

1. Inicie el asistente para la conexión de base de datos 524.
2. Haga clic en **Conexiones ADO**.
3. Ahora haga clic en **Generar**.

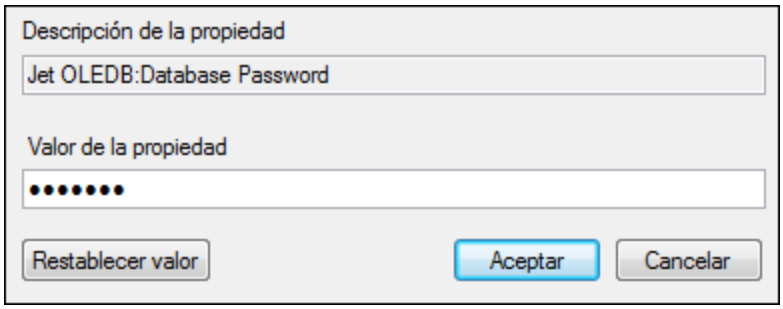
 © 2016-2022 Altova GmbH



4. Seleccione **Microsoft Office 15.0 Access Database Engine OLE DB Provider** y haga clic en **Siguiente**.



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`.

10.2.10.10 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**

 © 2016-2022 Altova GmbH

(MSOLEDBSQL), que puede descargar en https://docs.microsoft.com/en-us/sql/connect/oledb/download-oledb-driver-for-sql-server?view=sql-server-ver15.

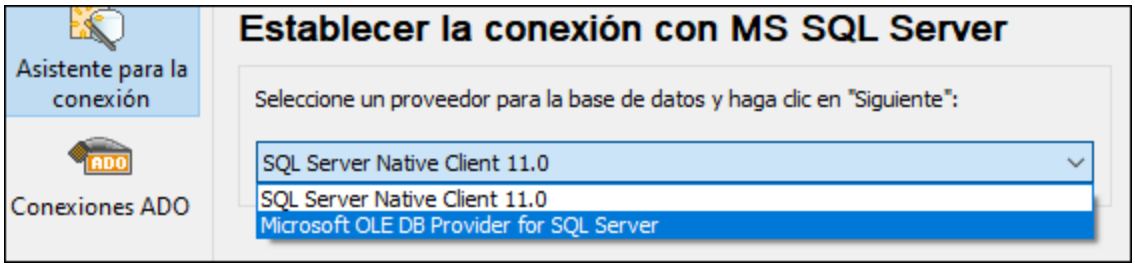
> Antes se seguir estas instrucciones, asegúrese de que descarga e instala el proveedor que mencionado más arriba en su equipo de trabajo. El proveedor ADO debe ser de la misma versión de plataforma que StyleVision (32 bits o 64 bits).

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 532.

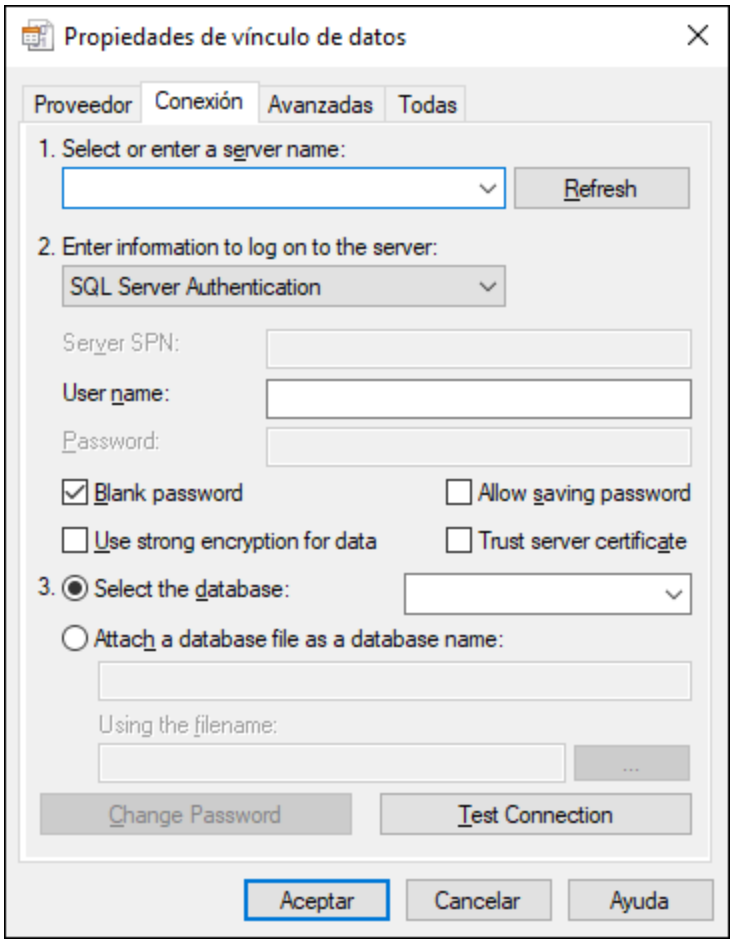
> Es un problema conocido que el proveedor de BD para SQL Server **Microsoft OLE DB Provider for SQL Server (SQLOLEDB)** tiene dificultades para enlazar parámetros de consultas complejas como las expresiones comunes de tabla (CTE) e instrucciones SELECT anidadas.

Para conectarse a SQL Server:

1. Inicie el asistente de conexión 524.
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.

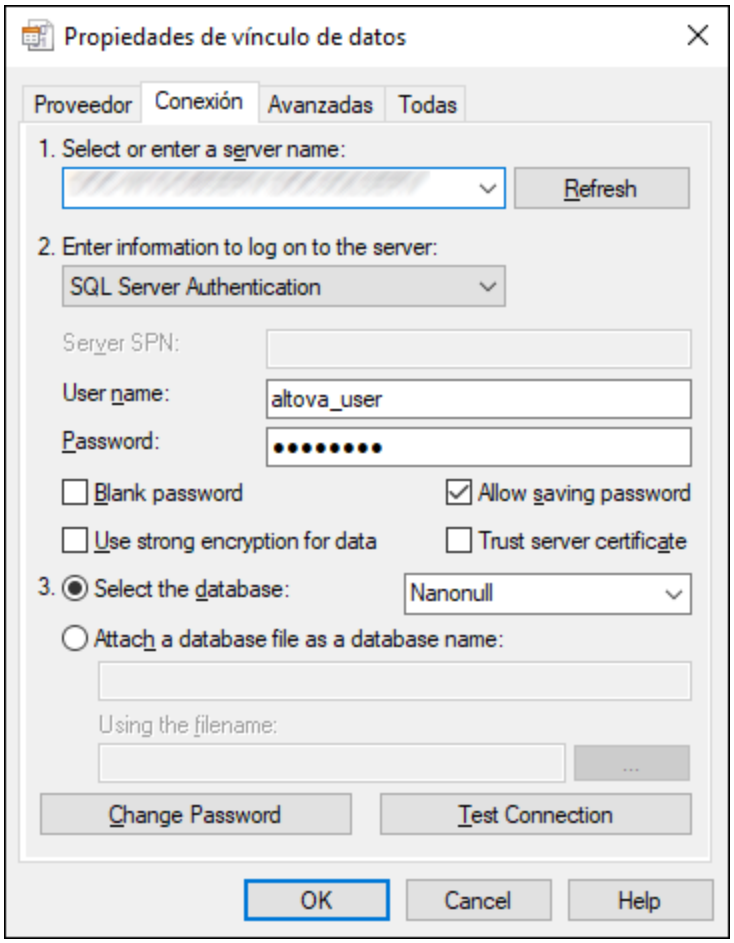


3. Haga clic en **Siguiente**. Se abre el cuadro de diálogo "Propiedades de enlace de datos".



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").

 © 2016-2022 Altova GmbH



7. Para comprobar la conexión haga clic en **Comprobar conexión**. Este paso es opcional, pero recomendamos no saltárselo.
8. Haga clic en **Aceptar**.

10.2.10.11 Microsoft SQL Server (ODBC)

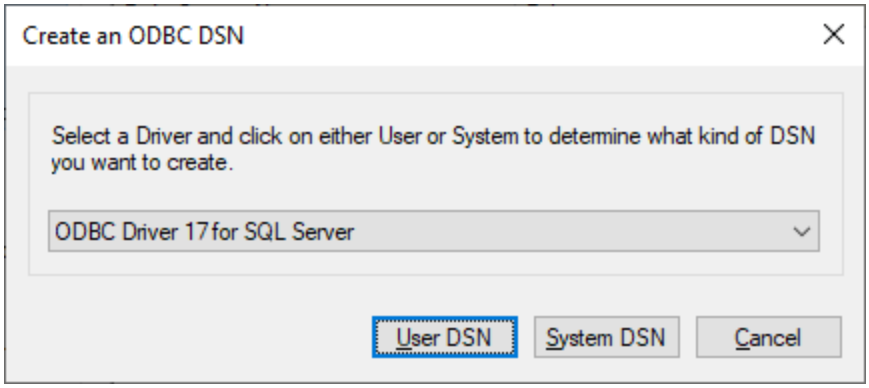
En este ejemplo aprenderá a conectar una base de datos a SQL Server con ODBC.

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.

Para conectarse a SQL Server con ODBC:

1. Inicie el asistente para la conexión de base de datos 524.
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.

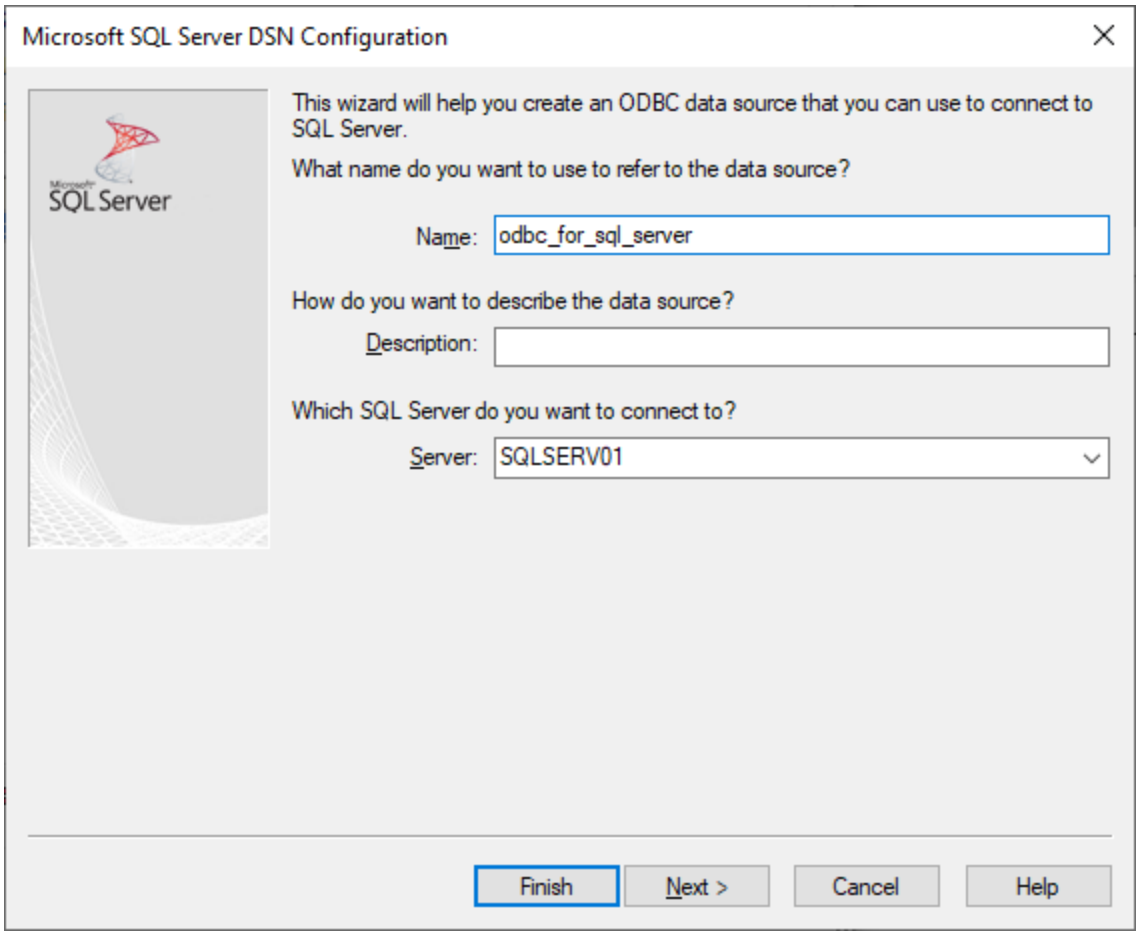


5. Haga clic en **DSN de usuario** (o en **DSN de sistema** si está creando un sistema DNS).

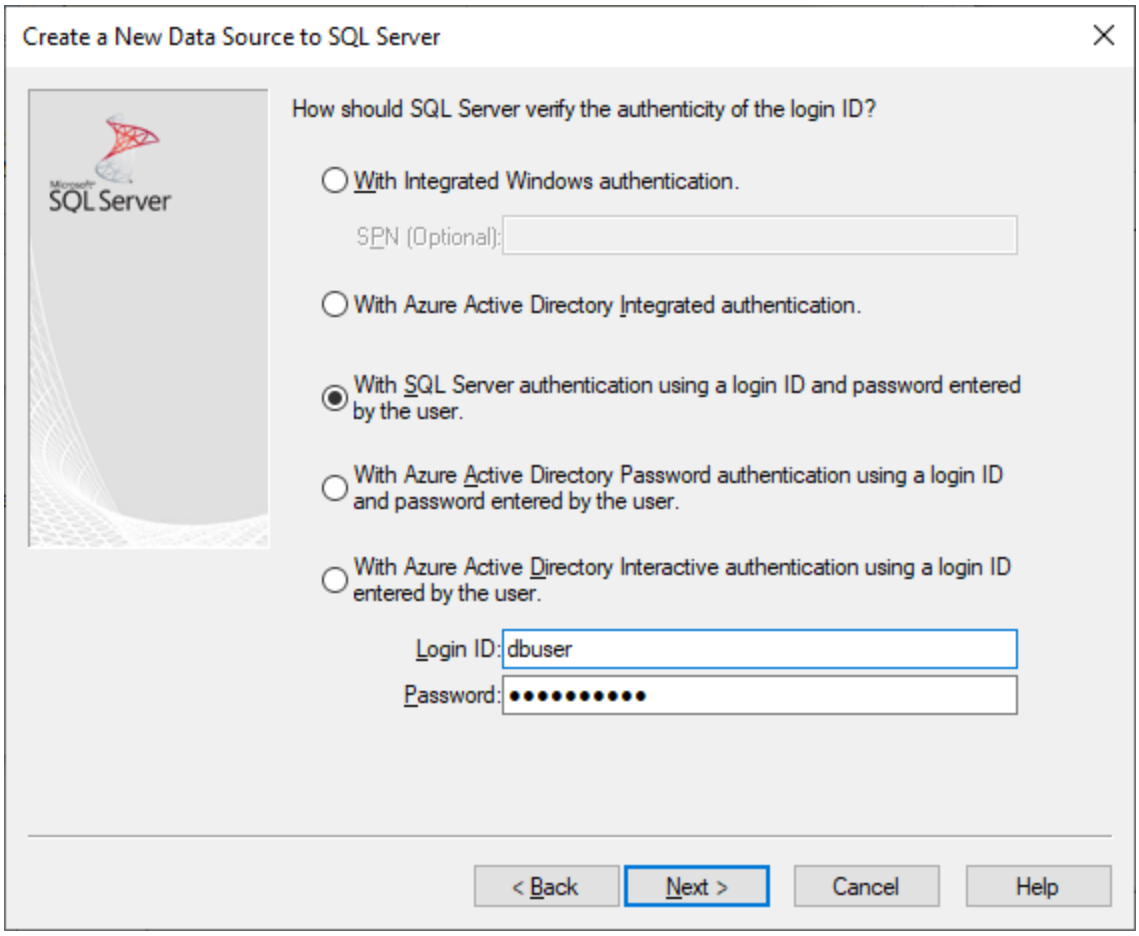
> Para crear un **Sistema DSN** necesita ejecutar StyleVision como administrador. Si lo que quiere es crear este sistema, salga del asistente y vuelva a ejecutar StyleVision como administrador y siga los pasos desde el principio.

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**).

© 2016-2022 Altova GmbH

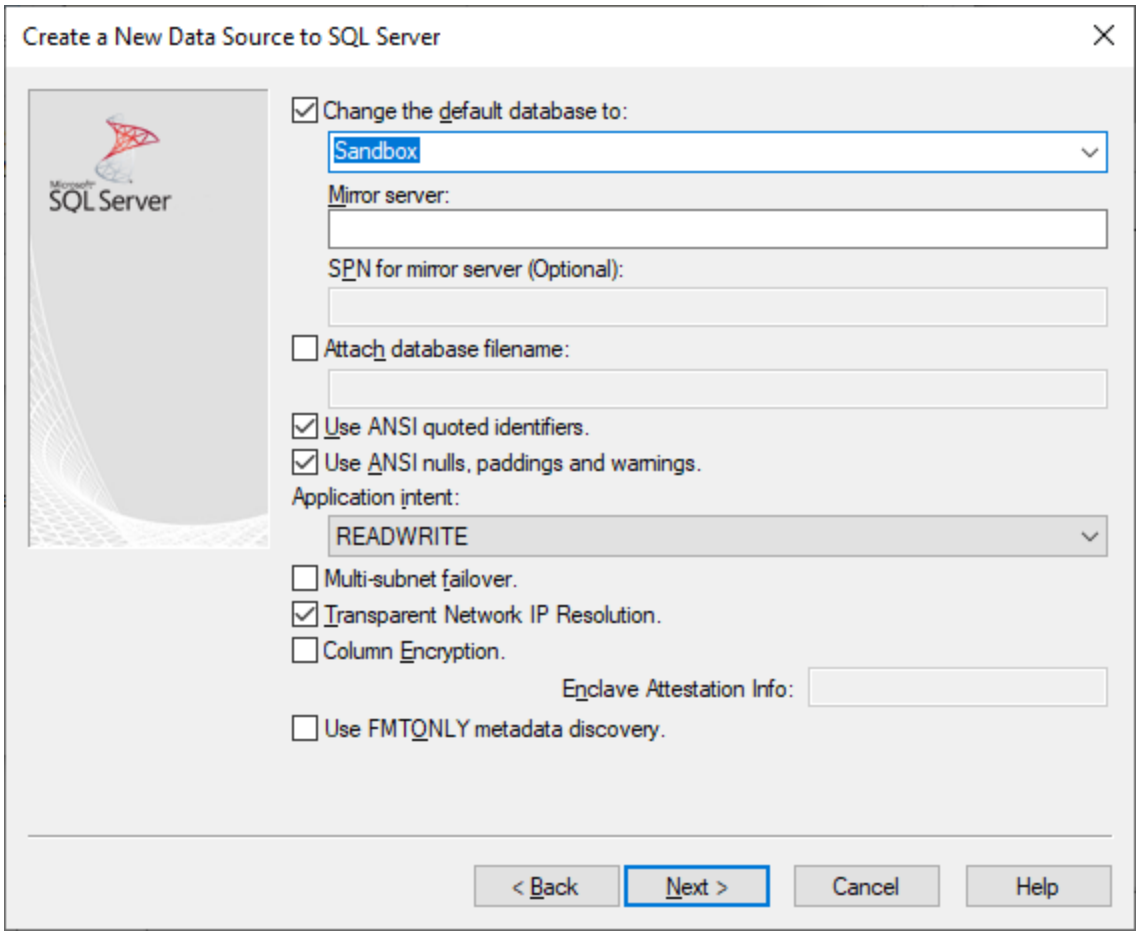


7. Si el servidor de BD se configuró para que permita conexiones de usuarios autenticados en el dominio Windows, seleccione **Con autenticación integrada de Windows**. De lo contrario, seleccione la opción que necesite. En este ejemplo usamos **Con autenticación de SQL Server...**, que necesita que se introduzcan el nombre de usuario y la contraseña en los campos correspondientes.

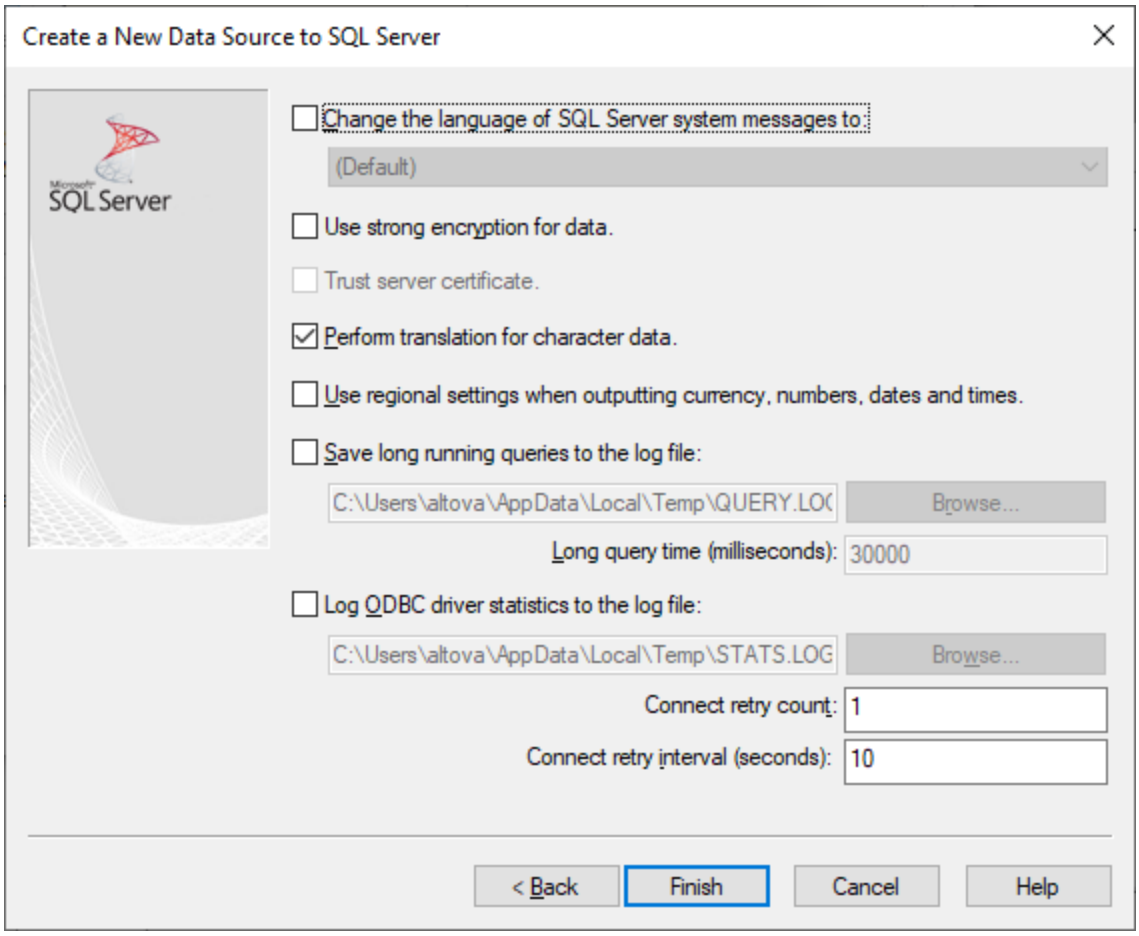


8. También puede marcar la casilla **Cambiar la base de datos predeterminada a** e introducir el nombre de la base de datos a la que se va a conectar (en este ejemplo, **Sandbox**).

© 2016-2022 Altova GmbH

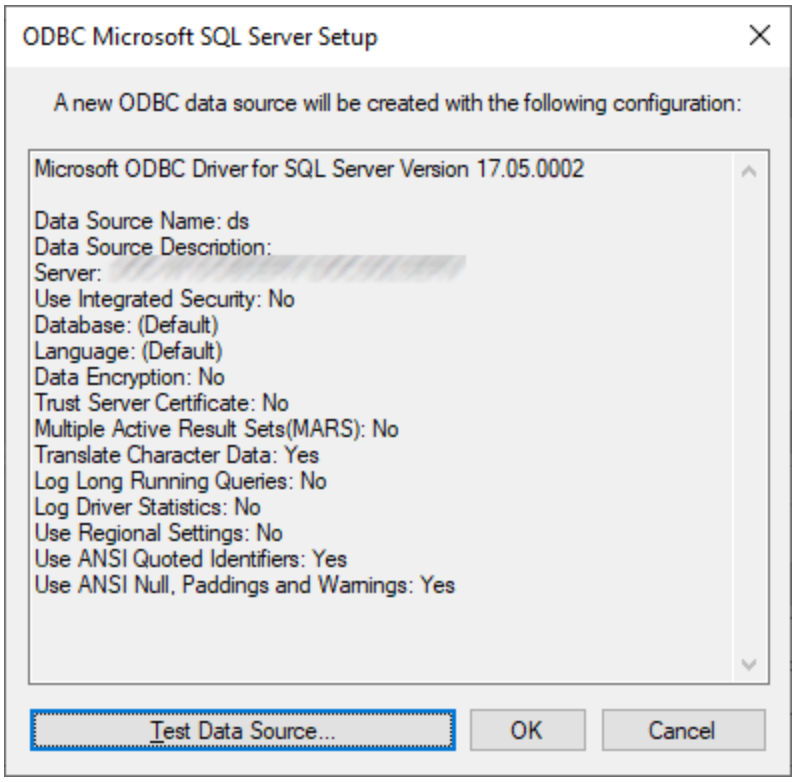


9. Haga clic en **Siguiente** y, si quiere, configure el resto de los parámetros de esta conexión.

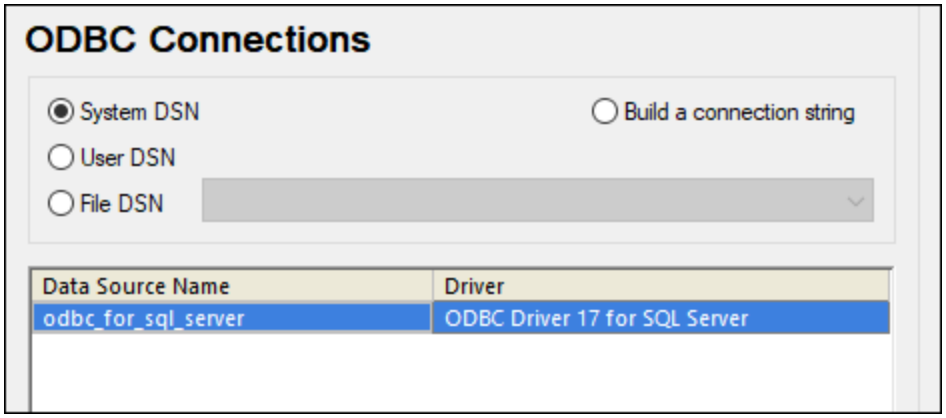


10. Haga clic en **Finalizar**. Aparece un cuadro de diálogo de conformación que contiene los detalles de la conexión.

© 2016-2022 Altova GmbH



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:



10.2.10.12 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.

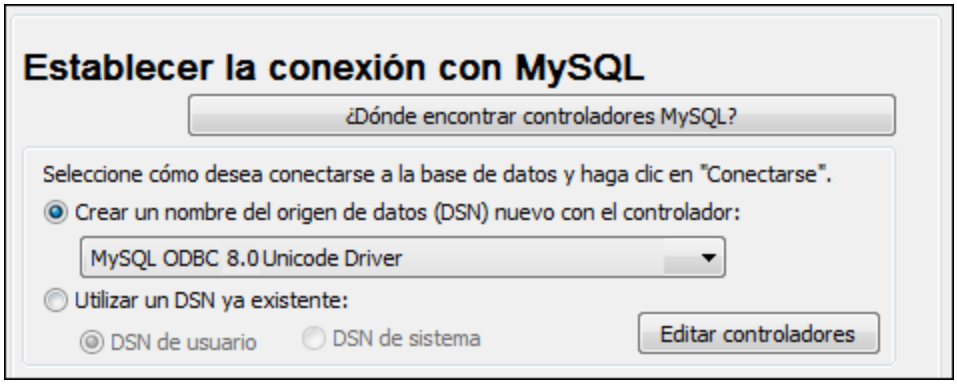
Requisitos previos:

- 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 StyleVision para plataformas de 64 bits.

Para conectarse a MySQL por ODBC:

1. Inicie el asistente para la conexión de base de datos 524.
2. Seleccione la opción *MySQL (ODBC)* y después haga clic en **Siguiente**.

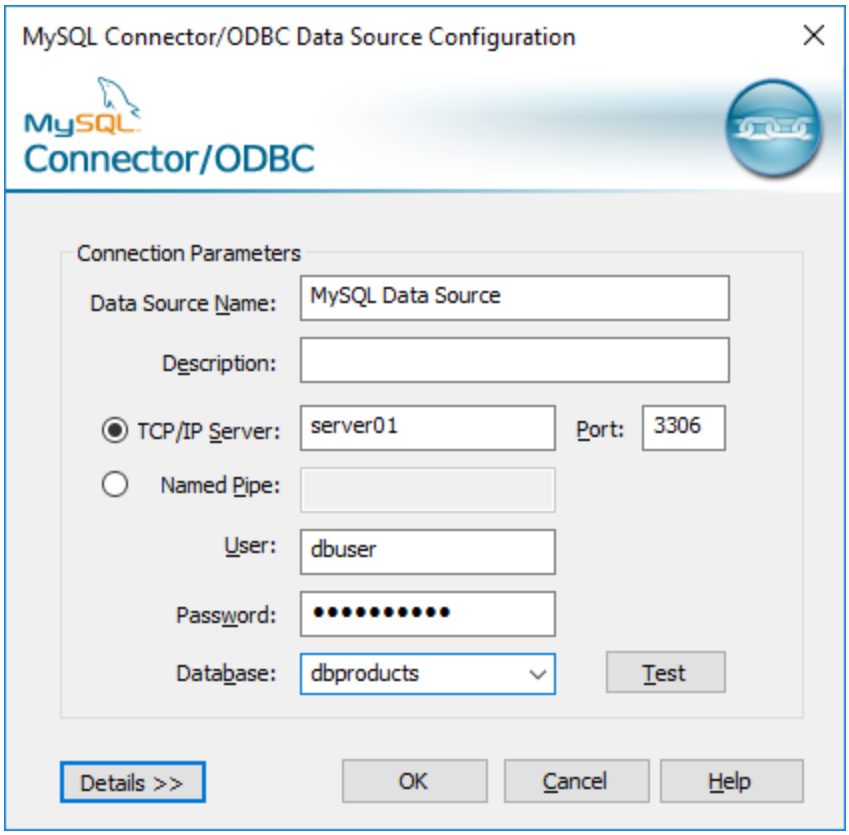


3. Seleccione la opción *Crear un DNS nuevo con el controlador* y seleccione un controlador MySQL. Si no hay ningún controlador MySQL en la lista, haga clic en **Editar controladores** y seleccione uno (la lista contiene todos los controladores ODBC que están instalados en el sistema).

   Si instaló la versión de StyleVision 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 543.

4. Haga clic en **Conectarse**.

 © 2016-2022 Altova GmbH



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.

10.2.10.13 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:

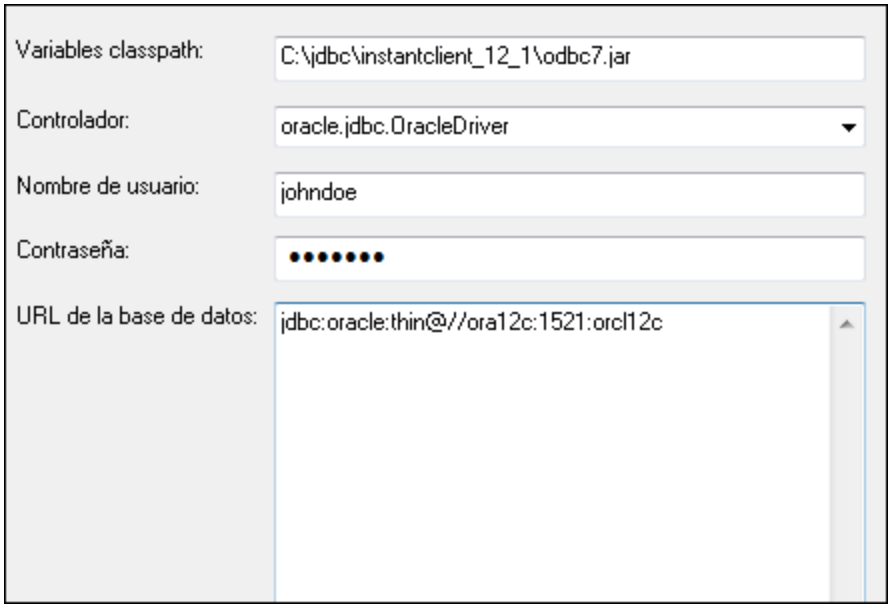
- JRE (Java Runtime Environment) o Java Development Kit (JDK) está instalado. Este último puede ser el JDK de Oracle o uno de código abierto, como Oracle OpenJDK. StyleVision identifica la ruta de

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 StyleVision (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 524.
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* 547).
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**.

 © 2016-2022 Altova GmbH

10.2.10.14 Oracle (ODBC)

Este ejemplo explica cómo conectarse a un servidor de base de datos Oracle desde StyleVision. 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 StyleVision. 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 541.

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)
    )
  )
```

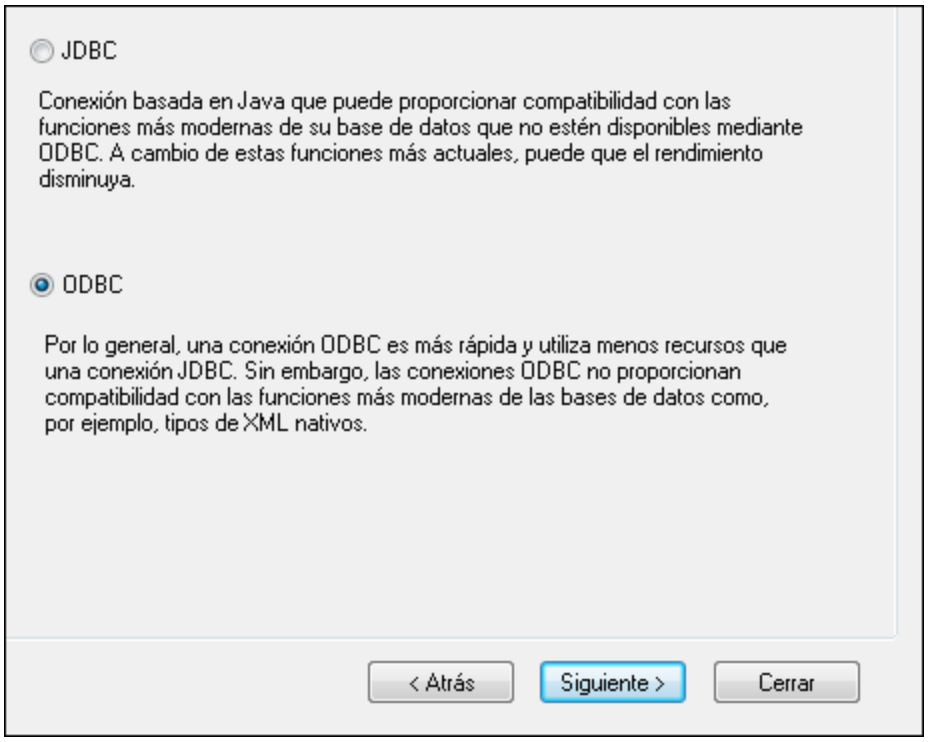
La ruta de acceso del archivo `tnsnames.ora` depende de la ubicación del directorio de inicio de Oracle. Por ejemplo, en el caso del cliente de base de datos Oracle 11.2.0, la ruta de acceso predeterminada del directorio de inicio podría ser:

```
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**.

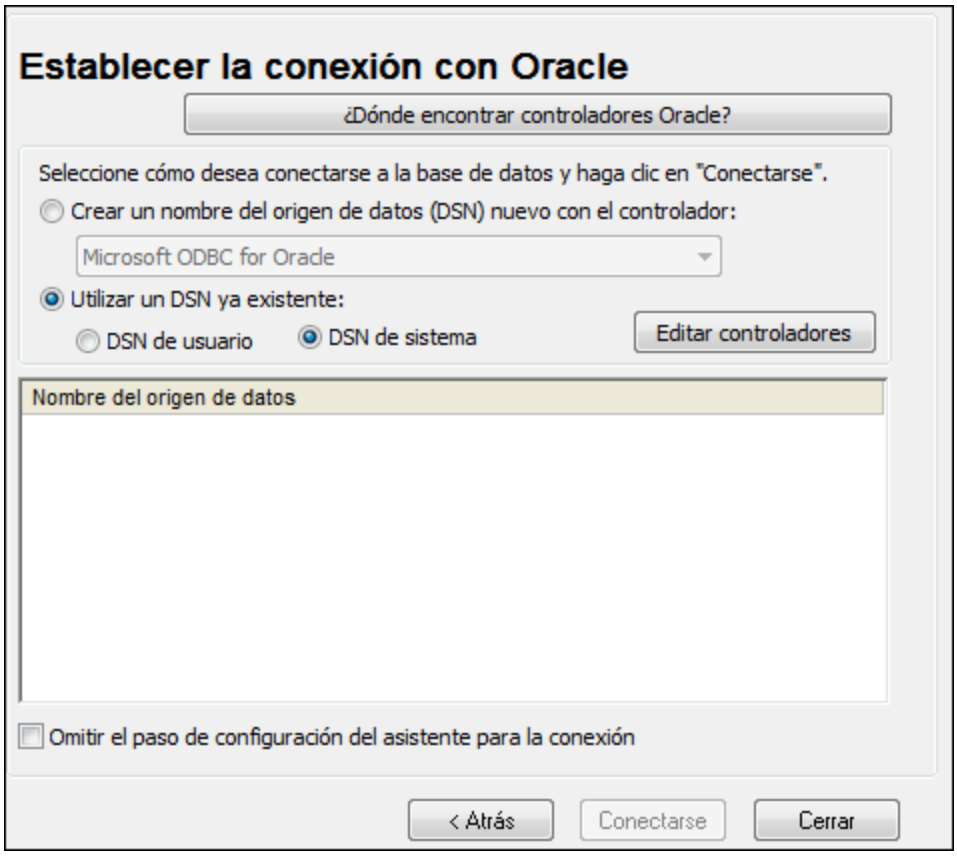
Para conectarse a Oracle por ODBC:

1. Inicie el asistente para la conexión de base de datos 524.
2. Seleccione la opción *Oracle (ODBC / JDBC)* y después haga clic en **Siguiente**.

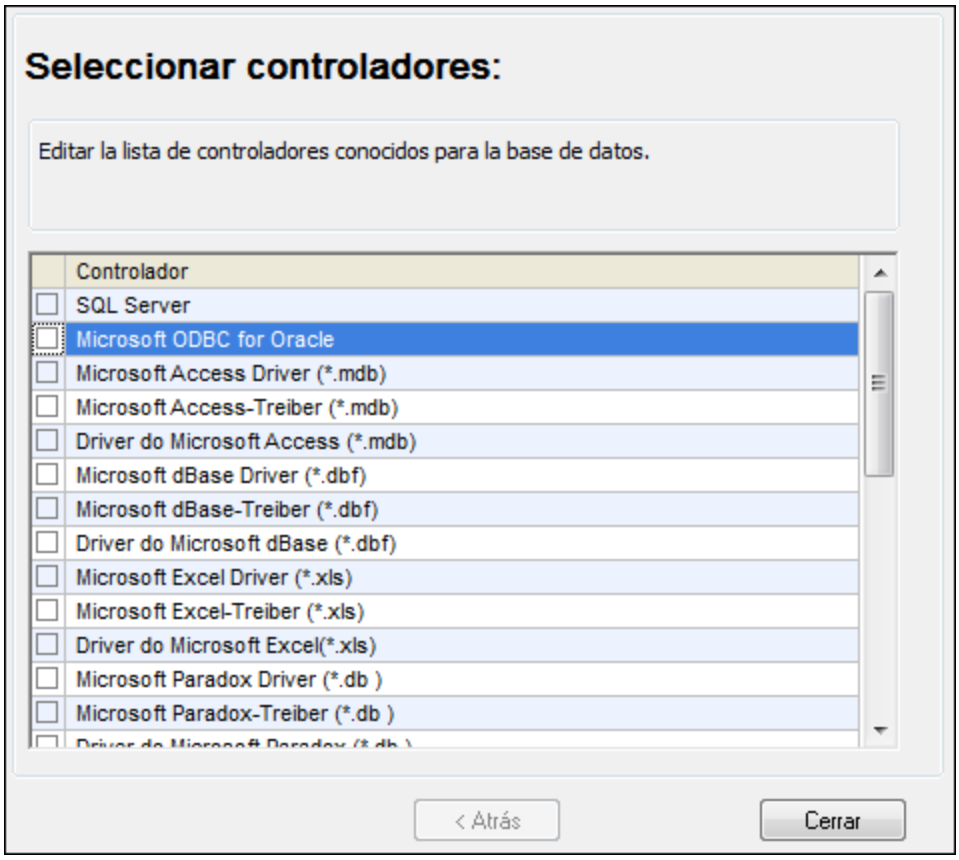


3. Seleccione el botón de opción *ODBC*.

© 2016-2022 Altova GmbH

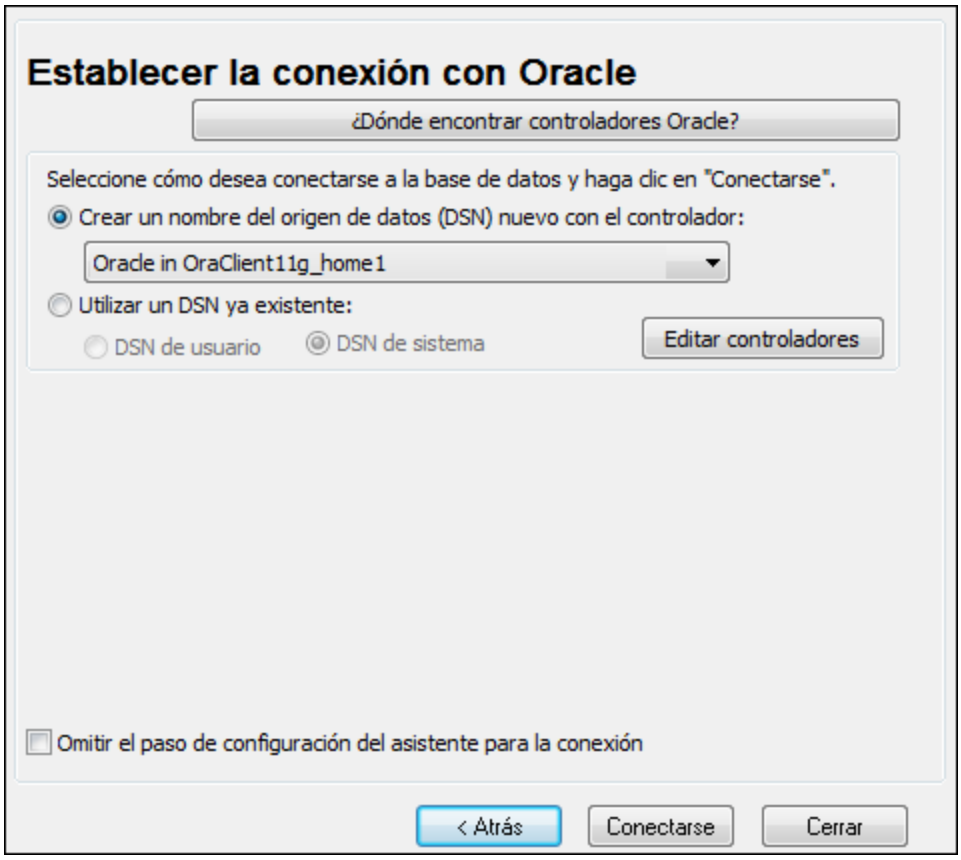


4. Haga clic en **Editar controladores**.



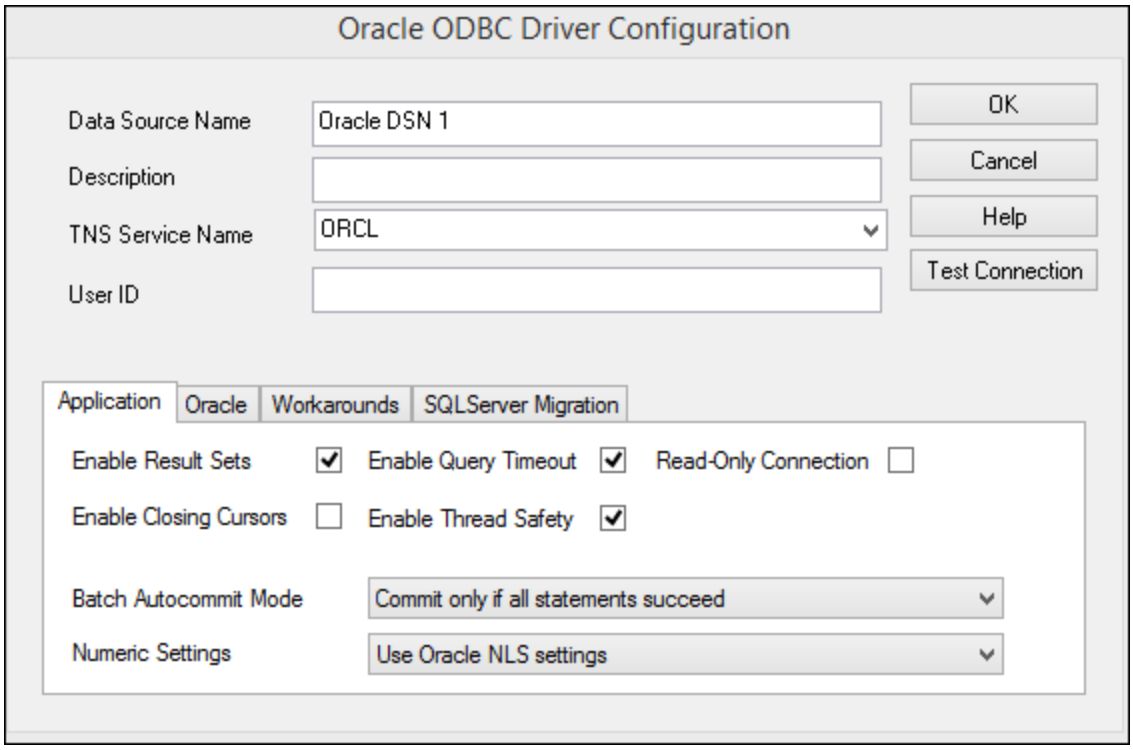
5. Seleccione los controladores Oracle que desea usar (en este ejemplo usamos **Oracle in OraClient11g_home1**). La lista incluye todos los controladores Oracle que están disponibles en el sistema después de instalar el cliente Oracle.
6. Haga clic en **Atrás**.
7. Seleccione la opción *Crear un DSN nuevo con el controlador* y después seleccione el controlador de Oracle elegido en el paso nº 4.

© 2016-2022 Altova GmbH

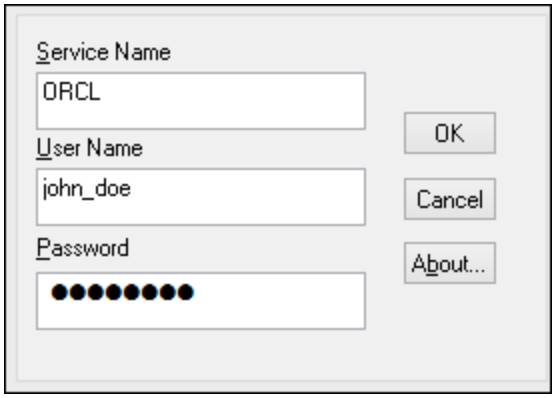


Recomendamos no utilizar el controlador **Microsoft ODBC for Oracle**. Microsoft recomienda utilizar el controlador ODBC que ofrece Oracle (consulte http://msdn.microsoft.com/en-us/library/ms714756%28v=vs.85%29.aspx)

8. Haga clic en **Conectarse**.



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 [589]). 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.

© 2016-2022 Altova GmbH

10.2.10.15 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 526).

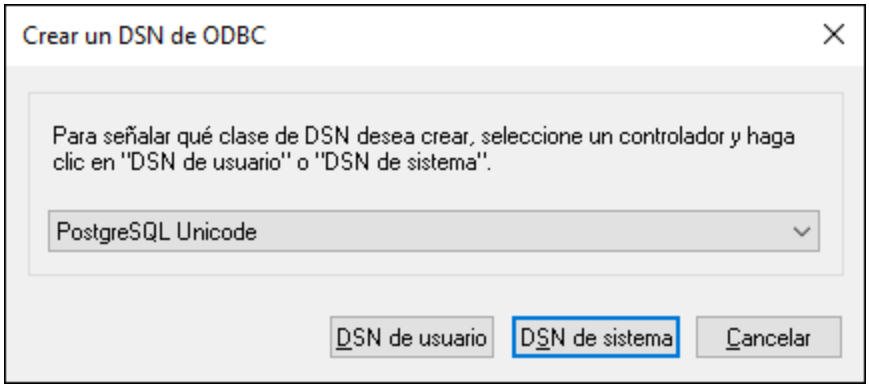
Nota: también se puede conectar a un servidor de base de datos PostgreSQL directamente (es decir, sin el controlador ODBC). Consulte el apartado Conexiones PostgreSQL 549 para más información.

Requisitos:

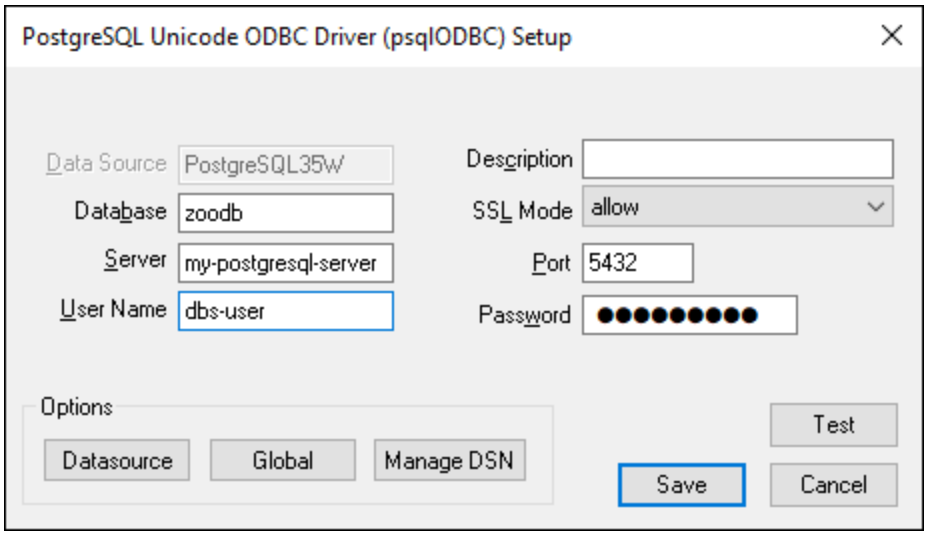
- El controlador *psqlODBC* está instalado en el sistema.
- Disponer de los datos de conexión: servidor, puerto, base de datos, nombre de usuario y contraseña.

Para conectarse a PostgreSQL por ODBC:

1. Inicie el asistente para la conexión de base de datos 524.
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.

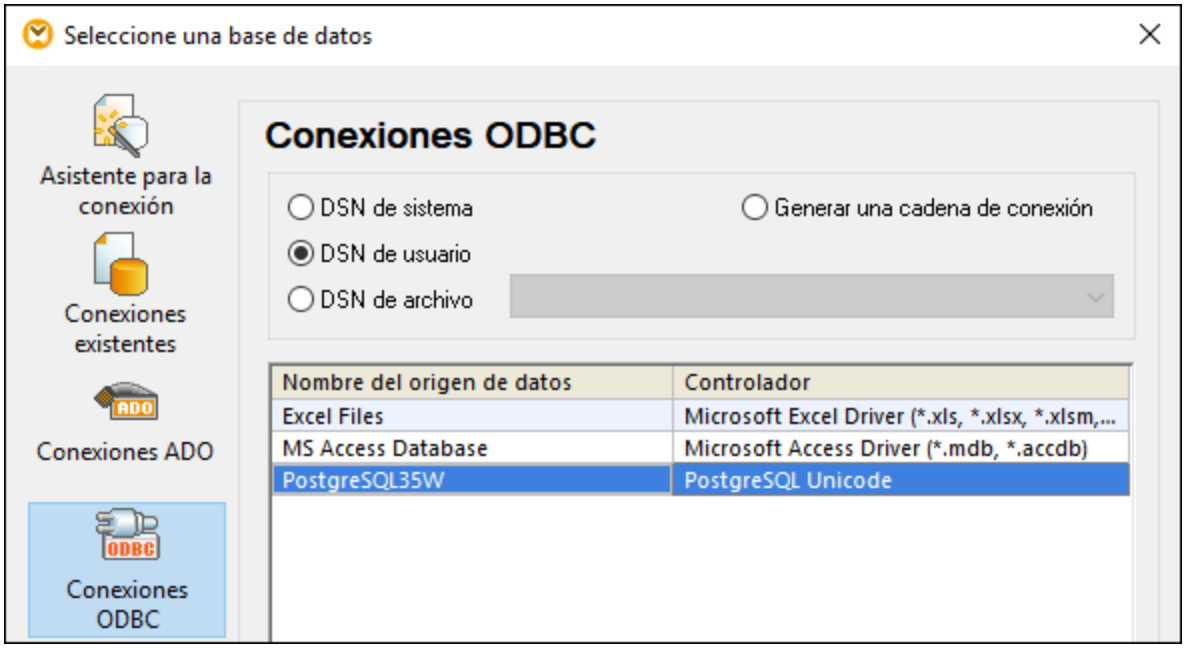


5. Haga clic en **DNS de usuario**.



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**.



10.2.10.16 Progress OpenEdge (JDBC)

En este apartado encontrará instrucciones para conectarse a un servidor de base de datos Progress OpenEdge 11.6 por JDBC.

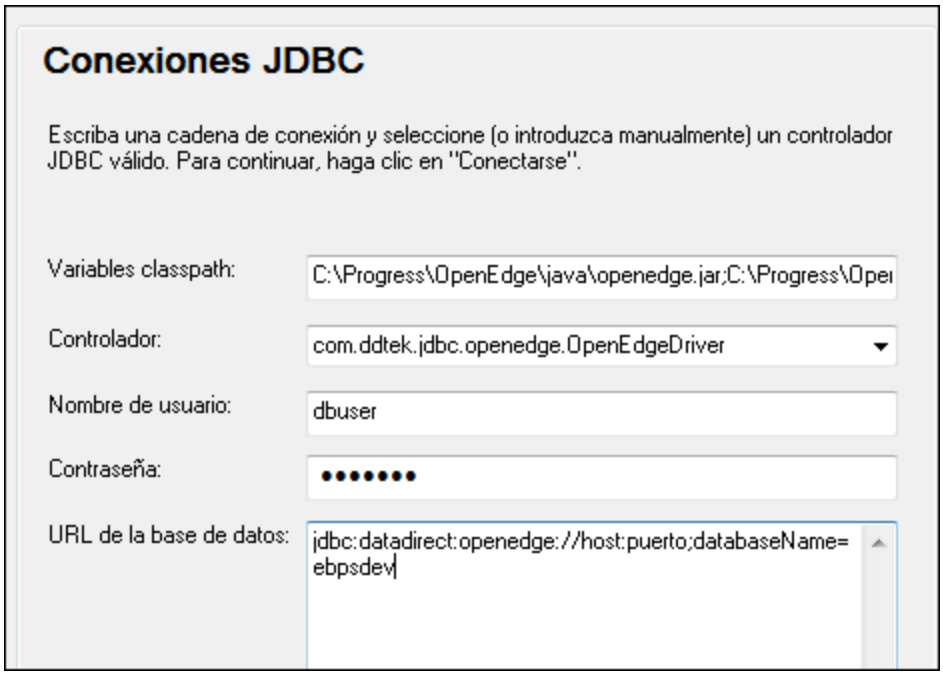
 © 2016-2022 Altova GmbH

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. StyleVision 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 StyleVision (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 524.
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* 547).
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.



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
```

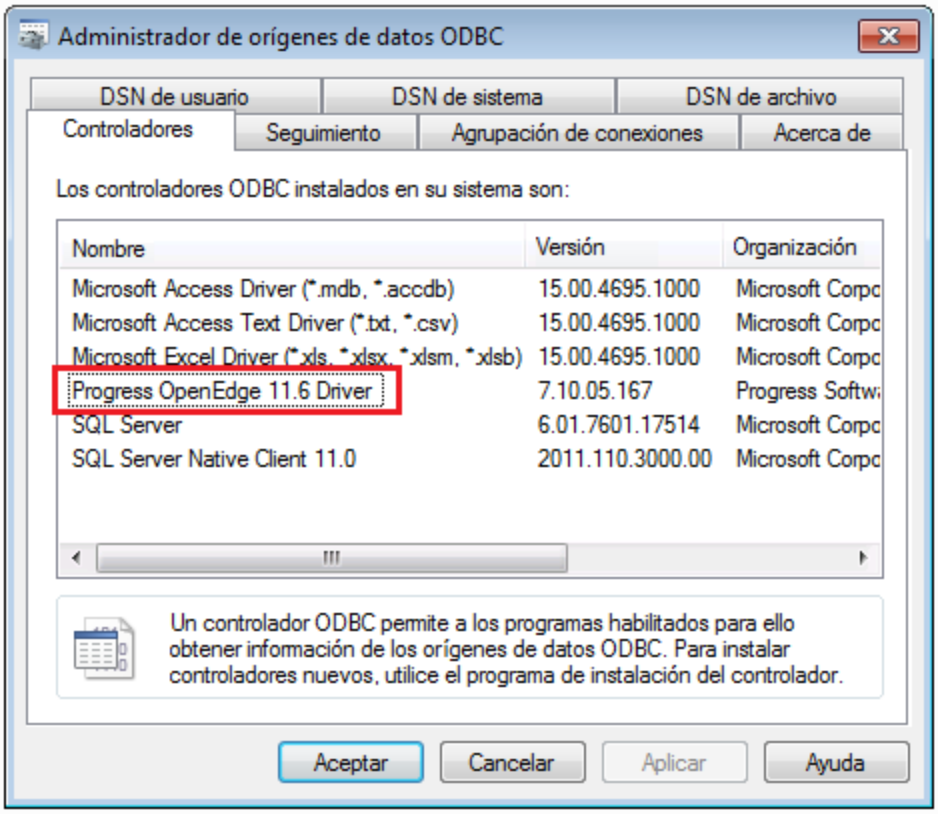
7. Haga clic en **Conectarse**.

10.2.10.17 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 526). Si trabaja con la versión de 32 bits de StyleVision, 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* 543).

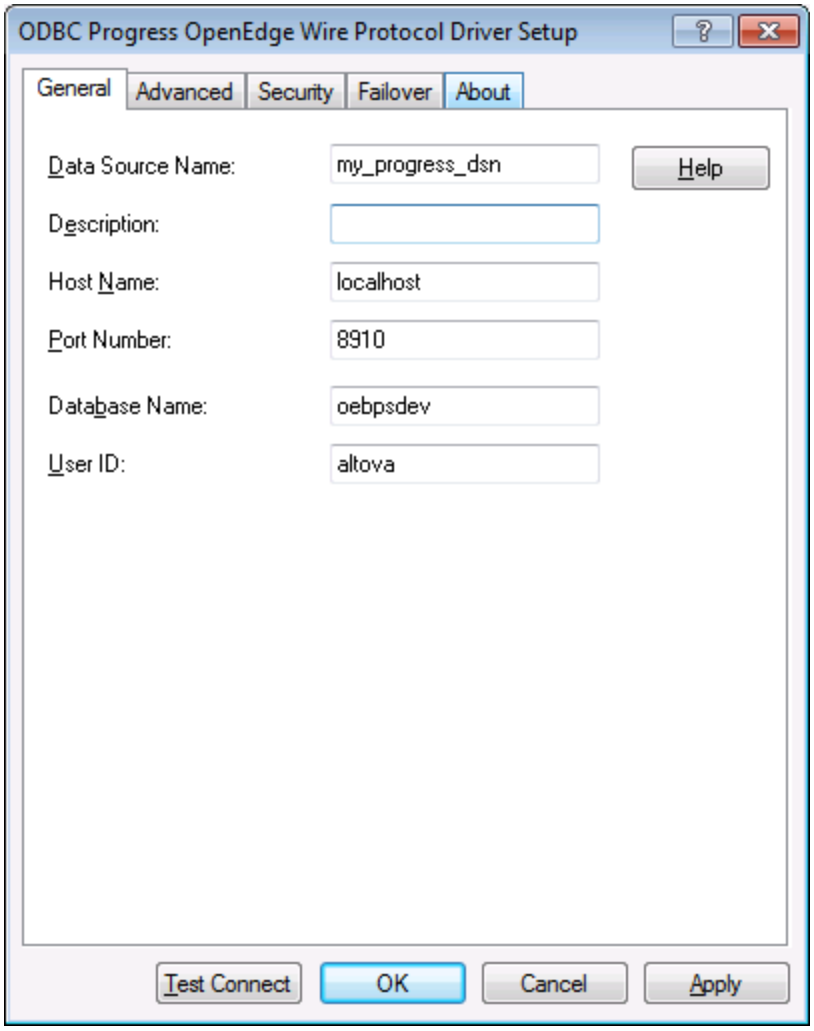


- Disponer de los datos de conexión: nombre de host, número de puerto, nombre de la base de datos, id. de usuario y contraseña.

Para conectarse a Progress OpenEdge por ODBC:

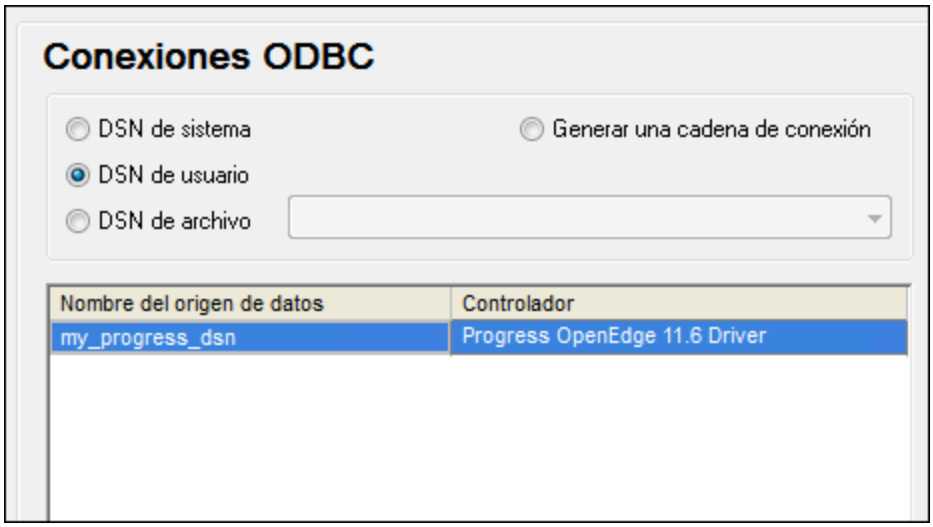
1. Inicie el asistente para la conexión de base de datos 524.
2. Haga clic en el botón Conexiones ODBC.
3. Haga clic en *DSN de usuario* (o en *DSN de sistema* o *DSN de archivo*).
4. Ahora haga clic en el icono **Agregar** .
5. Seleccione el controlador **Progress OpenEdge** de la lista y haga clic en el botón **DSN de usuario** (o **DSN de sistema** según corresponda).

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.

© 2016-2022 Altova GmbH



8. Para terminar haga clic en **Conectarse**.

10.2.10.18 Sybase (JDBC)

Este ejemplo explica cómo conectarse a un servidor de base de datos Sybase por JDBC.

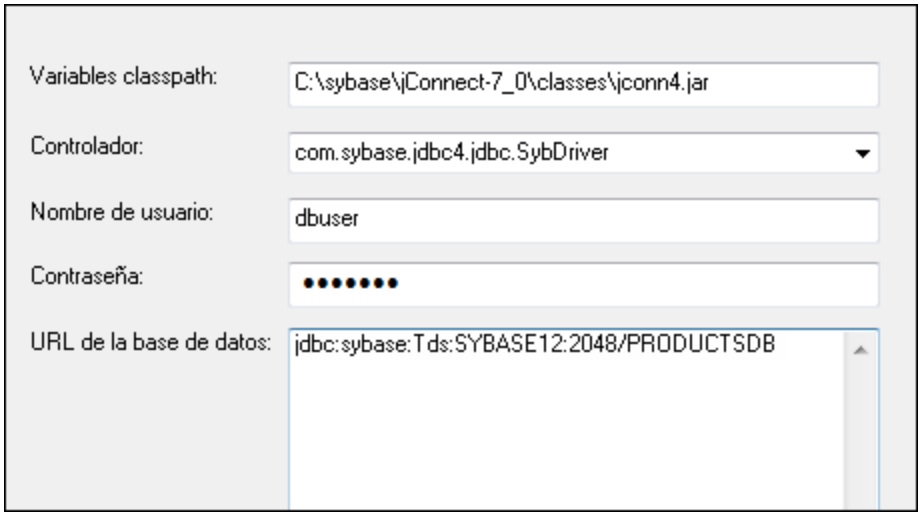
Requisitos:

- 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. StyleVision 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 StyleVision (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 524.
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* 547).

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**.

10.2.10.19 Teradata (JDBC)

Este ejemplo explica cómo conectarse a un servidor de base de datos Teradata por JDBC.

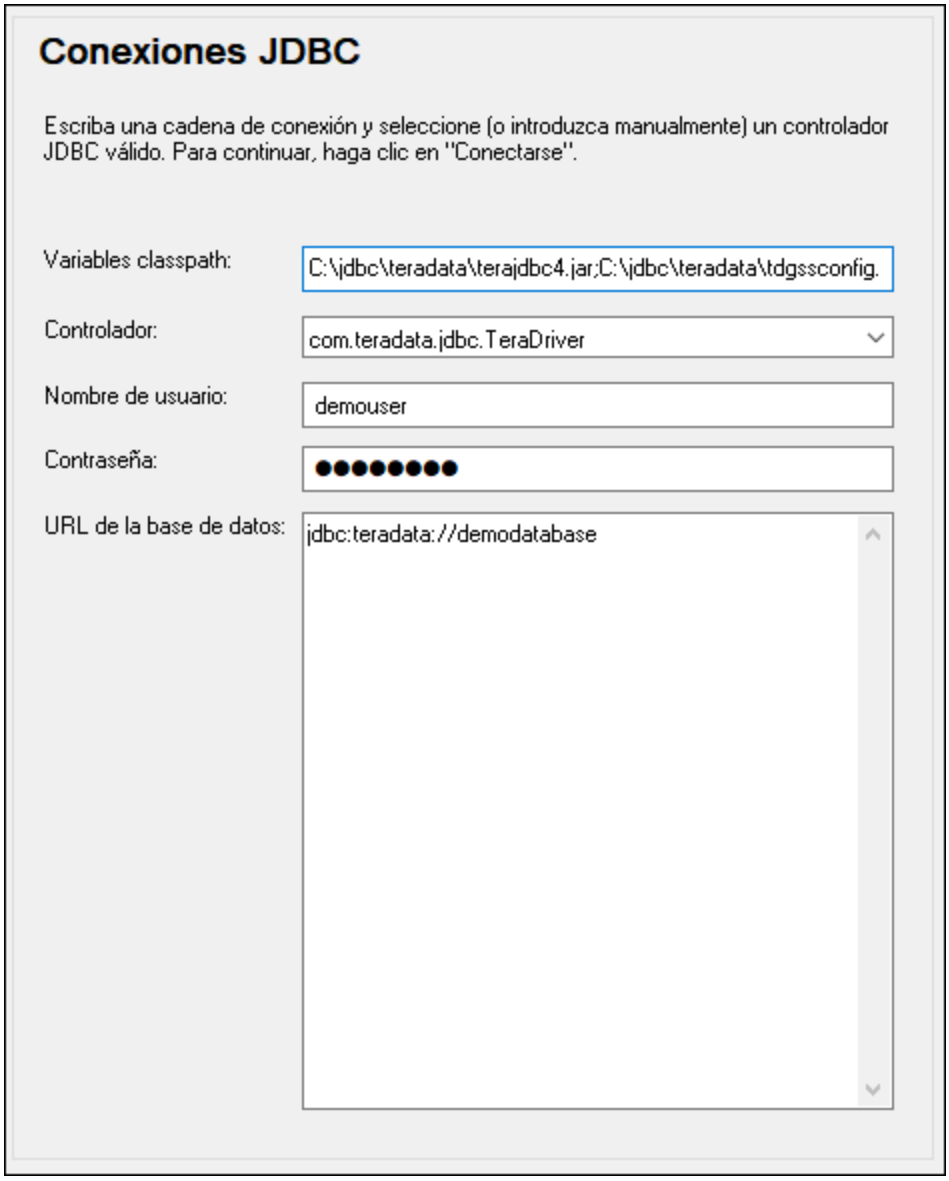
Requisitos:

- 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. StyleVision 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 StyleVision (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 524.

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 547).
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).

```
jdbc:teradata://nombreServidorBaseDatos
```

7. Por último, haga clic en **Conectarse**.

10.2.10.20 Teradata (ODBC)

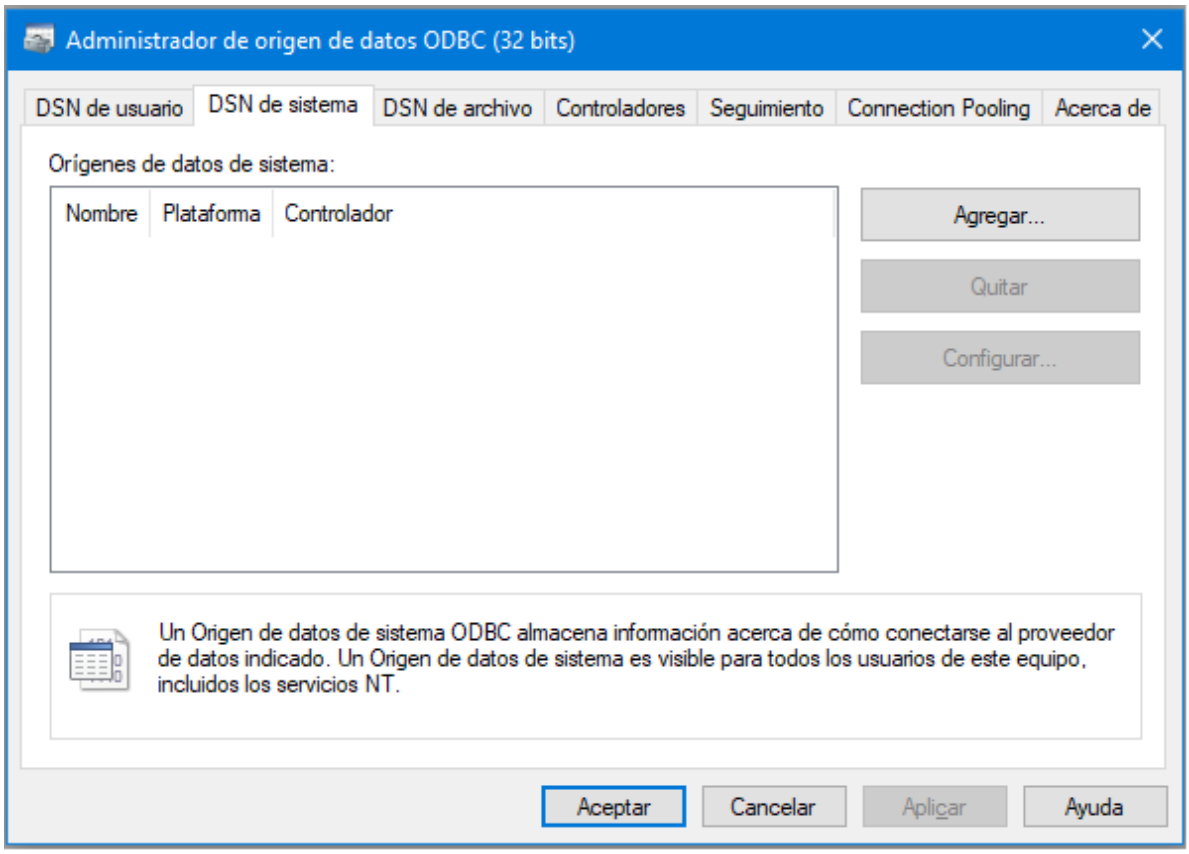
Este ejemplo explica cómo conectarse a un servidor de base de datos Teradata por ODBC.

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.

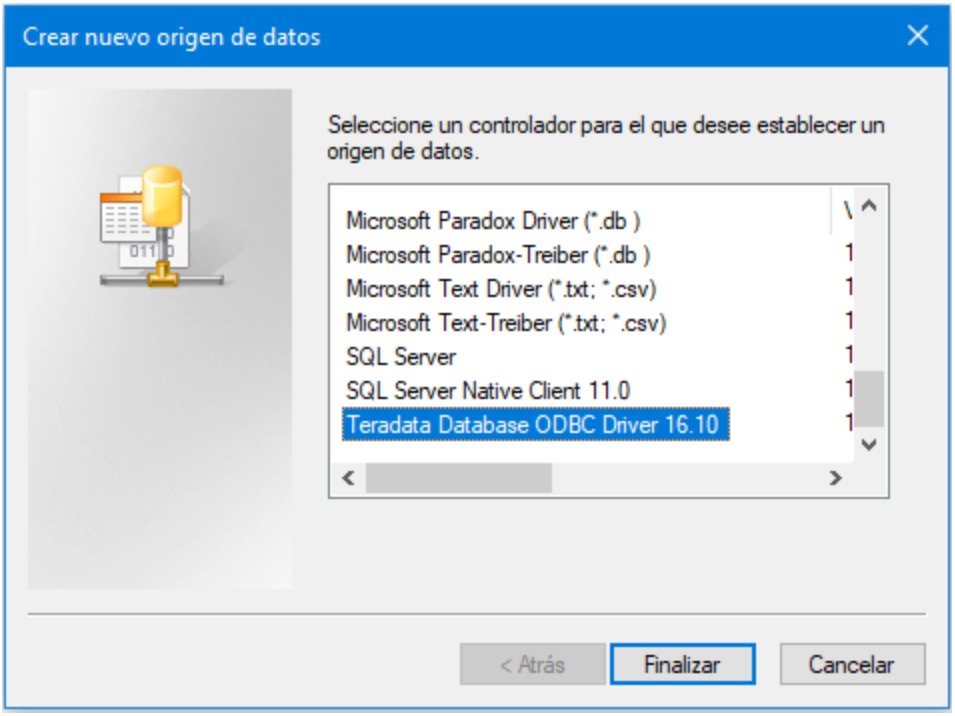
Para conectarse a Teradata por ODBC:

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 StyleVision de 64 bits a partir de ese momento.

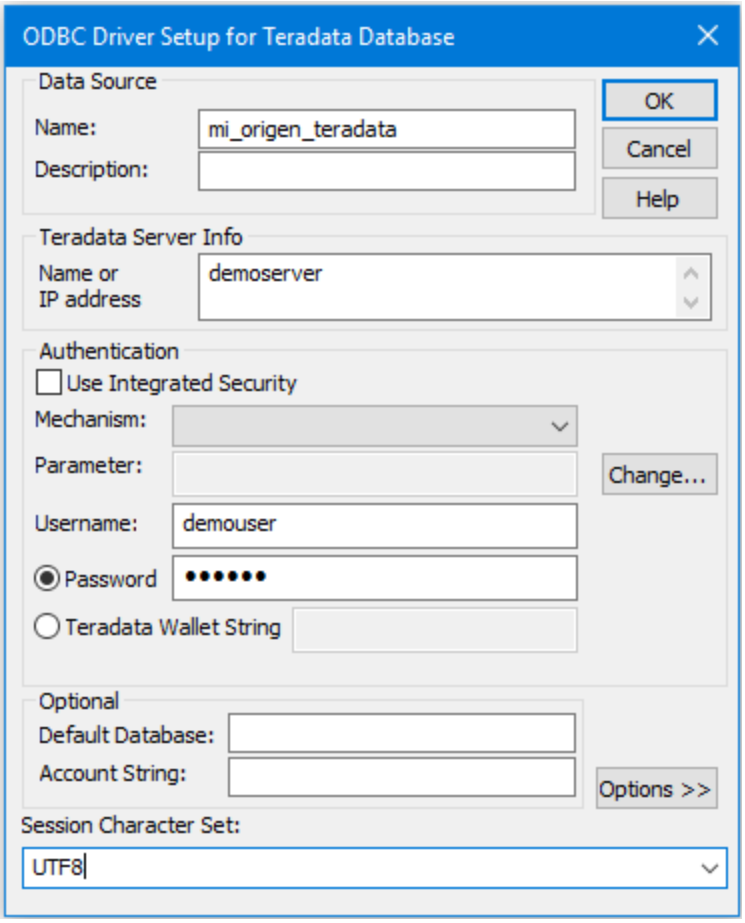


 © 2016-2022 Altova GmbH

2. Haga clic en la pestaña *DSN de sistema* y después haga clic en **Agregar**.

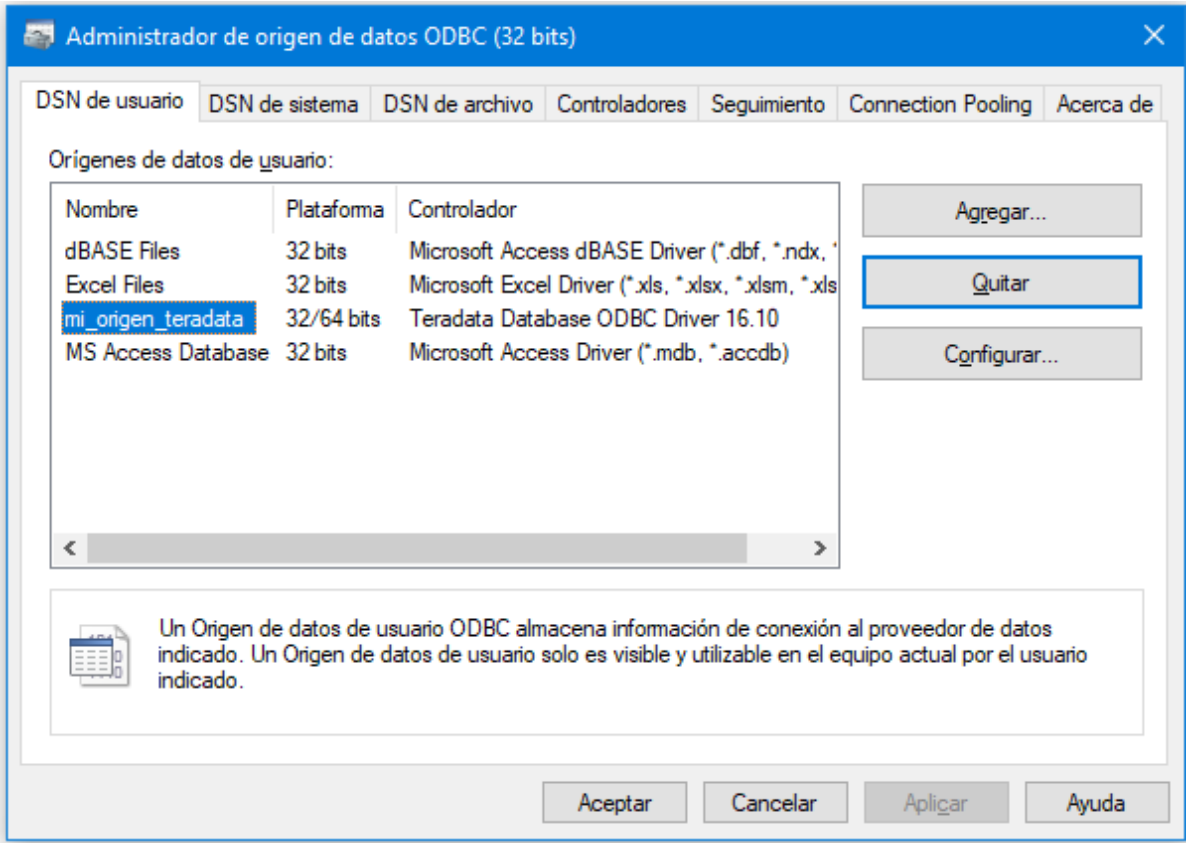


3. Seleccione **Teradata Database ODBC Driver** y después haga clic en **Finalizar**.

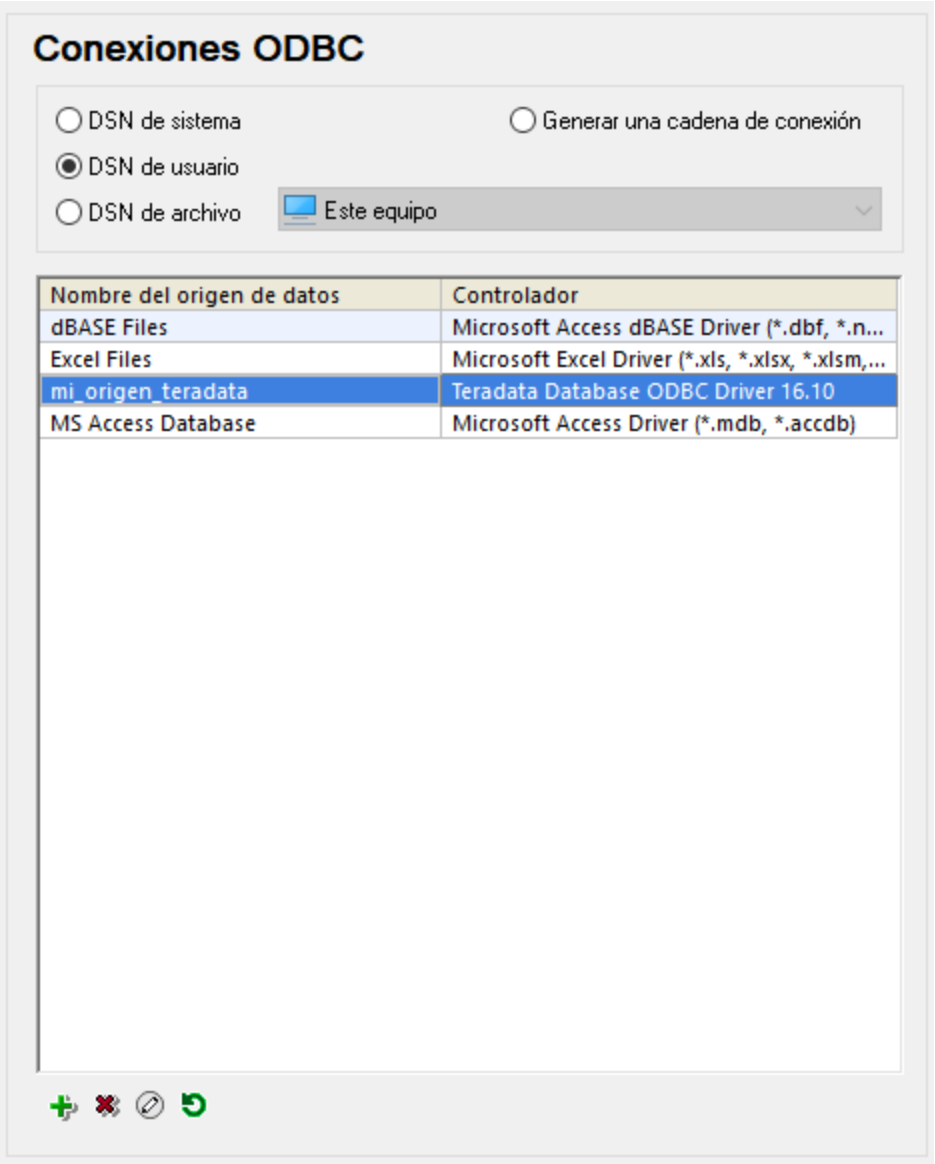


4. Introduzca el nombre y, si quiere, una descripción que le ayude a identificar este origen de datos ODBC más adelante. Además, deberá introducir las credenciales de la conexión de base de datos (servidor de BD, usuario y contraseña) y, si quiere, seleccione una base de datos.
5. Haga clic en **Aceptar**. El origen de datos aparecerá en la lista.

© 2016-2022 Altova GmbH



6. Ejecute StyleVision e inicie el asistente para la conexión de base de datos 524.
7. Haga clic en **Conexiones ODBC**.



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.

© 2016-2022 Altova GmbH

10.3 Seleccionar datos de la BD

Para seleccionar el esquema y los datos XML que se usarán en el diseño SPS es necesario seleccionar tablas, una celda o un esquema concreto, dependiendo de si la BD es una BD no XML (como MS Access) o una BD XML (IBM DB2 versión 9.0, etc). La selección del esquema y de los datos XML de la BD se denomina *selección de datos de la BD* y se lleva a cabo inmediatamente después de establecer la conexión con la BD.

El método utilizado para seleccionar datos en la BD depende del tipo de BD con el que se esté trabajando:

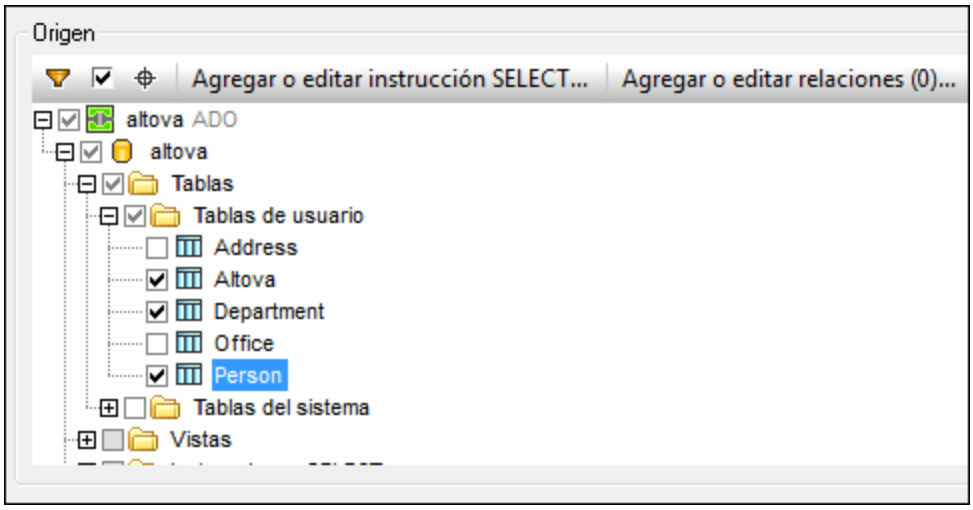
- Si se trata de una base de datos no XML 609: seleccione las tablas para las que desea crear el diseño SPS. StyleVision genera automáticamente: (i) un esquema basado en la estructura de las tablas y (ii) archivos XML temporales basados en este esquema y con los datos de las tablas seleccionadas. Para aprender a seleccionar las tablas consulte el apartado Seleccionar datos de la BD | Bases de datos no XML 609.
- Si se trata de una base de datos XML 617, entonces debe hacer dos cosas: primero, seleccionar la celda XML 617 de la BD donde están almacenados los datos XML necesarios. Estos datos XML se cargan como archivo XML de trabajo 30 del diseño SPS. Después debe seleccionar el esquema 617 en el que se basará el diseño SPS. Esto se explica en el apartado Seleccionar datos de la BD | Bases de datos XML 617.

10.3.1 Bases de datos no XML

Una vez establecida la conexión con la BD no XML, aparece el cuadro de diálogo "Insertar objetos de la base de datos". Este cuadro de diálogo se divide en dos paneles. El panel superior se llama *Origen* y contiene una representación gráfica de las tablas de la BD. En este panel puede seleccionar las tablas que necesita para trabajar en el diseño SPS. StyleVision genera un esquema XML y archivos de datos XML a partir de las tablas seleccionadas. El panel inferior se llama *Vista previa* y muestra una vista previa del contenido de la tabla seleccionada.

Panel Origen

El panel *Origen* ofrece una representación gráfica de las tablas de la BD (*imagen siguiente*). Seleccione las tablas que necesita para su diseño SPS marcando las casillas pertinentes.

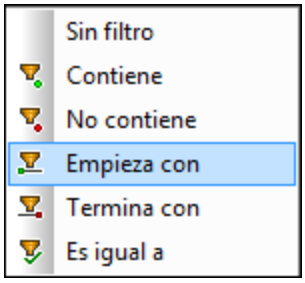


El panel *Origen* también incluye una barra de herramientas (*imagen siguiente*) con tres iconos que son, de izquierda a derecha: **Filtrar contenido de carpeta**, **Mostrar objetos seleccionados** y **Localizador de objetos**. Haga clic en el icono **Mostrar objetos seleccionados** para ver solamente las tablas seleccionadas. Haga clic otra vez para ver todas las tablas.

Filtrar el contenido de las carpetas

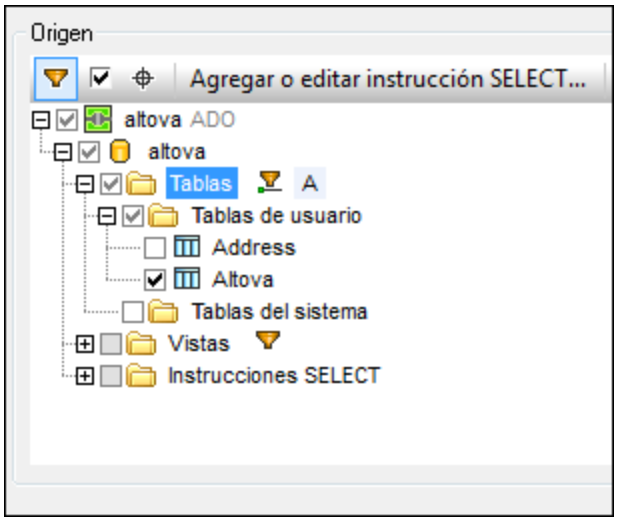
Siga estos pasos para filtrar los objetos del panel *Origen*:

1. Haga clic en el icono **Filtrar contenido de carpeta** de la barra de herramientas del panel *Origen*. Aparece un icono en forma de embudo junto a la carpeta `Tablas`.
2. Haga clic en el icono en forma de embudo y seleccione un filtro en el menú contextual que aparece (*imagen siguiente*). Por ejemplo, **Empieza con**.



3. En el campo de entrada que aparece escriba la cadena de texto que desea usar para filtrar el contenido de las carpetas. Por ejemplo, escriba `A` para filtrar el contenido de la carpeta `Tablas`. El filtro se aplica a medida que se teclea.

 © 2016-2022 Altova GmbH



Localizador de objetos

Si quiere buscar un elemento concreto de la BD por su nombre, utilice el localizador de objetos del panel *Origen*. Así funciona:

1. Haga clic en el icono **Localizador de objetos** de la barra de herramientas del panel *Origen*. Observe que aparece un cuadro combinado al final del panel *Origen*.
2. Escriba la cadena de búsqueda en el campo de entrada, por ejemplo `Altova` (*imagen siguiente*). Haga clic en la punta de flecha para abrir la lista desplegable. Esta lista incluye todos los objetos cuyo nombre contiene la cadena de búsqueda.

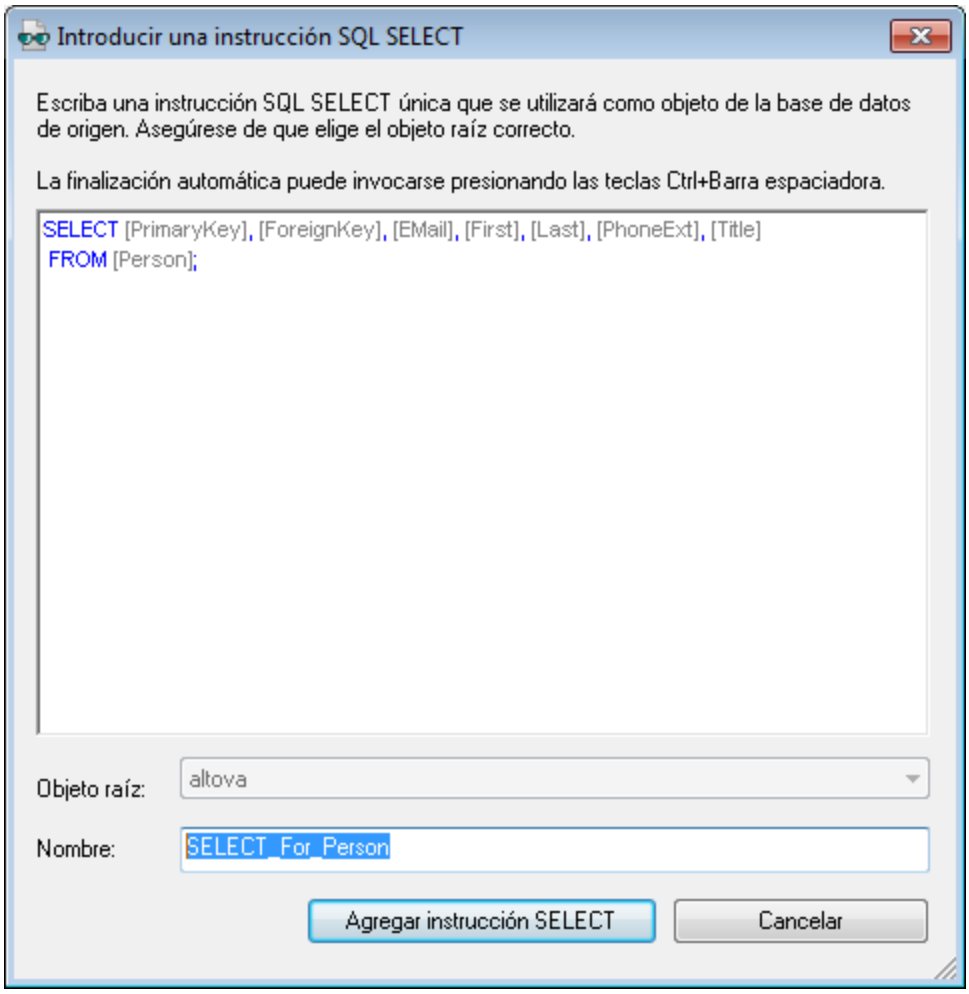
3. Haga clic en un objeto de la lista para verlo en el panel *Origen*.

Agregar y editar instrucciones SELECT para vistas locales

Puede crear instrucciones `SELECT` en SQL para crear vistas locales. Cuando el esquema se genera a partir de una conexión de BD que tiene vistas locales (o instrucciones `SELECT`) definidas, el esquema que se genera para la BD contendrá una tabla por cada instrucción `SELECT`.

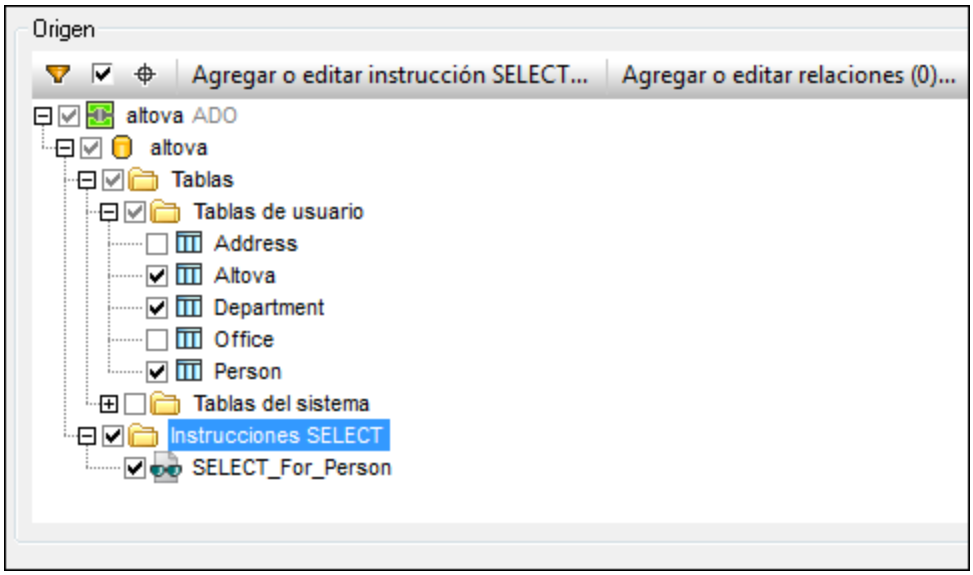
Siga estos pasos para crear una instrucción `SELECT`:

1. Haga clic en el botón **Agregar o editar instrucción SELECT** para abrir el cuadro de diálogo "Introducir una instrucción SQL SELECT" (*imagen siguiente*).



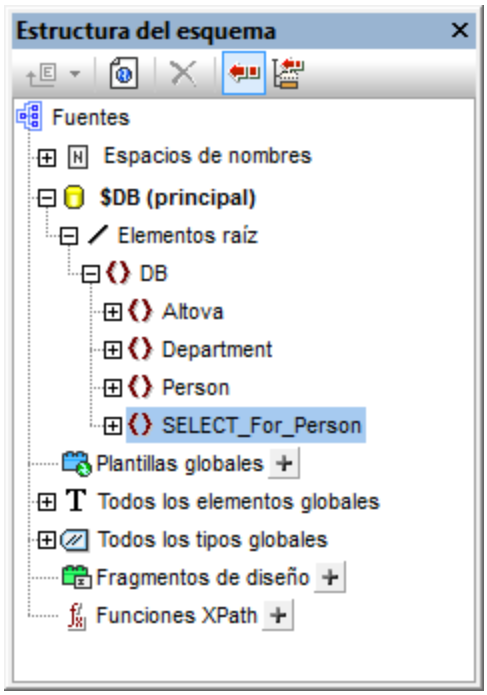
2. Escriba la instrucción `SELECT` en el panel principal. Si se conecta a una BD Oracle o IBM DB2 con JDBC, entonces la instrucción `SELECT` no puede tener un punto y coma al final. Si desea crear una instrucción `SELECT` para toda la tabla, haga clic con el botón derecho en la tabla en el cuadro de diálogo "Insertar objetos de la base de datos" y seleccione el comando **Generar y agregar una instrucción SELECT** del menú contextual.
3. En el cuadro de diálogo "Introducir una instrucción SQL SELECT" haga clic en el botón **Agregar instrucción SELECT...** y la instrucción se añade a la lista de instrucciones SELECT que aparecen al final del panel *Origen* del cuadro de diálogo "Insertar objetos de la base de datos" (*imagen siguiente*).

 © 2016-2022 Altova GmbH



Nota: si se conecta a una BD Oracle o IBM DB2 con JDBC y usa una instrucción `SELECT` con el comando Agregar/Quitar tabla para recuperar datos, entonces la instrucción `SELECT` no puede tener un punto y coma al final.

Al hacer clic en el botón **Finalizar** del cuadro de diálogo "Insertar objetos de la base de datos" StyleVision crea una tabla por cada instrucción `SELECT` (*imagen siguiente*).

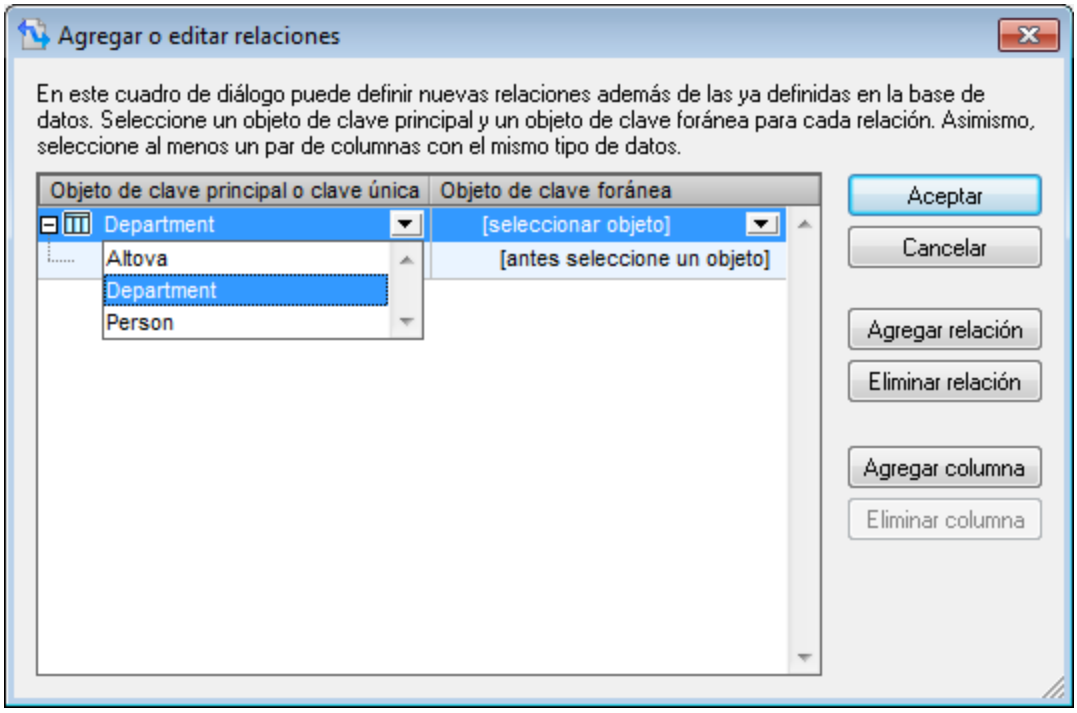


Relaciones locales entre tablas

Puede crear relaciones locales entre dos tablas, similares a las relaciones de tipo clave principal / clave foránea. La relación es local en StyleVision. Esto significa que la BD propiamente dicha no debe modificarse. La relación local se representará en el esquema que StyleVision genera.

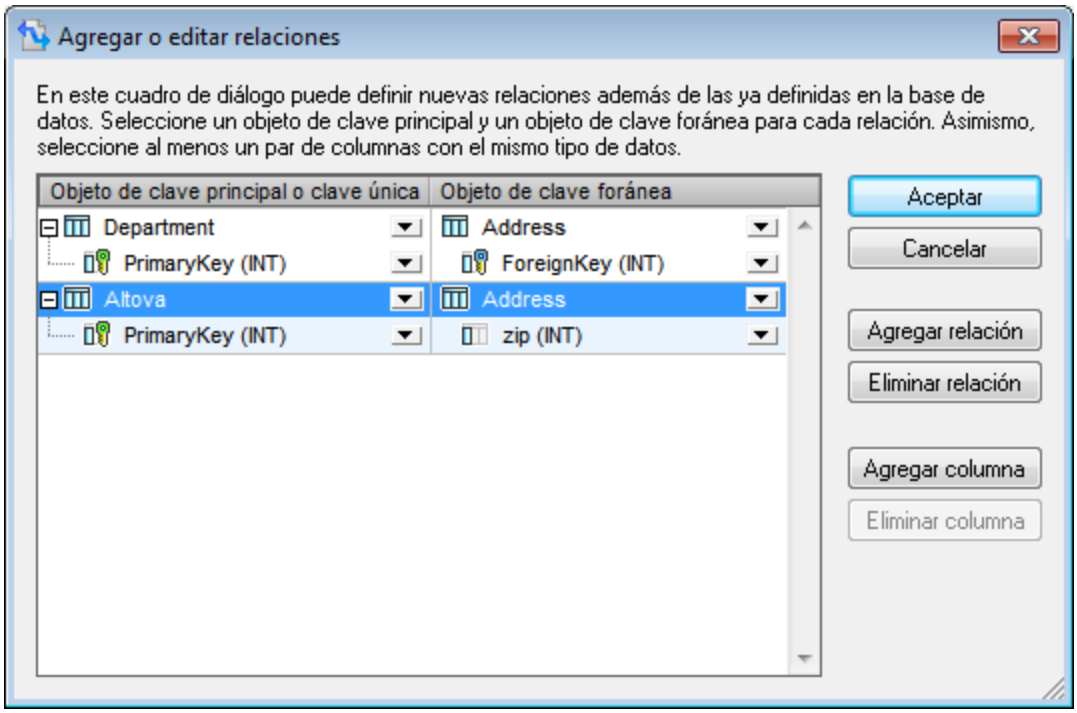
Siga estos pasos para crear una relación local:

1. En el cuadro de diálogo "Insertar objetos de la base de datos" haga clic en el botón **Agregar o editar relaciones**, que abre otro cuadro de diálogo llamado "Agregar o editar relaciones" (*imagen siguiente*).
2. Haga clic en el botón **Agregar relación** de este cuadro de diálogo. En la columna *Objeto de clave principal o clave única* del panel principal haga clic en la flecha para abrir la lista desplegable del cuadro combinado (*imagen siguiente*). Ahora seleccione una tabla para la columna *Objeto de clave foránea.* La relación que se genera seleccionará filas donde la columna *Clave principal/única* seleccionada coincide con la columna *Clave foránea* seleccionada.

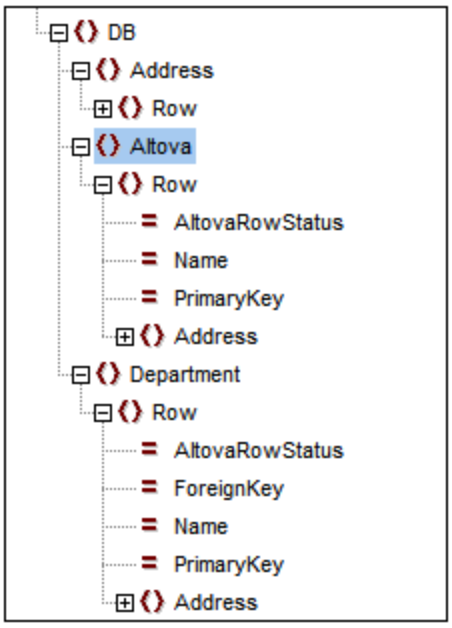


3. Seleccione la columna *Clave principal/única* que debe coincidir con la columna *Clave foránea* y después seleccione la columna *Clave foránea*. Vuelve a usar los cuadros combinados de las respectivas columnas (*imagen siguiente*). Recuerde que si los tipos no coinciden aparece un mensaje de error.
4. Añada todas las relaciones locales que necesite repitiendo los pasos 2 y 3.

 © 2016-2022 Altova GmbH



5. Cuando termine, haga clic en **Aceptar**. El esquema generado reflejará las relaciones recién creadas.

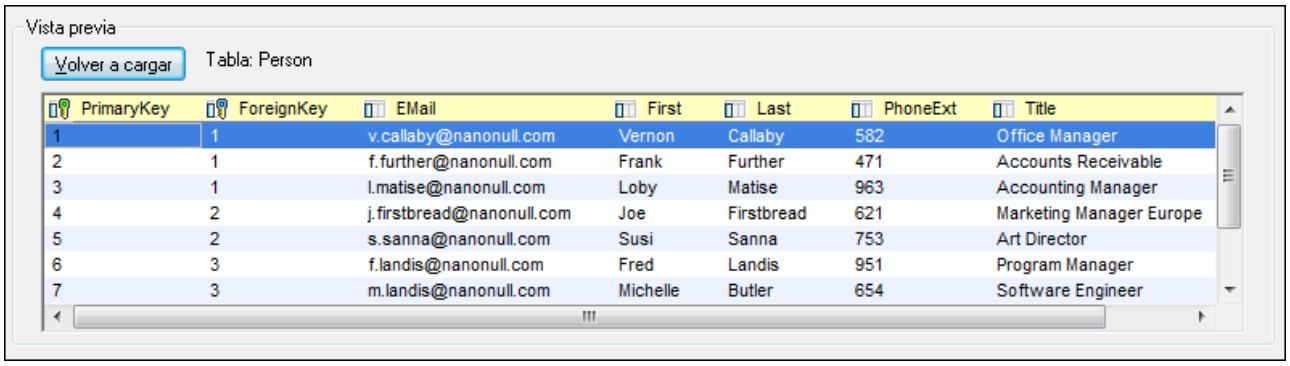


Observe que en las dos imágenes anteriores, en el esquema generado tanto `Altova` como `Department` contienen `Address`. Es decir, se generarán las filas de `Department` con valores `PrimaryKey` iguales al valor `ForeignKey` de la fila `Address`. Y se generarán las filas de `Altova` con valores `PrimaryKey` iguales al valor `zip` de la fila `Address`.

Nota: no son compatibles las relaciones locales entre instrucciones SQL `SELECT`.

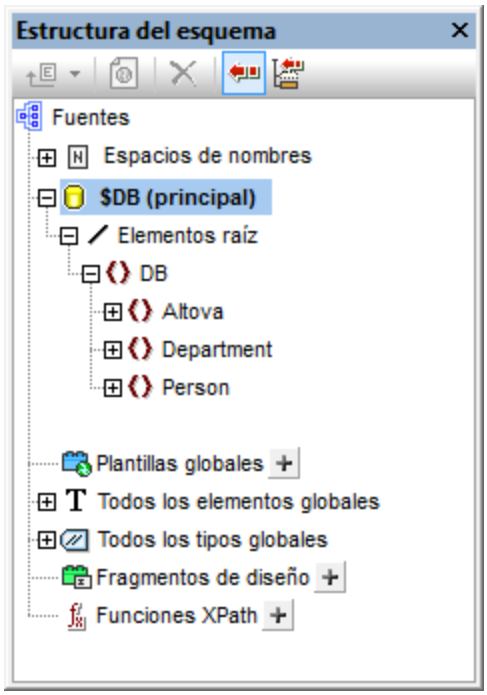
Panel Vista previa

Para obtener la vista previa de la estructura y del contenido de una tabla, seleccione la tabla en el panel *Origen* y después haga clic en el botón **Vista previa** (o **Volver a cargar,** si ya está abierta una vista previa) del panel inferior *Vista previa* (*imagen siguiente*). Al hacer clic en el botón aparece el contenido de la tabla en forma de tabla en el panel de vista previa (*imagen siguiente*).



Generar el esquema XML y el archivo XML de trabajo a partir de la BD

Cuando termine de seleccionar las tablas que desea usar en el diseño SPS, haga clic en el botón **Finalizar** para generar y cargar el esquema XML. StyleVision muestra en la ventana Estructura del esquema un esquema XML cuya estructura se corresponde con la de la BD de las tablas seleccionadas. StyleVision también genera un archivo XML de trabajo cuya estructura se corresponde a la definida en el esquema generado y cuyo contenido procede de las tablas seleccionadas. Este archivo XML de trabajo se usa para la vista previa de los documentos de salida.



© 2016-2022 Altova GmbH

Recuerde que todas las tablas seleccionadas de la BD se crean en el esquema XML como secundarios del elemento de documento `DB` y como elementos en la lista **`Elementos globales`** de la ventana Estructura del esquema. Para más información sobre la estructura del esquema XML generado, consulte el apartado Esquema XML y archivos XML de la BD 620. No olvide que el esquema XML generado a partir de la BD no se modificará con ningún filtro construido a partir de ahora.

Cuando termine de conectarse a la BD y de generar el esquema XML podrá usar todas las funciones de diseño de StyleVision para diseñar el documento SPS.

10.3.2 Bases de datos XML

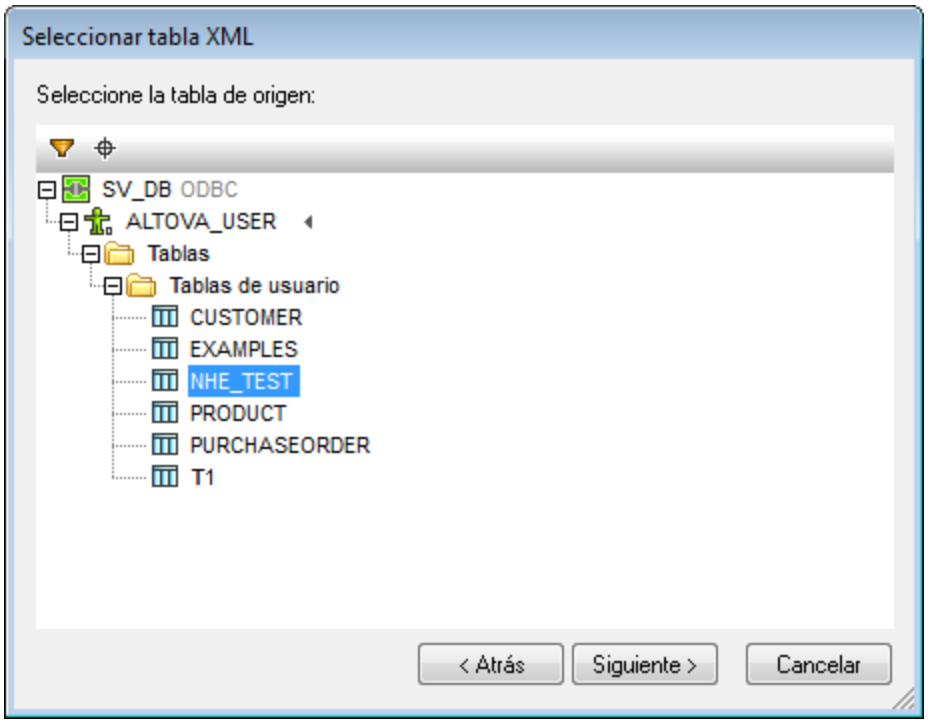
Tras establecer la conexión con la BD XML (por ahora solo son compatibles las BD XML IBM DB2) desde el cuadro de diálogo "Abrir la base de datos" (con el comando **Archivo | Nuevo | Nuevo a partir de una columna XML de una tabla de la base de datos**), el siguiente paso es:

- Seleccionar la celda de la BD que contiene el documento XML necesario. El documento XML se cargará automáticamente como archivo XML de trabajo.
- Seleccionar el esquema XML para el diseño SPS.

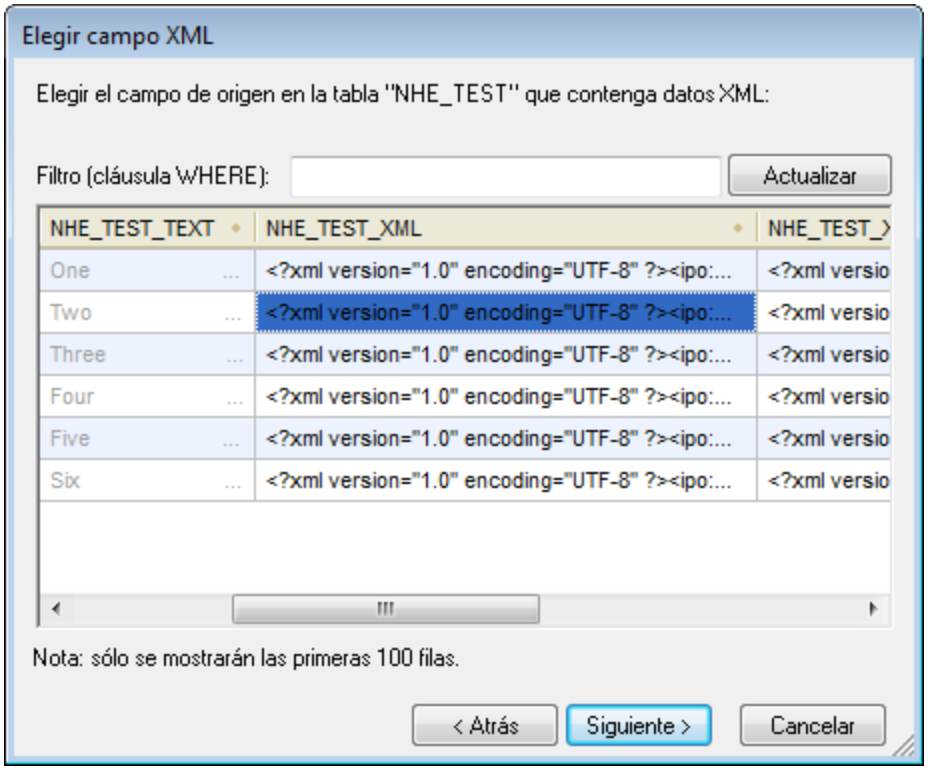
Seleccionar la celda XML y el archivo XML de trabajo

Cuando se establece la conexión con la base de datos IBM DB2, aparece el cuadro de diálogo "Seleccionar tabla XML" (*imagen siguiente*).

1. En el cuadro de diálogo "Seleccionar tabla XML" seleccione la tabla que contiene los datos XML que desea usar para crear el archivo XML de trabajo. Por ejemplo, en la imagen siguiente está seleccionada la tabla `NHE_TEST`.



2. Haga clic en **Siguiente**. Se abre el cuadro de diálogo "Elegir campo XML" (*imagen siguiente*). Si quiere filtrar la selección que aparece en el panel, escriba una cláusula SQL `WHERE` y haga clic en **Actualizar**. Recuerde que la cláusula `WHERE` debe ser la condición solamente (es decir, sin la palabra clave `WHERE`. P. ej.: `NHE_TEST_TEXT= 'Two'`)



3. Seleccione la celda que contiene los datos XML que desea usar para crear el archivo XML de trabajo. Por ejemplo, en la imagen anterior aparece resaltada en azul la celda seleccionada.
4. Haga clic en **Siguiente**. Se abre otro cuadro de diálogo llamado "Elegir esquema XML" donde puede seleccionar el esquema XML que se debe usar con el SPS (*ver más abajo*).

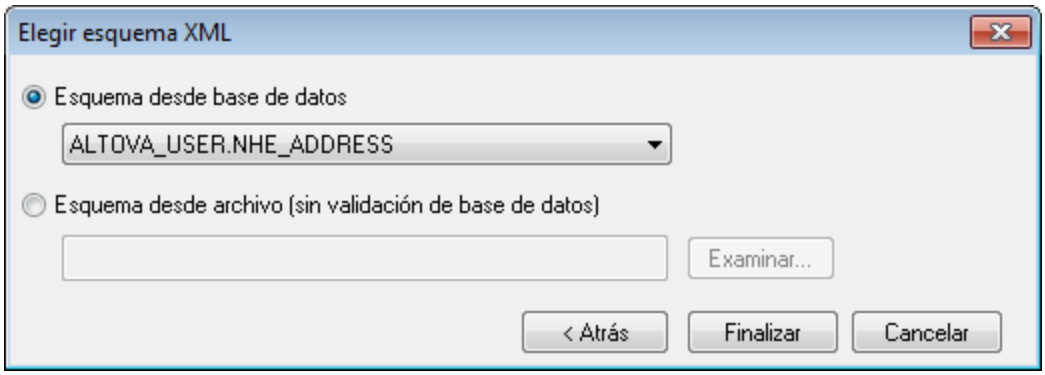
Nota: si se conecta a una BD Oracle o IBM DB2 con JDBC y usa una instrucción `SELECT` con el comando Agregar/Quitar tabla para recuperar datos, entonces la instrucción `SELECT` no puede tener un punto y coma al final.

Seleccionar el esquema XML para el diseño SPS

El esquema que se usará para el diseño SPS puede ser un esquema XML ubicado en la BD o un esquema situado en cualquier otra ubicación a la que pueda acceder StyleVision. Siga estas instrucciones para seleccionar el esquema:

1. En el cuadro de diálogo "Elegir esquema XML" (*imagen siguiente*) seleccione uno de los dos botones de opción, dependiendo de si quiere usar un esquema almacenado en la BD o en el sistema de archivos. Tenga en cuenta que si selecciona un esquema del sistema de archivos no se realizará una validación de la BD.

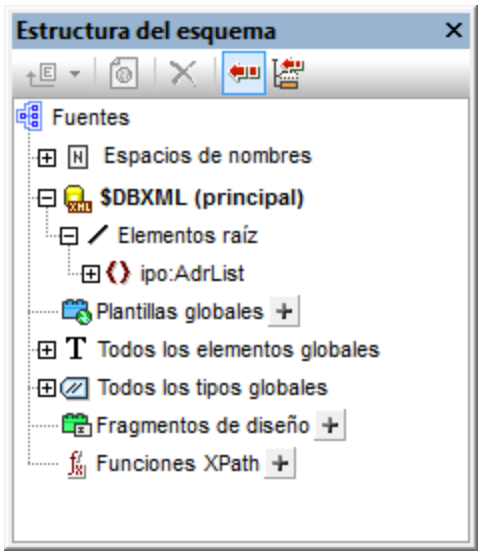
 © 2016-2022 Altova GmbH



2. Seleccione el esquema. Los esquemas almacenados en la BD aparecen en la lista desplegable del cuadro combinado *Esquema desde base de datos*. Para seleccionar un esquema externo haga clic en el botón **Examinar**.
3. Cuando termine haga clic en **Finalizar**.

La estructura del esquema

Una vez completado el proceso de selección de datos XML y del esquema XML, StyleVision genera un archivo XML de trabajo con los datos XML seleccionados y carga el esquema en el diseño SPS. Ambos documentos aparecen en la ventana Estructura del esquema (*imagen siguiente*).



Ahora puede diseñar el documento SPS con ayuda de los mecanismos de diseño de StyleVision. Recuerde que los datos del archivo XML de trabajo se pueden editar en la vista Authentic y guardarse en la BD.

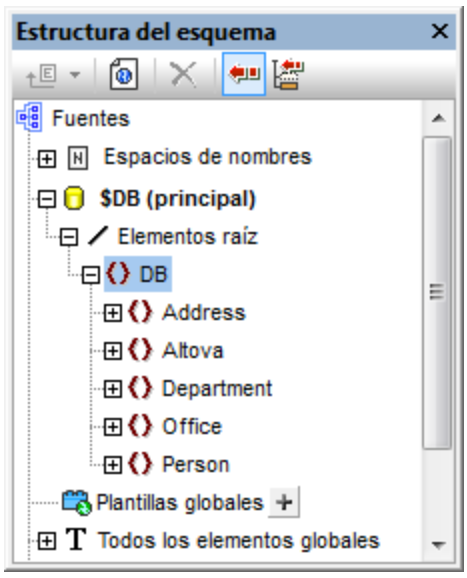
Nota: el archivo XML de trabajo debe ser válido con respecto al esquema seleccionado para el diseño SPS. Compruebe también que el elemento raíz (elemento de documento) 221 del esquema se corresponde con el elemento raíz del documento XML.

10.4 Esquema XML y archivos XML de la BD

El esquema XML de la BD

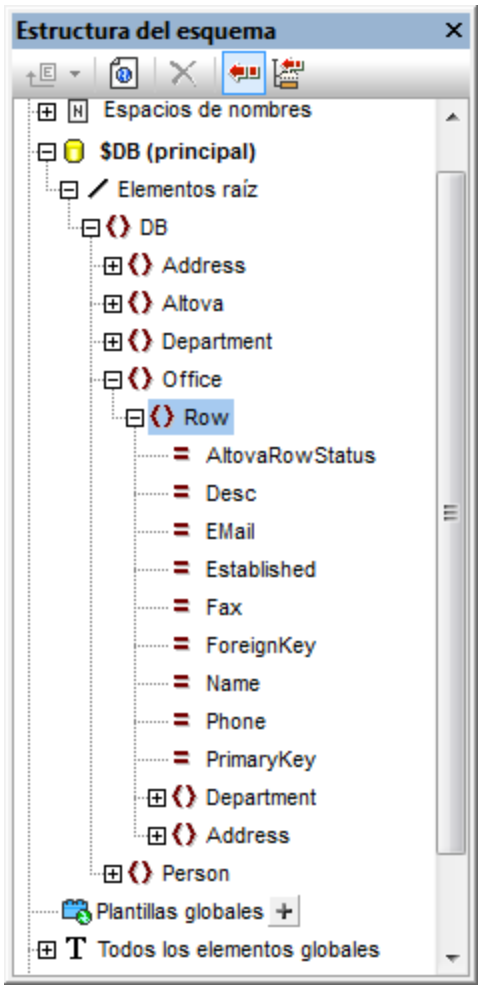
Cuando se carga una BD no XML, StyleVision genera un esquema XML cuya estructura se basa en la estructura de la BD y el esquema aparece en la ventana Estructura del esquema. Cuando se carga una BD XML, el esquema XML ya existe (bien dentro de la BD, bien en el sistema de archivos). Este apartado, por tanto, se centra en los esquemas que StyleVision genera para BD no XML.

El esquema XML se crea con un elemento de documento llamado `DB`. Este elemento `DB` contiene elementos secundarios que se corresponden con todas las tablas de nivel superior de la BD. Estos elementos de tabla de nivel superior también aparecen en la lista **`Elementos globales`** de la ventana Estructura del esquema. Por ejemplo, en la imagen siguiente, los elementos de nivel superior son: `Address`, `Altova`, `Department`, `Office` y `Person` y se corresponden con las tablas de la BD.



Cada elemento de tabla de nivel superior puede tener un número ilimitado de filas (*imagen siguiente*). Cada fila se corresponde con un registro de la BD. En la estructura del esquema cada fila se representa con un elemento `Row`. Cada elemento `Row` tiene atributos que equivalen a los campos de la tabla. Por cada fila de cada tabla, StyleVision genera el atributo `AltovaRowStatus`, que almacena el estado actual de la fila: agregada, actualizada o eliminada. Los demás atributos equivalen a campos de la tabla en la BD.

© 2016-2022 Altova GmbH



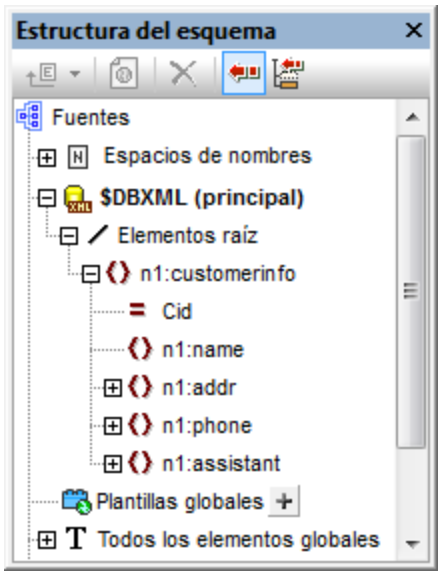
Nota: en la imagen anterior puede ver un ejemplo de la estructura del esquema XML generado. Las tablas seleccionadas durante el proceso de conexión se incluyen en la estructura. Los filtros de BD utilizados durante el proceso de conexión no afectan a la estructura del esquema XML.

Nueva estructura de los esquemas de BD

La estructura del esquema XML generado por StyleVision a partir de una BD cambió a partir de la versión 2005 de StyleVision. Esta nueva estructura permite editar las BD en la vista Authentic de los productos de Altova, característica no disponible en las versiones anteriores de StyleVision. Como consecuencia, los SPS generados con versiones anteriores a la versión 2005 de StyleVision darán error al abrirse en la versión 2005 o en una versión posterior. Para poder usar la característica de edición de BD de StyleVision, vuelva a crear el SPS en la versión actual de StyleVision.

Archivos de datos XML de la BD

Una vez establecida la conexión con la BD XML y seleccionados el esquema XML y la columna de datos XML, la ventana Estructura del esquema (*imagen siguiente*) muestra el esquema seleccionado y la columna que se utilizará para crear el archivo XML de trabajo.



A partir de la BD se generan dos archivos XML temporales (consulte el apartado Bases de datos en StyleVision 521 para ver un diagrama):

- Un archivo XML temporal editable, que se puede editar en la vista Authentic
- Un archivo XML temporal no editable, que se usa como archivo XML de trabajo (para las vistas previas y la generación de archivos de salida)

El **archivo XML editable** temporal se genera cuando se carga la BD en StyleVision. Se puede editar en la vista Authentic una vez creado el diseño SPS. Los datos que se presentan en la vista Authentic se pueden filtrar con el mecanismo de consulta de la vista. Los cambios realizados en la vista Authentic en los datos editables se escriben en el archivo XML temporal. Para guardar en la BD los datos modificados del archivo XML editable haga clic en el comando de menú **Archivo | Guardar datos XML de Authentic**.

El **archivo XML no editable** temporal se genera cuando se carga la BD en StyleVision y se usa como archivo XML de trabajo para la vista previa y para generar archivos de salida en formato HTML y RTF.

Nota:

- En la vista Authentic de otros productos de Altova solamente se crea un archivo XML temporal (editable) cuando se abre un diseño SPS basado en una BD. Los cambios realizados en la vista Authentic se escriben en este archivo. Cuando se guarda un archivo, la información del archivo XML se escribe en la BD.
- Puede filtrar los datos que se pasan al archivo XML temporal no editable (consulte el apartado Filtros de BD 861 para obtener más información.)
- No es necesario asignar un archivo XML de trabajo concreto para poder obtener una vista previa en HTML y RTF. El archivo XML temporal (no editable) generado automáticamente se usa para estas vistas previas.

 © 2016-2022 Altova GmbH

10.5 Filtros de BD: filtrar datos de la BD

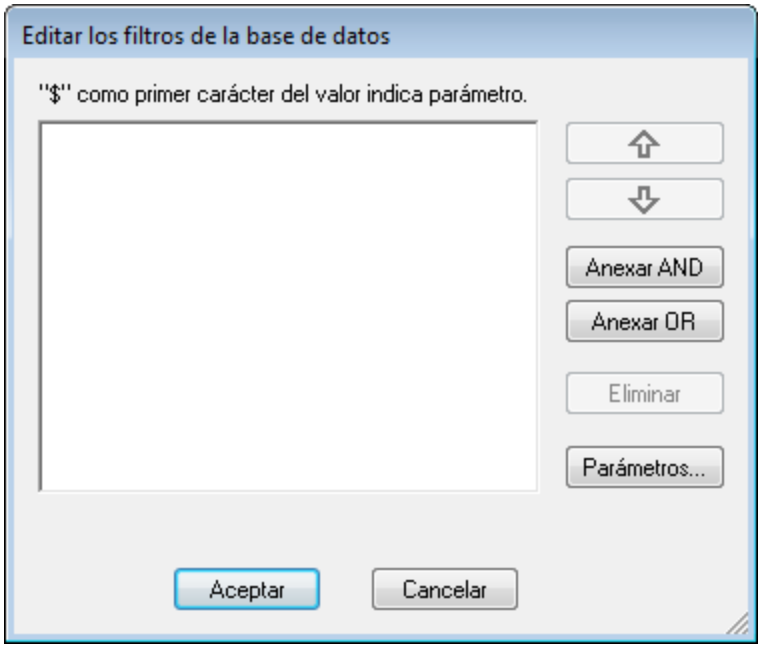
En StyleVision puede filtrar qué datos de la BD se importan en el archivo XML **no editable** temporal. Recuerde que el archivo XML no editable se utiliza para generar informes y que el efecto del filtro de BD solamente se observa en la vista previa HTML y RTF y no en la vista Authentic (que solo muestra el **archivo XML editable** temporal).

El filtro de BD se puede crear dentro de la BD misma (si su aplicación de BD ofrece esta función) o dentro del archivo SPS. En el archivo SPS puede crear un filtro de BD por cada tabla de nivel superior del esquema XML (es decir, por cada tabla de datos que sea un elemento secundario del elemento `DB`). Cada vez que se crea o modifica un filtro de BD, los datos de la BD se vuelven a cargar en el archivo XML no editable temporal que StyleVision genera para la BD. Por consiguiente, los filtros de la BD sirven para crear archivos XML de tamaño óptimo y así generar documentos de salida con mayor eficacia.

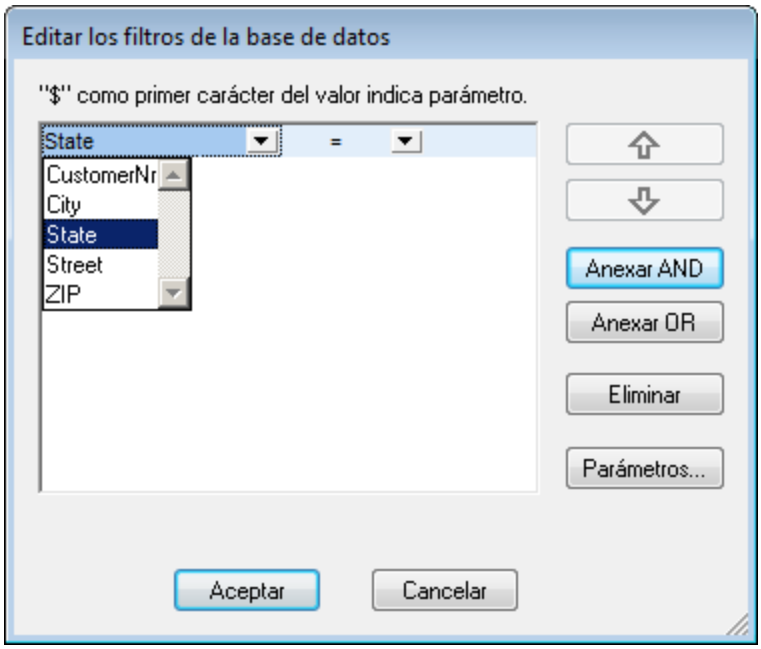
Nota: los filtros de BD modifican los datos que se importan en el archivo XML no editable temporal. Si guarda un diseño SPS con un filtro de BD y después genera un archivo XML a partir del diseño SPS, el contenido del archivo XML generado estará filtrado de acuerdo con los criterios del filtro de BD.

Crear un filtro de BD

1. En el documento de diseño o en la ventana Estructura del esquema seleccione el elemento de tabla para el que desea crear el filtro de BD (clic en la etiqueta inicial o final del elemento o clic en el elemento en la Estructura del esquema).
2. Seleccione el comando **Base de datos | Editar filtro de la BD** 861 o haga clic en el icono de la barra de herramientas. Se abre este cuadro de diálogo:



3. Haga clic en el botón **Anexar AND** o **Anexar OR**. Se anexa un criterio vacío para el filtro.



4. Escriba la expresión para el criterio. Una expresión se compone de (i) un nombre de campo (a elegir en el cuadro combinado); (ii) un operador (a elegir en el cuadro combinado); y (iii) un valor (que se debe introducir a mano). Siga leyendo para obtener más información sobre cómo construir expresiones 624.
5. Si quiere añadir otro criterio, haga clic en el botón **Anexar AND** o **Anexar OR**, dependiendo del operador lógico que desee usar para unir los dos criterios (AND u OR). Después añada un criterio nuevo. Para más información sobre operadores lógicos, consulte el apartado Reordenar criterios de los filtros de BD 626.

Expresiones de los criterios del filtro

Las expresiones de los criterios de filtro de BD se componen de un nombre de campo, un operador y un valor. Los **nombres de campo disponibles** son las columnas de la tabla de datos de nivel superior seleccionada. Los nombres de campo aparecen en la lista desplegable de un cuadro combinado (*imagen anterior*). Los **operadores disponibles** son estos:

```
=             Igual a
<>            No igual a
<             Menor que
<=            Menor o igual que
>             Mayor que
>=            Mayor o igual que
LIKE          Se parecen fonéticamente
NOT LIKE      No se parecen fonéticamente
IS NULL       Está vacío
NOT NULL      No está vacío
```

Si selecciona `IS NULL` o `NOT NULL`, se deshabilita el campo *Valor*. Los **valores** deben introducirse sin comillas (y sin delimitador alguno). Los valores también deben tener el mismo formato que los del campo de

BD correspondiente. De lo contrario, la expresión dará `FALSE` como resultado. Por ejemplo, si un criterio para un campo de tipo `date` de una BD MS Access tiene la expresión `StartDate=25/05/2004`, la expresión dará `FALSE` como resultado porque el tipo de datos `date` de las BD MS Access tiene el formato `AAAA-MM-DD`.

Usar parámetros con filtros de BD

También puede introducir el nombre de un parámetro como valor de una expresión. En este caso se llama al parámetro y su valor se usa como valor de la expresión. El parámetro puede estar declarado ya en la hoja de estilos o puede declararse más tarde. Sin embargo, tenga en cuenta que, si escribe un parámetro que no está declarado todavía, el botón **Aceptar** se deshabilita.

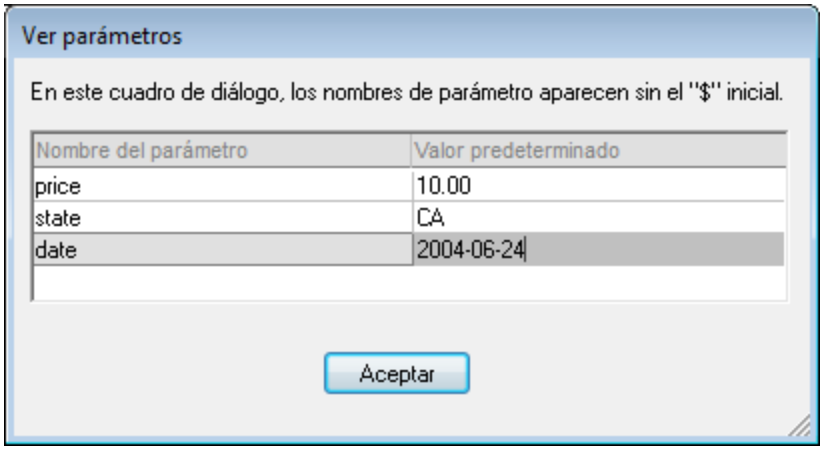
Los parámetros son prácticos si desea usar un solo valor en varias expresiones o si quiere pasar un valor a un parámetro desde StyleVision Server.

Para escribir el nombre de un parámetro como valor de una expresión, escriba `$` en el campo *Valor* seguido del nombre del parámetro (sin espacios). Si el parámetro ya está declarado (ver el apartado Parámetros 327), la entrada aparece en color verde. Si el parámetro no está declarado todavía, la entrada aparece en rojo y debe declarar el parámetro.

Declarar parámetros desde el cuadro de diálogo "Editar los filtros de la BD"

Para declarar un parámetro desde el cuadro de diálogo "Editar los filtros de la BD":

1. Haga clic en el botón **Parámetros...** del cuadro de diálogo. Esto abre un diálogo nuevo llamado "Ver parámetros".



2. Escriba el nombre y el valor del parámetro y haga clic en **Aceptar**.

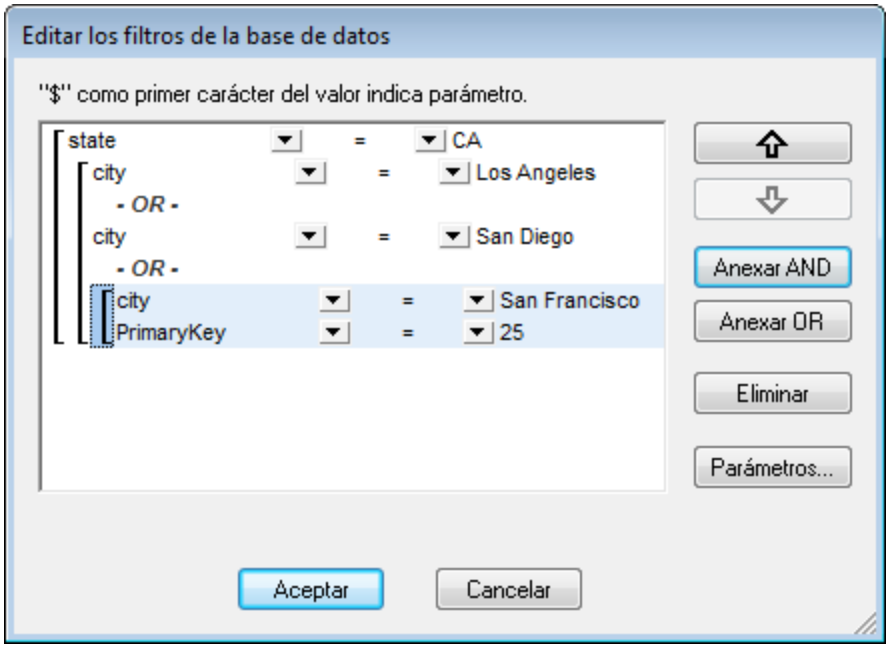
También puede seleccionar el comando de menú **Edición | Parámetros de la hoja de estilos**, que abre el cuadro de diálogo "Editar parámetros". En este cuadro de diálogo también puede declarar parámetros para sus filtros de BD.

Nota: el cuadro de diálogo "Editar parámetros" contiene todos los parámetros definidos para la hoja de estilos. Aunque es un error usar un parámetro no declarado en el diseño SPS, declarar un parámetro y no usarlo no es un error.

Una vez creado el filtro de BD para un elemento de tabla, el elemento aparece en la Estructura del esquema con un icono en forma de embudo. Por ejemplo, el elemento `Addresses` de la imagen siguiente tiene definido un filtro de BD.

Reordenar criterios en los filtros de BD

La estructura lógica del filtro de BD y la relación entre parejas de criterios o grupos de criterios se indica de forma gráfica en el diálogo "Editar los filtros de la base de datos". Cada nivel de la estructura lógica se señala con un gran corchete. Si dos criterios o dos grupos de criterios son adyacentes, significa que el operador que los une es AND. Si los dos criterios o grupos de criterios están separados por – *OR* – entonces se trata del operador OR. En la representación gráfica también se aplica sangría a los criterios para ofrecer una visión de conjunto clara de la estructura lógica del filtro de BD.



El filtro de BD de la imagen anterior se podría representar con este texto:

```
State=CA AND (City=Los Angeles OR City=San Diego OR (City=San Francisco AND
CustomerNr=25))
```

Puede reordenar el filtro de BD subiendo y bajando criterios o grupos de criterios. Siga estos pasos para reordenar un criterio o grupo de criterios:

1. Haga clic en el criterio para seleccionarlo o haga clic en un corchete para seleccionar todo el nivel.
2. Haga clic en el botón en forma de flecha **Subir** o **Bajar** del cuadro de diálogo.

Debe tener en cuenta estos aspectos:

- Si el criterio adyacente en la dirección en la que se mueve el criterio está al mismo nivel, los dos criterios intercambian puestos.

 © 2016-2022 Altova GmbH

- Los grupos de criterios (es decir, criterios comprendidos en un corchete) cambian de posición dentro del mismo nivel (no cambian de nivel).
- Los criterios cambian de posición dentro del mismo nivel. Si el criterio adyacente está en un nivel superior o inferior (es decir, no está en el mismo nivel), el criterio seleccionado **sube o baja un nivel**.

Si quiere eliminar un criterio del filtro de BD, selecciónelo y haga clic en el botón **Eliminar**.

Modificar un filtro de BD

Para modificar un filtro de BD haga clic en el comando **Base de datos | Editar filtro de la base de datos** 861. Esto abre el cuadro de diálogo "Editar los filtros de la base de datos". En este diálogo puede editar las expresiones de los criterios enumerados, agregar criterios nuevos, reordenar los criterios o eliminar criterios del filtro. Cuando termine de modificar el filtro haga clic en **Aceptar**. Los datos de la BD se vuelven a cargar automáticamente en StyleVision para reflejar los cambios realizados en el filtro.

Borrar (eliminar) un filtro de BD

Para borrar (eliminar) un filtro de BD, seleccione el elemento cuyo filtro de BD desea borrar, bien en la ventana Estructura del diseño, bien en la vista **Diseño**. (Recuerde que hay un filtro de BD por cada elemento de tabla de nivel superior.) Haga clic en el comando **Base de datos | Borrar filtro de la base de datos** 861**.** El filtro se elimina y el elemento ya no aparece con un icono en forma de embudo en la ventana Estructura del esquema.

10.6 Características de diseño para BD

Los diseños SPS basados en BD se diseñan igual que los SPS basados en esquemas, es decir, arrastrando nodos de esquema desde la ventana Estructura del esquema y colocándolos en la vista **Diseño**, insertando contenido estático directamente en el documento de diseño y aplicando formato a los componentes de diseño. Sin embargo, los SPS basados en BD tienen algunas características propias.

Crear una tabla dinámica para una tabla de BD

Si quiere crear una tabla dinámica para una tabla de BD, siga estos pasos:

1. En la ventana Estructura del esquema seleccione la tabla de BD de nivel superior y arrástrela hasta la vista **Diseño**.
2. Al soltarla, seleccione **Contenido** en el menú contextual que aparece y después seleccione el marcador de posición `(contenido)` que se crea en el diseño. Si en ese momento está activada la característica **Agregar controles de BD automáticamente**, el diseño tendrá este aspecto:

3. En la ventana Estructura del esquema seleccione el elemento `Row` de la tabla de BD que quiere usar para crear la tabla dinámica.
4. Arrástrela dentro del elemento `Addresses`.
5. Al soltar el elemento `Row` dentro del elemento `Addresses`, seleccione **Crear tabla** en el menú contextual y elija qué campos de la BD se usan para crear las columnas de la tabla dinámica. Se crea una tabla dinámica a partir de la tabla de BD.

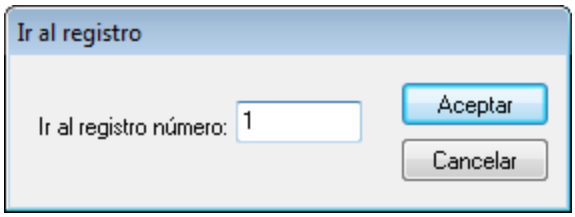
Nota: también puede crear otros componentes a partir de la tabla de BD. Por ejemplo puede crear `(contenido)`.

Agregar controles de BD automáticamente

El comando de menú **Authentic | Agregar controles de BD automáticamente** o el icono de la barra de herramientas activa/desactiva la inserción automática de controles de navegación para tablas de BD. Si el comando está activado, el icono de la barra de herramientas tiene un contorno negro. Si está desactivado, el icono no tiene contorno. Si este comando está activado, cuando coloque una tabla de BD en el diseño, se insertarán automáticamente controles de BD (*imagen siguiente*). Estos controles se insertan por defecto detrás de la etiqueta de inicio de la tabla.

Estos controles permiten al usuario de la vista Authentic navegar por los registros de la tabla de BD en la vista Authentic. El primer botón (al final a la izquierda) le lleva al primer registro de la tabla. El segundo le lleva al registro anterior al registro actual. El tercero es el botón **Ir a**, que abre el cuadro de diálogo "Ir al registro" (*imagen siguiente*) donde puede escribir el número de registro al que desea ir. El cuarto botón le lleva al registro posterior al registro actual y el último botón le lleva al último registro de la tabla.

© 2016-2022 Altova GmbH



Si quiere insertar controles de navegación a mano (p. ej. si quiere insertar los controles en una posición distinta a la posición predeterminada), siga estos pasos:

1. Desactive el icono **Agregar controles de BD automáticamente**.
2. Ponga el cursor en la posición donde desea que aparezcan los controles de navegación (en cualquier posición entre la etiqueta inicial y fina del elemento de tabla de BD).
3. Haga clic con el botón derecho y seleccione **Insertar control de base de datos | Navegación** o **Insertar control de base de datos | Navegación+Ir a** en el menú contextual. También puede acceder a estos comandos desde el menú **Insertar**.

Insertar un botón de consulta de BD para la vista Authentic

El botón de consulta de BD permite al usuario de la vista Authentic realizar consultas en la BD y así crear condiciones para los registros que desea ver en la vista Authentic. Los botones de consulta se pueden insertar en cualquier posición de una tabla de BD, siempre que sea entre:

- La etiqueta inicial del elemento de tabla de BD y la etiqueta inicial del elemento `Row` (secundario) del elemento de tabla de BD.
- La etiqueta final del elemento de tabla de BD y la etiqueta final del elemento `Row` (secundario) del elemento de tabla de BD.

Siga estos pasos para insertar un botón de consulta de BD en el diseño:

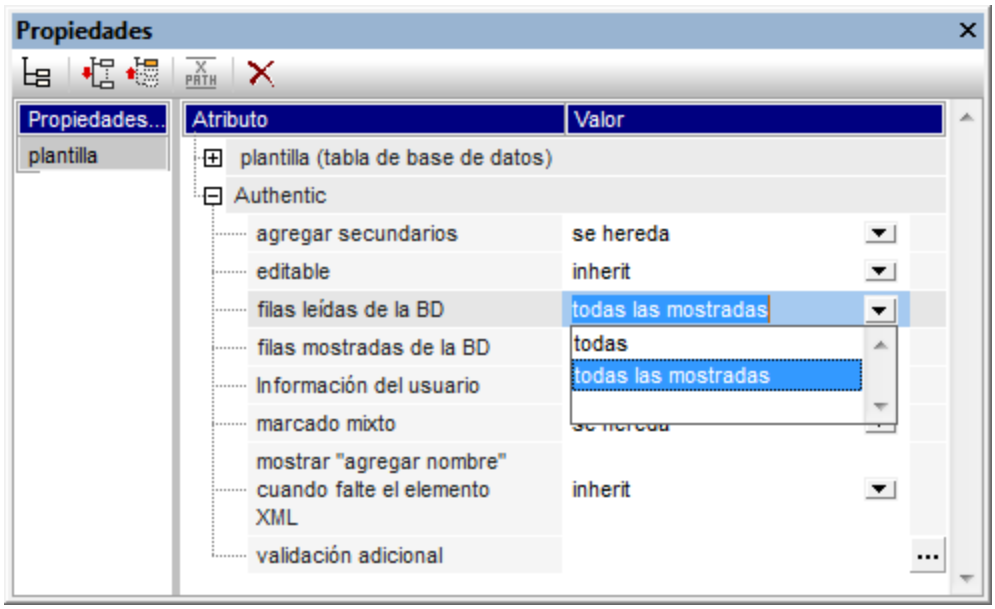
1. Ponga el cursor en la posición correcta (ver punto anterior) donde desea que aparezca el botón de consulta.
2. Haga clic con el botón derecho y en el menú contextual (en el menú **Insertar**) seleccione **Insertar control de base de datos | Botón de consulta**. El botón de consulta se inserta en esa posición del documento de diseño.

Cuando el usuario de la vista Authentic hace clic en el botón de consulta, aparece el cuadro de diseño "Editar la consulta de base de datos". Este cuadro de diálogo se describe en la documentación de la vista Authentic 675.

Registros mostrados y registros recuperados

También puede controlar cuántos registros se muestran en la vista Authentic y cuántos registros se recuperan al cargarse el archivo. Para definir estos valores:

1. En la vista **Diseño** seleccione el elemento `Row` que corresponde a los registros que deben mostrarse/recuperarse.
2. En la ventana Propiedades seleccione el componente **`plantilla`** de la columna *Propiedades para* y expanda el grupo de propiedades *Authentic*.



3. Para la propiedad `filas mostradas de la BD`, seleccione el valor `todas` en la lista desplegable o escriba el número de registros que se deben mostrar.
4. Para la propiedad `filas leídas de la BD`, seleccione el valor `todas` o `todas las mostradas` en la lista desplegable o escriba el número de registros que se deben leer. El valor predeterminado es `todas las mostradas`. Esta propiedad determina cuántos registros se leen cuando se cargan los datos de la BD. Si el valor es menor que el valor de la propiedad `filas mostradas de la BD`, entonces se usa el valor de `filas leídas de la BD` como valor de `filas leídas de la BD`. Por tanto, la propiedad `filas leídas de la BD` sirve para reducir el número de registros que se cargan en un principio, lo cual acelera el proceso de carga y visualización de datos. Los demás registros se cargan y aparecen cuando el usuario de la vista Authentic hace clic en los botones de navegación de filas (registros).

Nota:

- el número de registros seleccionado afecta a la vista Authentic.
- si se abre un número elevado de tablas en la vista Authentic, el usuario recibe un mensaje de error advirtiendo de que hay demasiadas tablas abiertas. Por un lado, puede reducir el número de tablas que se abren reduciendo el número de registros que se muestran (porque un registro puede contener tablas). Por otro lado, el usuario puede usar consultas para reducir el número de tablas que se cargan en la vista Authentic.

AltovaRowStatus

En el esquema XML que se crea a partir de una BD, todas las tablas tienen el atributo **`AltovaRowStatus`**. StyleVision determina automáticamente el valor de este atributo, que está compuesto por tres caracteres que se inicializan con `---`. Si se modifica una fila o se añade una fila nueva, el valor de este atributo cambia y se usan estos caracteres.

`A`	la fila se añadió (pero no se guardó en la BD)
`U,u`	la fila se actualizó (pero no se guardó en la BD)

 © 2016-2022 Altova GmbH

`D,d, X`	la fila se eliminó (pero no se guardó el cambio en la BD)

Estos valores se pueden utilizar para ofrecer información a los usuarios sobre las filas que está editando. La información de estado de la fila existe hasta que se guarda el archivo. En cuanto se guarda el archivo la información de estado se inicializa (con `---`).

Formato de los componentes de diseño

Cuando se añaden, modifican o eliminan registros, StyleVision aplica cierto formato a los registros añadidos/modificados/eliminados para que el usuario pueda distinguirlos de los demás registros. Los errores de tipo de datos también se señalan en color rojo. Si quiere conservar esta diferenciación, compruebe que el formato asignado a las filas de una tabla no tiene las mismas propiedades que el formato asignado por StyleVision. El formato que asigna StyleVision es este:

Añadido	`A`	en negrita y subrayado
Modificado (actualizado)	`U, u`	subrayado
Eliminado	`D, d, X`	tachado
Error de tipo de datos		texto en color rojo

10.7 Generar archivos de salida

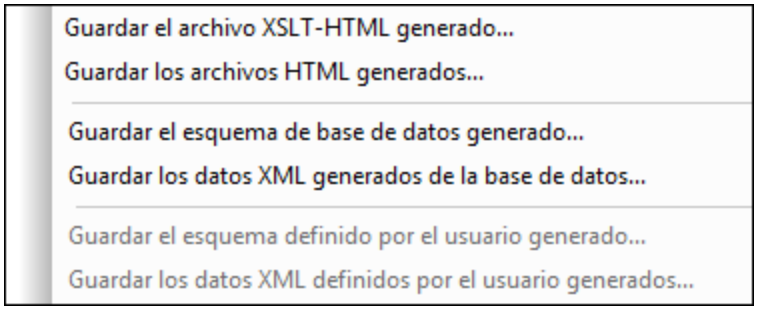
Cuando termine de diseñar el SPS basado en una BD, puede generar archivos de salida y guardarlos. Estos son los archivos que puede generar y guardar:

- El esquema XML basado en la estructura de la BD
- El archivo XML que tiene la estructura del esquema XML generado y el contenido de la BD
- El archivo XSLT para el formato de salida HTML
- El archivo HTML de salida
- El archivo XSLT para el formato de salida RTF
- El archivo RTF de salida

Puede generar y guardar estos archivos desde la interfaz gráfica del usuario o desde StyleVision Server.

Desde la interfaz gráfica de StyleVision

1. Seleccione el comando de menú **Archivo | Guardar archivos generados**. Se abre este submenú:



2. Seleccione qué archivo desea generar. Se abre el cuadro de diálogo "Guardar como".
3. Busque la carpeta donde desea guardarlo, escriba el nombre de archivo que quiere darle y haga clic en **Aceptar**.

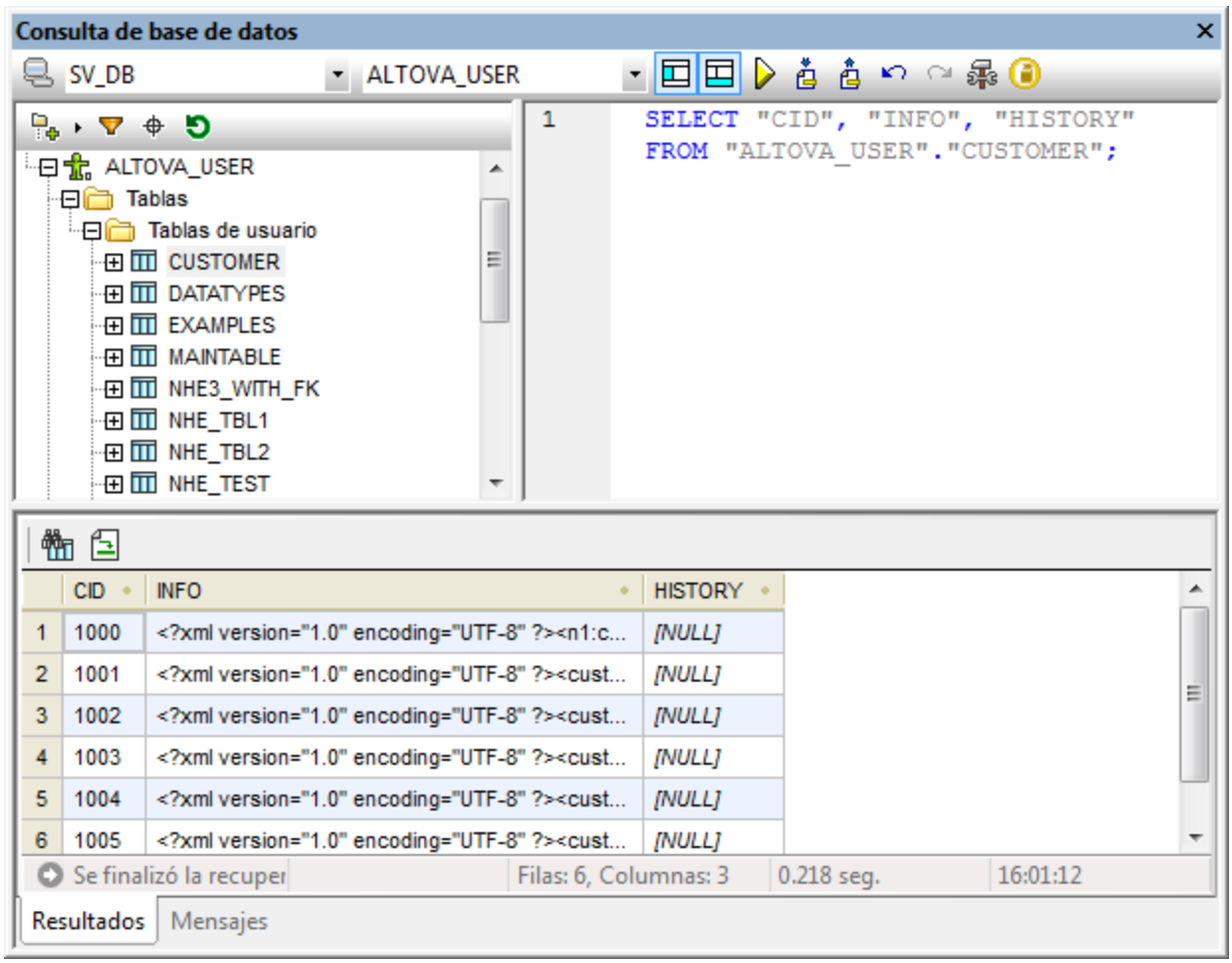
Desde la línea de comandos

Puede llamar a StyleVision desde la línea de comandos para que genere y guarde los archivos asociados con el diseño SPS basado en una BD. No solo puede guardar los archivos de esquema XML y los archivos XSLT, sino también el archivo XML con los datos de la BD y archivos HTML y RTF de salida a partir del diseño SPS.

 © 2016-2022 Altova GmbH

10.8 Consulta de bases de datos

El comando de menú **Base de datos | Consultar la base de datos** abre la ventana "Consulta de base de datos" (*imagen siguiente*). Una vez abierta, esta ventana se puede ocultar/mostrar haciendo clic otra vez en el comando **Base de datos | Consultar la base de datos** o en el icono de la barra de herramientas **Consultar la base de datos** .



Características principales de la ventana "Consulta de la base de datos"

La ventana "Consulta de la base de datos" tiene tres partes:

- El panel **Explorador** 637 situado en la parte superior izquierda, que muestra información sobre la conexión y las tablas de la BD.
- El panel **Consulta** 642 situado en la parte superior derecha, donde se introduce la consulta.
- El panel inferior que tiene las pestañas **Resultados/Mensajes** 645. La pestaña *Resultados* muestra los resultados de la consulta, en lo que denominamos la *cuadrícula de resultados*. La pestaña *Mensajes* muestra mensajes de error y advertencia sobre la ejecución de la consulta.

La ventana "Consulta de la base de datos" tiene una barra de herramientas con varios iconos. Por ahora explicaremos dos de ellos y el resto están documentados en la sección Panel de consultas: descripción y características 642.

Muestra/Oculta el panel Explorador.

Muestra/Oculta el panel Resultados/Mensajes.

Funcionamiento general de las consultas de bases de datos

El funcionamiento general de las consultas de BD aparecen a continuación y se describe con más detalle en los apartados de la presente sección.

1. Se establece la conexión con la BD 634.
2. La BD conectada o partes de ella aparecen en el panel Explorador 637, que se puede configurar según los requisitos de cada momento.
3. En el panel Consulta 644 se introduce una consulta 644 escrita en la sintaxis correspondiente a la BD y la consulta se ejecuta.
4. Los resultados de la consulta 645 se pueden analizar usando filtros.

10.8.1 Orígenes de datos

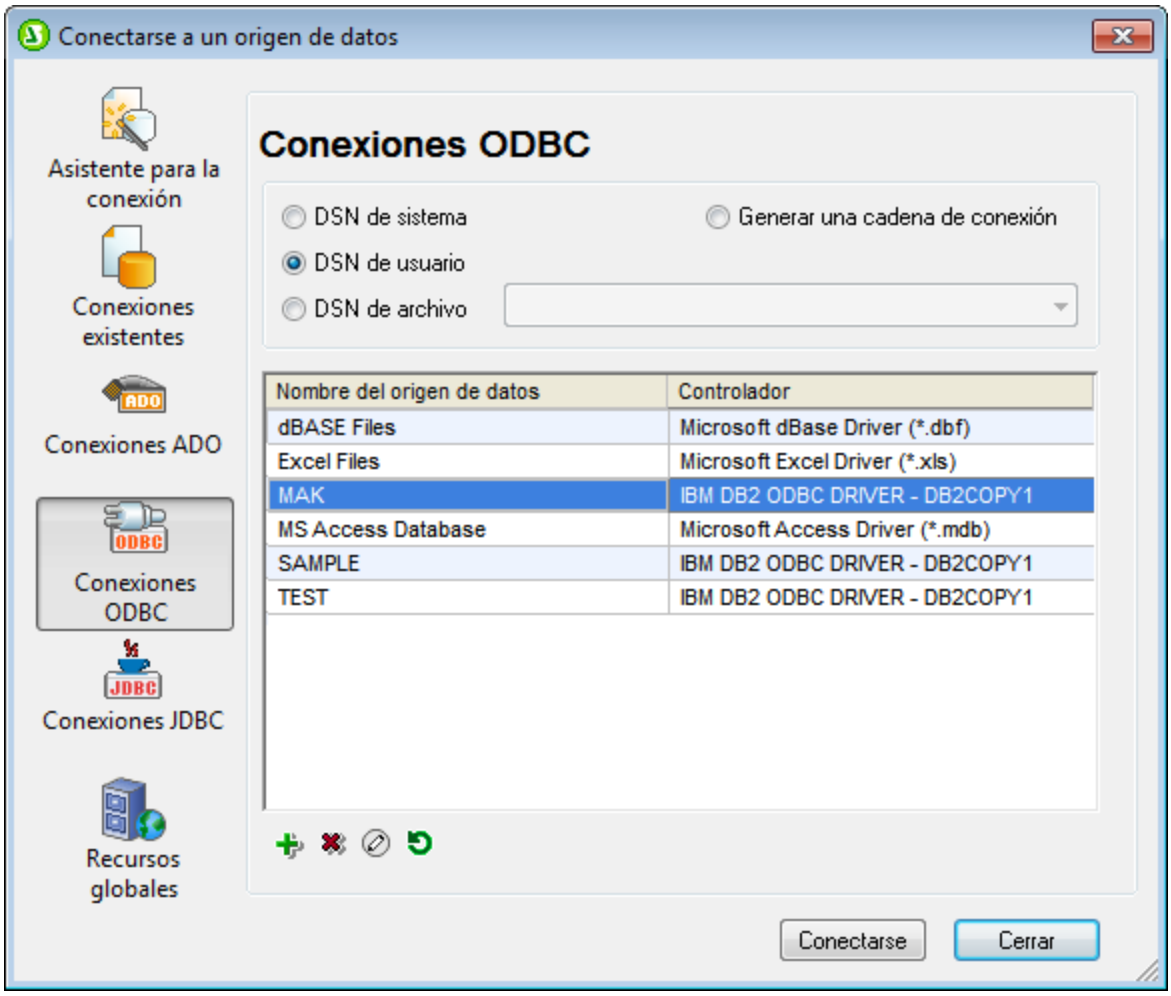
Para poder consultar una BD antes debe a ella. Este apartado describe:

- Cómo conectarse a una BD.
- Cómo seleccionar el origen de datos y el objeto raíz necesarios entre las conexiones existentes.

Conectarse a una base de datos

Al hacer clic en el comando **Base de datos | Consultar la base de datos** por primera vez en una sesión (y cuando no existe ninguna conexión a una BD), se abre el cuadro de diálogo "Conectarse a un origen de datos" (*imagen siguiente*). Una vez establecida la conexión, si quiera crear otra conexión nueva, haga clic en el icono **Conexión rápida** de la ventana "Consulta de base de datos". Si ya existen varias conexiones, puede seleccionar una de ellas 636.

 © 2016-2022 Altova GmbH



El apartado Conectarse a una base de datos 523 explica cómo conectarse a una BD desde el cuadro de diálogo "Conectarse a un origen de datos".

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 StyleVision, compruebe que tiene acceso a los controladores de BD de 64 bits de la BD a la que quiere conectarse.

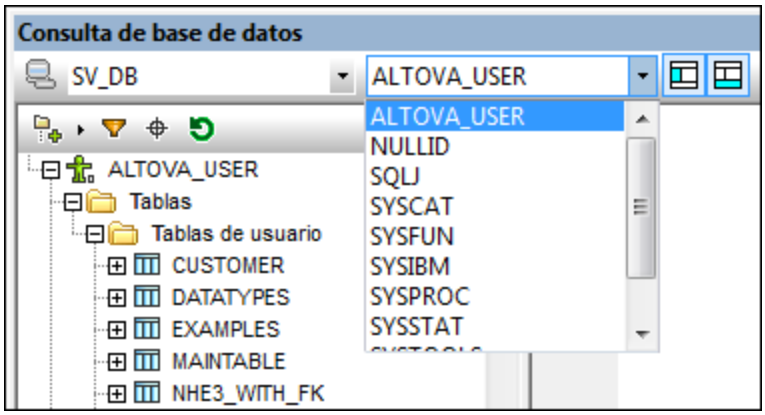
Base de datos	Observaciones
Firebird 2.x, 3.x	
IBM DB2 8.x, 9.x, 10.x, 11.x	
IBM Db2 for i 6.x, 7.4	Los archivos lógicos son compatibles y se muestran en vistas.
IBM Informix 11.70 y superior	
MariaDB 10 y superior	

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 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 apropiado

Todas las conexiones existentes y sus objetos raíz aparecen en dos cuadros combinados de la barra de herramientas de la ventana "Consulta de bases de datos" (*imagen siguiente*). Tras seleccionar el origen de datos en el cuadro combinado de la izquierda, seleccione el objeto raíz en el cuadro combinado de la derecha.

 © 2016-2022 Altova GmbH



En la imagen anterior, por ejemplo, seleccionamos la BD llamada `SV_DB`. Esta BD tiene varios objetos raíz, de los cuales seleccionamos `ALTOVA_USER`. La BD y el objeto de raíz aparecen en el panel Explorador 637.

10.8.2 Explorador: visualizar objetos de la BD

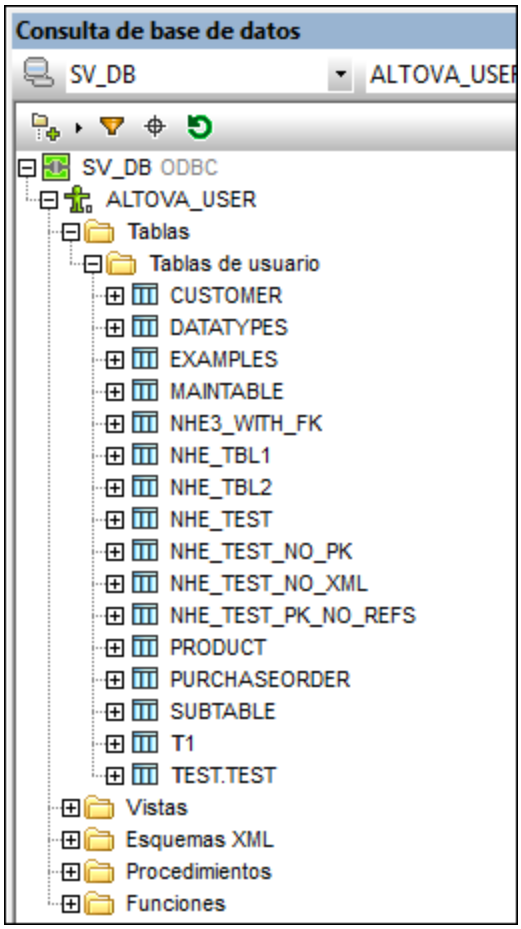
El panel Explorador ofrece una visión de conjunto de los objetos de la BD seleccionada. Esta vista incluye información sobre restricciones de la BD, p. ej. si una columna es una clave principal o foránea. Si se trata de una BD IBM DB2 versión 9, el Explorador también muestra una carpeta con los esquemas XML registrados (*imagen siguiente*).

Este apartado describe todas las características del panel Explorador:

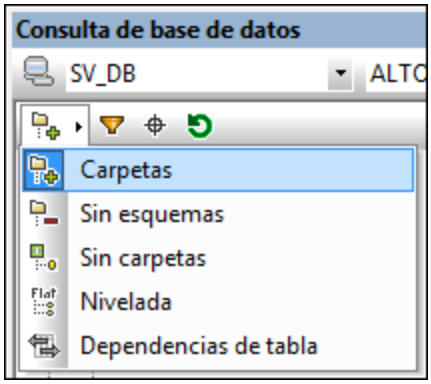
- La disposición de los objetos 637 en el panel Explorador.
- Cómo filtrar 639 objetos de la BD.
- Cómo encontrar 640 objetos de la BD.
- Cómo actualizar 641 el objeto raíz del origen de datos activo.

Disposición de los objetos en el Explorador

La disposición predeterminada es una disposición en carpetas, que organiza los objetos de la BD de forma jerárquica. Al hacer clic con el botón derecho en un objeto, aparece un menú contextual cuyos comandos difieren dependiendo del objeto seleccionado.



Para cambiar la disposición de los objetos del Explorador, haga clic en el icono **Disposición** de la barra de herramientas del Explorador (*imagen siguiente*) y seleccione la disposición de los objetos en la lista desplegable. Observe que el icono cambia de aspecto dependiendo de la opción seleccionada..



Estas son las opciones disponibles:

- **Carpetas:** los objetos de la BD se organizan por tipo en carpetas diferentes, creando un árbol jerárquico. Esta es la opción predeterminada.
- **Sin esquemas:** similar a la opción **Carpetas**, excepto que no ofrece una carpeta para esquemas de la BD. Por tanto, las tablas no se organizan según el esquema de BD.

 © 2016-2022 Altova GmbH

- **Sin carpetas:** los objetos de la BD se organizan jerárquicamente pero no se incluyen en carpetas.
- **Nivelada:** divide los objetos de la BD por tipo en el primer nivel jerárquico. Por ejemplo, las columnas, en vez de aparecer dentro de la tabla correspondiente, aparecen todas en una carpeta llamada `Columnas`.
- **Dependencias de tabla:** las tablas se organizan según su relación con las demás tablas. Las categorías son: tablas con claves foráneas, tablas a las que hacen referencia claves foráneas y tablas que no tienen relación con ninguna otra tabla.

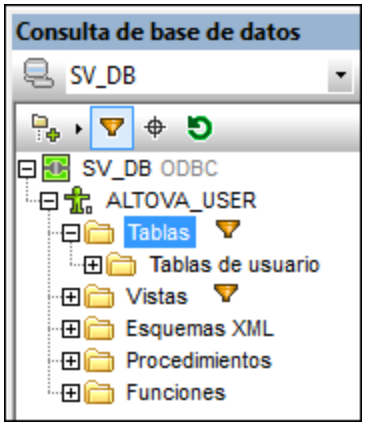
Para ordenar las tablas por tablas del usuario y del sistema, seleccione la opción **Carpetas**, **Sin esquemas** o **Nivelada** y haga clic con el botón derecho en la carpeta `Tablas`. Después seleccione **Ordenar por tablas de usuario y del sistema** en el menú contextual. Las tablas se ordenan alfabéticamente en dos carpetas `Tablas de usuario` y `Tablas del sistema`.

Filtrar los objetos de la BD

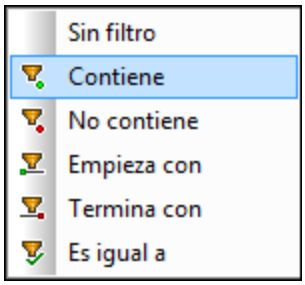
En el panel Explorador (con cualquier disposición excepto la disposición **Sin carpetas** y la disposición **Dependencias de tabla**) puede filtrar los esquemas, las tablas y las vistas usando su nombre o parte de él. Los objetos se filtran a medida que se tecla y el filtro no distingue entre mayúsculas y minúsculas.

Siga estas instrucciones para filtrar los objetos del panel Explorador:

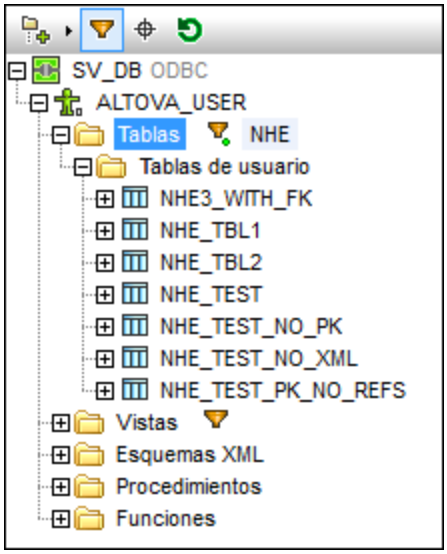
1. Haga clic en el icono **Filtrar contenido de carpeta** de la barra de herramientas del panel Explorador. El icono de filtrado (en forma de embudo) aparece junto a las carpetas `Tablas` y `Vistas` en la disposición actual de los objetos (*imagen siguiente*).



2. Haga clic en el icono de filtrado de la carpeta que desea filtrar y seleccione una opción de filtrado en el menú emergente. Por ejemplo, **Contiene**:



3. Aparece un campo de entrada donde puede introducir la cadena de filtrado. Por ejemplo, en la imagen siguiente, se insertó la cadena de filtrado `NHE` en la carpeta `Tablas`. El filtro se aplica al contenido de la carpeta a medida que se teclea.

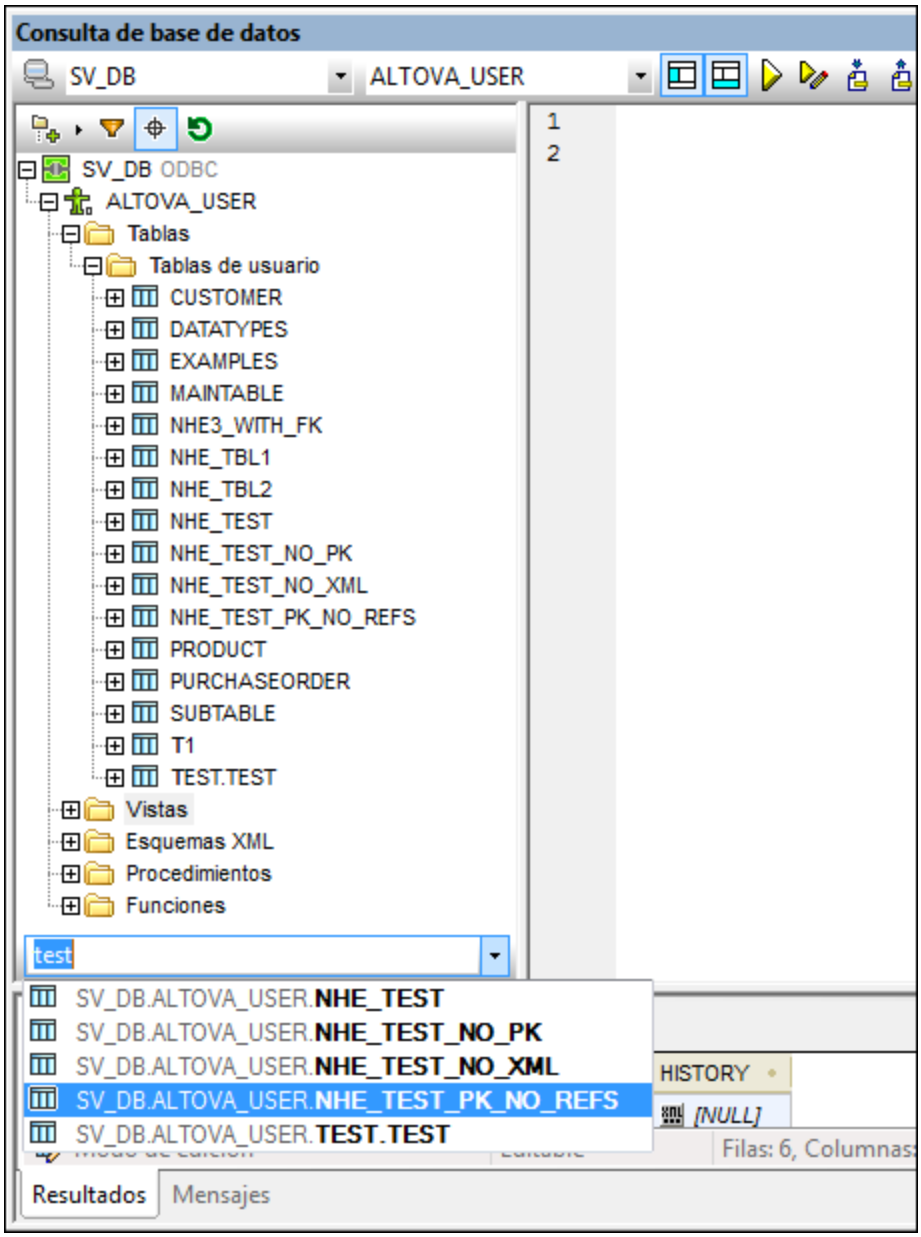


Buscar objetos de la base de datos

Si quiere buscar un objeto concreto de la BD y conoce su nombre, utilice el localizador de objetos del Explorador. Así es como funciona el localizador:

1. En la barra de herramientas del panel Explorador haga clic en el icono **Localizador de objetos**. Aparece una lista desplegable al pie del panel Explorador.
2. Escriba la cadena de búsqueda en el campo de entrada de la lista. En el ejemplo de la imagen siguiente se buscó `n`. Haga clic en la punta de flecha para ver una lista desplegable con todos los objetos que contienen la cadena de búsqueda.

 © 2016-2022 Altova GmbH



3. Haga clic en el objeto que le interesa para verlo en el panel Explorador.

Actualizar el objeto raíz

El objeto raíz del origen de datos activo se puede actualizar con el botón **Actualizar** de la barra de herramientas del panel Explorador.

10.8.3 Panel de consulta: descripción y características

El panel de consulta es un editor SQL inteligente para realizar consultas en la BD seleccionada. Tras introducir la consulta, puede ejecutar la consulta con el comando **Ejecutar** 642 de la ventana "Consulta de base de datos" y ver el resultado y mensajes de ejecución en el panel Resultados/Mensajes 645. El apartado siguiente (Panel de consulta: trabajar con consultas 644) describe con más detalle cómo crear y ejecutar consultas. Este apartado repasa las características principales del panel de consulta:

- Iconos del editor SQL 642 de la barra de herramientas de la ventana "Consulta de base de datos"
- Opciones del editor SQL 642
- Finalización automática de instrucciones SQL 643
- Definición de regiones en un script SQL 644
- Inserción de comentarios en un script SQL 644
- Uso de marcadores 644

Iconos del editor SQL de la barra de herramientas de la ventana "Consulta de base de datos"

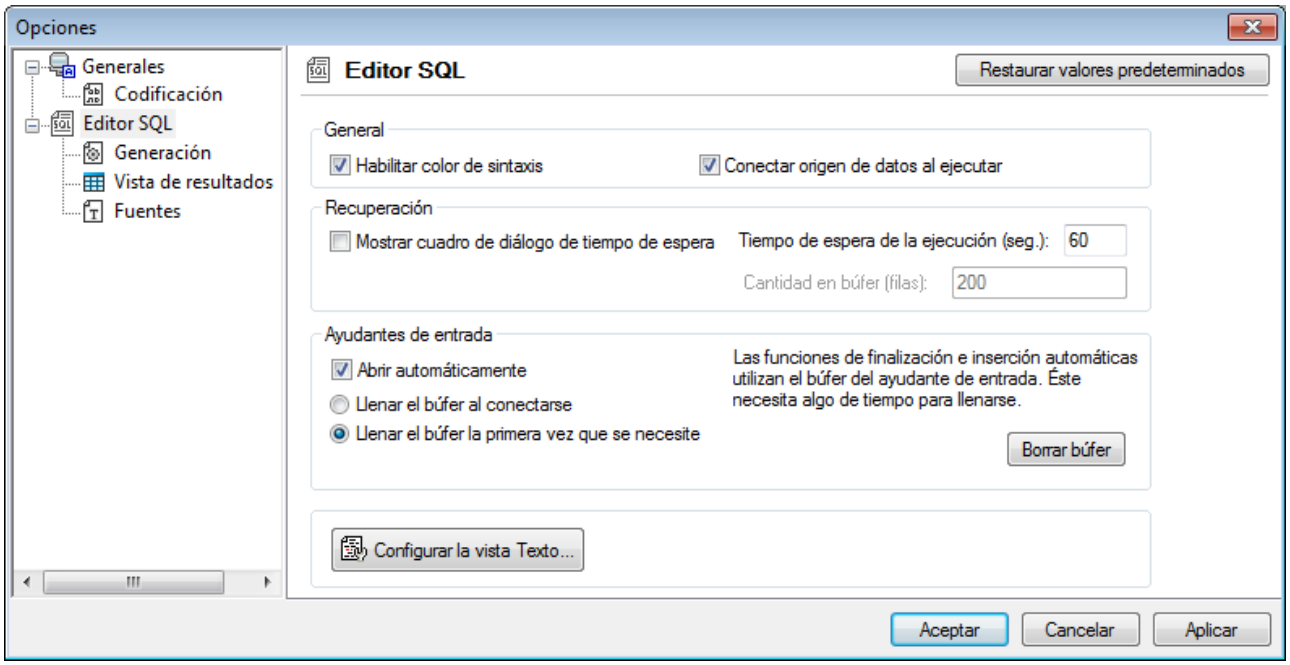
Estos iconos de la barra de herramientas de la ventana "Consulta de base de datos" se usan para trabajar con el editor SQL:

	Ejecutar	Ejecuta la instrucción SQL seleccionada. Si el script contiene varias instrucciones y no selecciona ninguna, se ejecutan todas.
	Importar archivo SQL	Abre un archivo SQL en el editor SQL.
	Exportar archivo SQL	Guarda consultas SQL en un archivo SQL.
	Deshacer	Deshace las entradas realizadas en el editor SQL un número ilimitado de veces.
	Rehacer	Rehace las entradas realizadas y deshechas en el editor SQL.
	Opciones	Abre el cuadro de diálogo "Opciones" del editor SQL.
	Abrir el script SQL en DatabaseSpy	Abre el script SQL en Altova DatabaseSpy.

Opciones del editor SQL

Haga clic en el icono **Opciones** de la barra de herramientas de la ventana "Consulta de base de datos" para abrir el cuadro de diálogo "Opciones" (*imagen siguiente*). En el panel izquierdo de este cuadro de diálogo puede seleccionar qué pantalla se muestra. Haga clic en el botón **Restaurar valores predeterminados** para restaurar los valores de las opciones de la pantalla.

© 2016-2022 Altova GmbH

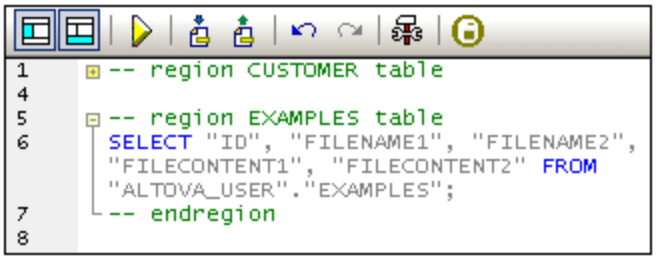


Estas son las principales opciones que puede configurar para el editor SQL:

- ***Generales | Codificación:*** opciones para definir la codificación de archivos SQL nuevos, de archivos SQL existentes cuya codificación no se puede detectar y para establecer la marca de orden de bytes (marca BOM). Si StyleVision detecta su codificación, los archivos SQL se abren y se guardan sin que cambie la codificación.
- ***Editor SQL:*** opciones para activar/desactivar el color de sintaxis y la conexión a origen de datos durante la ejecución. Puede definir un tiempo de espera para la ejecución de la consulta y que aparezca un cuadro de diálogo si se supera 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 cuando se establece la conexión al origen de datos o cuando se necesite por primera vez. El botón **Configurar la vista Texto** abre un cuadro de diálogo donde puede definir la sangría y el tamaño de tabulación que se aplica al texto en el Editor SQL.
- ***Editor SQL | Generación:*** la aplicación genera instrucciones SQL al arrastrar objetos del Explorador hasta el panel de consulta. En esta pantalla puede definir las opciones de generación de instrucciones SQL. En el cuadro de lista *Base de datos* seleccione un tipo de BD y defina las opciones de generación para cada uno de los tipos de BD compatibles. Marque la casilla *Aplicar a todas las bases de datos* para aplicar las opciones activas en ese momento a todos los tipos de BD. Entre las opciones se encuentran anexar el punto y coma a las instrucciones y poner los identificadores entre caracteres de escape.
- ***Editor SQL | Vista de resultados:*** opciones para configurar la pestaña *Resultados*.
- ***Editor SQL | Fuentes:*** opciones para establecer el estilo de las fuentes de texto del editor de texto y de la pestaña *Resultados*.

Definición de regiones en un script SQL

Una región es una sección de un script SQL que se marca y declara como unidad. Las regiones se pueden contraer y expandir para ocultar o mostrar partes del script. También se pueden anidar regiones dentro de otras regiones. Las regiones se delimitan con los comentarios **`--region`** (al principio de la región) y **`--endregion`** (al final de la región). Si quiere, también puede ponerle un nombre a las regiones. El nombre se escribe después del delimitador inicial **`--region`** (*imagen siguiente*).



Para insertar una región, seleccione las instrucciones que desea convertir en región, haga clic con el botón derecho y seleccione el comando **Insertar región**. Se crea una región que se puede expandir o contraer. Si quiere, añada un nombre para la región. En la imagen anterior puede ver que el editor incluye numeración de líneas. Para quitar una región, elimine los dos delimitadores (**`--region`** y **`--endregion`**).

Insertar comentarios en un script SQL

Si quiere puede convertir en comentario parte del script SQL. Estas porciones del script se pasan por alto cuando se ejecuta el script.

- Para convertir un bloque en comentario, márquelo, haga clic con el botón derecho y seleccione **Convertir el bloque en comentario/quitar comentario**. Para eliminar el comentario, márquelo, haga clic con el botón derecho y seleccione el mismo comando.
- Para convertir una línea o parte de una línea en comentario, ponga el cursor donde quiere que empiece el comentario, haga clic con el botón derecho y seleccione Convertir la línea en comentario/quitar comentario. Para eliminar el comentario, márquelo, haga clic con el botón derecho y seleccione el mismo comando.

Uso de marcadores

Puede insertar marcadores en líneas concretas y después navegar por el documento con ayuda de estos marcadores. Si quiere insertar un marcador, ponga el cursor en la línea pertinente, haga clic con el botón derecho y seleccione **Insertar o quitar marcador**. Para navegar al marcador siguiente/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 que incluye el marcador, haga clic con el botón derecho y seleccione **Insertar o quitar marcador** otra vez. Para quitar todos los marcadores, haga clic con el botón derecho y seleccione **Quitar todos los marcadores**.

10.8.4 Panel de consulta: trabajar con consultas

Una vez establecida la conexión con la BD, puede introducir un script SQL en el editor SQL y ejecutarlo. Este apartado describe:

 © 2016-2022 Altova GmbH

- Cómo introducir scripts SQL en el editor SQL.
- Cómo ejecutar scripts en la ventana "Consulta de base de datos".

Estos son los iconos de la barra de herramientas relacionados con estas operaciones:

	Ejecutar consulta	Ejecuta la instrucción SQL seleccionada. Si el script contiene varias instrucciones y ninguna está seleccionada, entonces se ejecutan todas.
	Importar archivo SQL	Abre un archivo SQL en el editor SQL.

Crear instrucciones SQL y scripts en el editor SQL

Estos son los métodos que puede usar para crear instrucciones o scripts SQL:

- *Arrastrar y colocar:* arrastrar un objeto del panel Explorador hasta el Editor SQL. Se genera una instrucción SQL para consultar ese objeto en la BD.
- *Menú contextual:* haga clic con el botón derecho en un objeto del panel Explorador y seleccione **Mostrar en el editor SQL | SELECT**.
- *Entrada manual:* escriba las instrucciones SQL en el editor directamente. La característica de finalización automática le ayudará con la edición.
- *Importar un script SQL:* haga clic en el icono **Importar archivo SQL** de la barra de herramientas.

Ejecutar 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** de la barra de herramientas. Si no está seleccionada ninguna instrucción, se ejecutan todas las instrucciones del script. Los datos de la BD se recuperan y se muestran en una cuadrícula en la pestaña *Resultados* 645. Los mensajes sobre la ejecución de la consulta aparecen en la pestaña *Mensajes* 646.

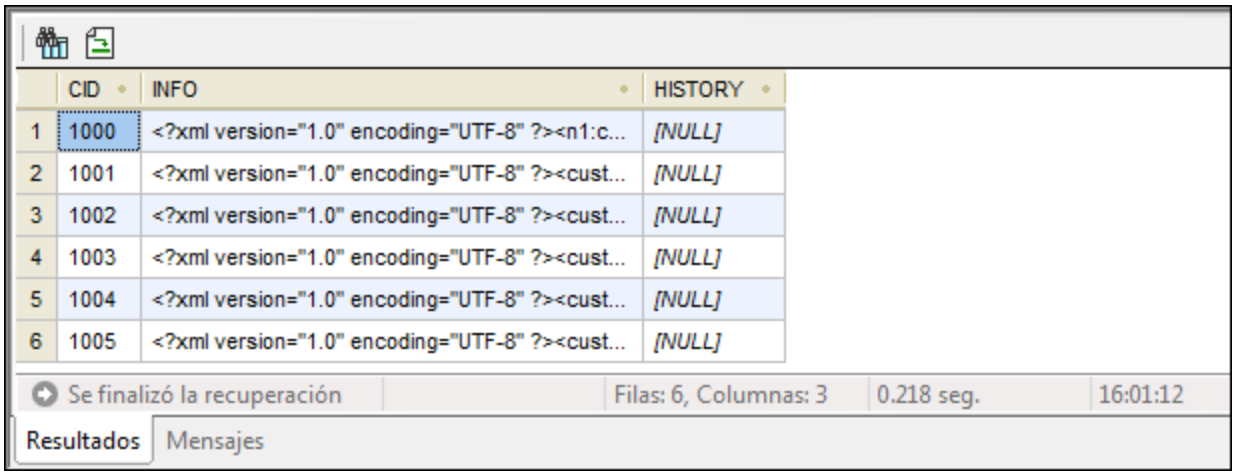
10.8.5 Resultados y mensajes

El panel inferior de la ventana "Consulta de base de datos" tiene dos pestañas:

- La pestaña *Resultados* 645 muestra los datos recuperados por la consulta.
- La pestaña *Mensajes* 646 muestra mensajes sobre la ejecución de la consulta.

Pestaña Resultados

Los datos recuperados por la consulta aparecen en una cuadrícula en la pestaña *Resultados* (*imagen siguiente*).



	CID	INFO	HISTORY
1	1000	<?xml version="1.0" encoding="UTF-8" ?><n1:c...	[NULL]
2	1001	<?xml version="1.0" encoding="UTF-8" ?><cust...	[NULL]
3	1002	<?xml version="1.0" encoding="UTF-8" ?><cust...	[NULL]
4	1003	<?xml version="1.0" encoding="UTF-8" ?><cust...	[NULL]
5	1004	<?xml version="1.0" encoding="UTF-8" ?><cust...	[NULL]
6	1005	<?xml version="1.0" encoding="UTF-8" ?><cust...	[NULL]

Se finalizó la recuperación | Filas: 6, Columnas: 3 | 0.218 seg. | 16:01:12

Resultados | Mensajes

Estas son las operaciones que puede realizar en esta pestaña (con ayuda del menú contextual que aparece al hacer clic con el botón derecho en la cuadrícula de resultados):

- *Ordenar:* haga clic con el botón derecho en la columna cuyos registros desea ordenar y seleccione **Orden | Ascendente/Descendiente/Restaurar valor predeterminado**.
- *Copiar en el portapapeles:* esto se hace en dos pasos: (i) primero seleccione el rango de datos y (ii) después copie la selección. Puede seleccionar datos de varias maneras: (i) haciendo clic en el encabezado de la columna o en el número de la fila para seleccionar la columna/la fila, respectivamente; (ii) seleccionando varias celdas diferentes (pulse la tecla **Mayús** y/o **Ctrl** para seleccionar varias); (iii) haciendo clic con el botón derecho y después en **Seleccionar | Fila/Columna/Todo**. Después haga clic con el botón derecho en el contenido seleccionado y elija el comando **Copiar las celdas seleccionadas**. La selección se copia en el portapapeles y se puede pegar después en aplicaciones externas. Para copiar el encabezado junto con las celdas, elija el comando **Copiar celdas seleccionadas con encabezado**.

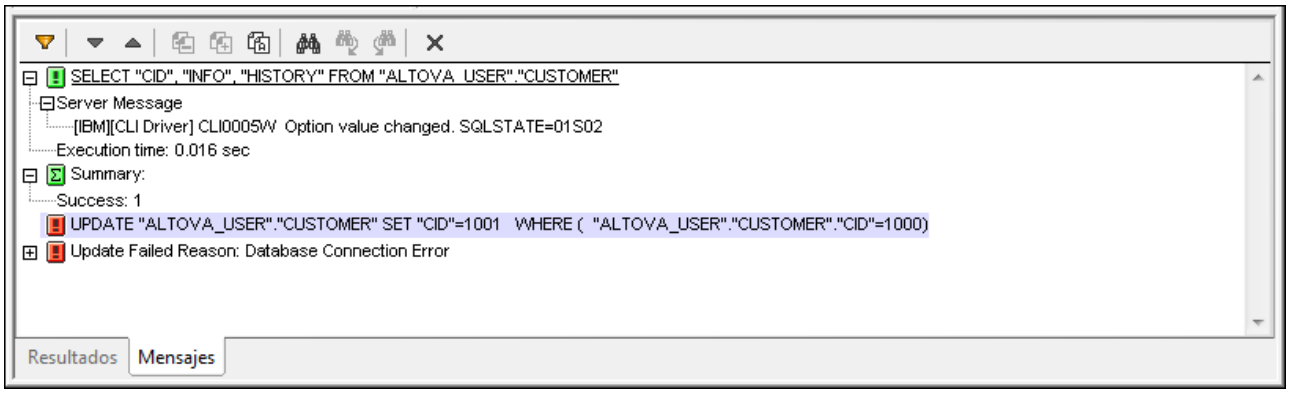
La pestaña *Resultados* cuenta con una barra de herramientas que incluye estos iconos:

	Buscar	Busca texto en la cuadrícula de resultados. También se busca en el contenido de documentos XML.
	Ir a la instrucción	Resalta en el editor SQL la instrucción que produjo el resultado actual.

Pestaña Mensajes

La pestaña *Mensajes* ofrece información sobre las instrucciones SQL ejecutadas y muestra errores y advertencias.

© 2016-2022 Altova GmbH



La pestaña *Mensajes* también cuenta con una barra de herramientas con varios iconos. Estos iconos sirven para personalizar la vista, navegar por ella y copiar mensajes en el portapapeles. El icono **Filtrar** permite escoger qué tipo de mensajes aparecen en la pestaña. Los iconos **Siguiente** y **Anterior** sirven para pasar de un mensaje a otro. Los mensajes también se pueden copiar en el portapapeles, con o sin sus mensajes secundarios. La función **Buscar** sirve para buscar un término de búsqueda en los mensajes. Por último, el icono **Borrar** elimina todo el contenido de la pestaña *Mensajes*.

Nota: estos iconos también están disponibles como comandos en el menú contextual que aparece al hacer doble clic en la ventana de mensajes.

11 Vista Authentic

La vista Authentic (*imagen siguiente*) es una representación gráfica de su documento XML. Permite visualizar documentos XML sin marcado y con características de formato y de entrada de datos apropiadas, como campos de entrada, cuadros combinados y botones de opción. Los datos que el usuario introduce en la vista Authentic se introducen en el archivo XML.

Vista previa Authentic

Mientras edita un diseño SPS en StyleVision puede ver una vista previa de la vista Authentic del archivo XML de trabajo asignado. Si hace clic en la pestaña Formulario electrónico de Authentic, podrá editar el documento XML igual que lo haría en una vista Authentic estándar y los cambios se guardan en el archivo XML de trabajo. Esta sección describe la Vista Authentic 649 y cómo editar documentos en ella 662.

© 2016-2022 Altova GmbH

11.1 Interfaz de la vista Authentic

La vista previa de Authentic se habilita haciendo clic en la pestaña Authentic del documento activo. Si no se asignó un archivo XML de trabajo al archivo SPS, se le pedirá que asigne uno.

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 previa de Authentic.
- 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

11.1.1 La interfaz gráfica del usuario (IGU)

La vista previa de Authentic le ofrece comandos de menú, iconos de barra de herramientas y menús contextuales para editar el documento XML activo en la ventana principal.

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 650.

Ventana principal

Esta es la ventana donde se visualiza y edita el documento XML de trabajo. Para más información, consulte la sección ventana principal de la vista Authentic 652.

Barra de estado

La barra de estado muestra la XPath al nodo que esté seleccionado en cada momento. En la vista previa de Authentic de StyleVision, la XPath al nodo seleccionado se indica en la ventana Estructura del esquema, donde el nodo seleccionado aparece resaltado en color gris. La XPath de la vista previa de Authentic no aparece en la barra de estado.

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.

11.1.2 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 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.

 © 2016-2022 Altova GmbH

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.

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.

Comandos para archivos XML

Estos iconos de la barra de herramientas se corresponden con los comandos del menú **Authentic**. De izquierda a derecha, los comandos son:

- *Guardar los datos XML de Authentic:* guarda el archivo de datos XML.
- *Guardar los datos XML de Authentic como...:* guarda el archivo de datos XML con otro nombre.
- *Volver a cargar la vista Authentic:* vuelve a cargar el archivo de datos XML guardado. Se perderán los cambios que no haya guardado.
- *Validar*: valida el archivo de datos XML.

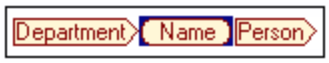
11.1.3 Ventana principal de la vista Authentic

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 650).

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:

 © 2016-2022 Altova GmbH

Nanonull, Inc.

Location: US

Street: 119 Oakstreet, Suite 4876
City: Vereno
State & Zip: DC 29213
Phone: +1 (321) 555 5155 0
Fax: +1 (321) 555 5155 4
E-mail: office@nanonull.com

Vereno Office Summary: 4 departments, 15 employees.

The company was established **in Vereno in 1995** as a privately held software company. Since 1996, Nanonull has been actively involved in developing nanoelectronic software technologies. It released the first version of its acclaimed *NanoSoft Development Suite* in February 1999. Also in 1999, Nanonull increased its capital base with investment from a consortium of private investment firms. The company has been expanding rapidly ever since.

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*).

2006-04-01: *Boston*, *USA*

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.

Office Summary: 4 departments, 15 employees.

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.

Visualización de contenidos

En la vista Authentic el contenido se puede visualizar de dos maneras:

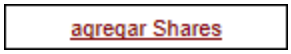
- Texto sin formato. El usuario escribe el texto y este texto se convierte en el contenido del elemento o en el valor del atributo.

- 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]`:

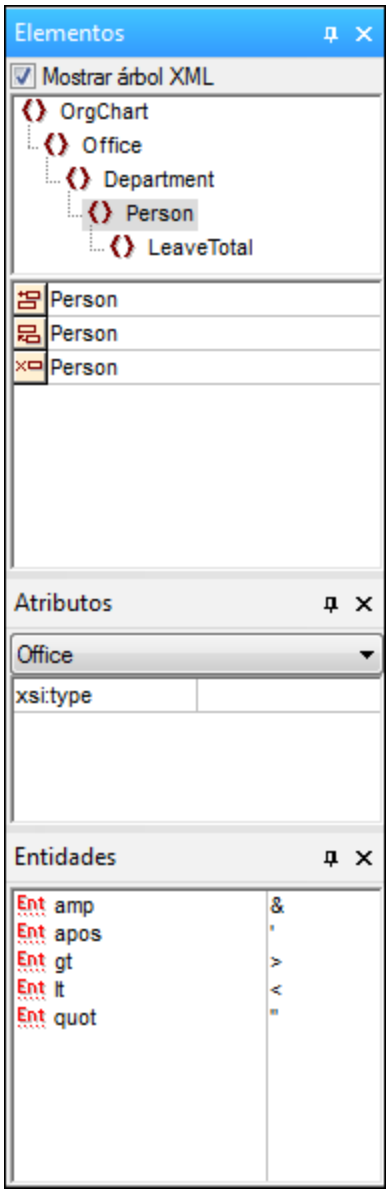


Al hacer clic en el aviso se agrega el elemento y el cursor se coloca en el campo de entrada de datos. Si hay varios nodos opcionales, aparece el aviso `agregar...` . Al hacer clic en el aviso, aparece un menú con los nodos opcionales.

11.1.4 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*).

© 2016-2022 Altova GmbH



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

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

 © 2016-2022 Altova GmbH

`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.

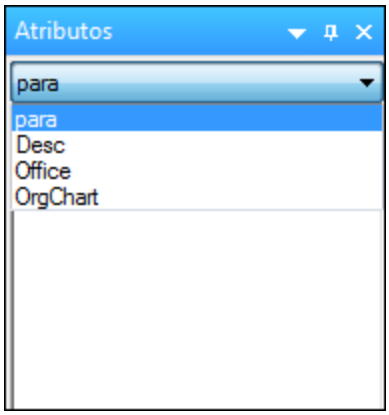
Borrar elemento (cuando se selecciona un texto)
Este icono aparece cuando se selecciona texto dentro de un elemento de contenido mixto. Si hace clic en el icono, se borra el elemento que rodea el texto seleccionado.

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.

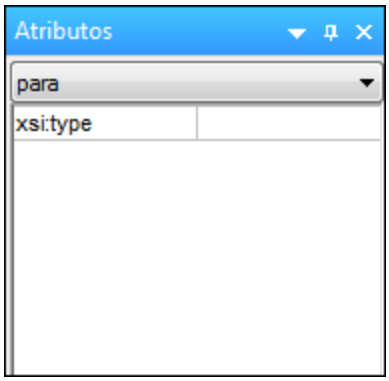
Ayudante de entrada Atributos

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`.



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.

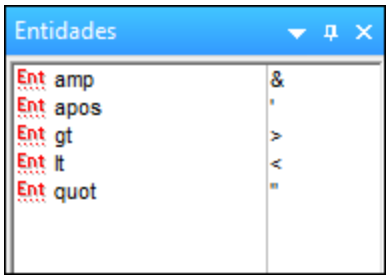
En el caso del atributo `xsi:nil`, que aparece en el ayudante de entrada Atributos cuando se selecciona un elemento que puede tener valores nulos, entonces el valor del atributo `xsi:nil` sólo se puede introducir seleccionando los valores permitidos (`true` o `false`) de la lista desplegable del campo 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.

 © 2016-2022 Altova GmbH



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 683 de la sección *Edición en la vista Authentic*.

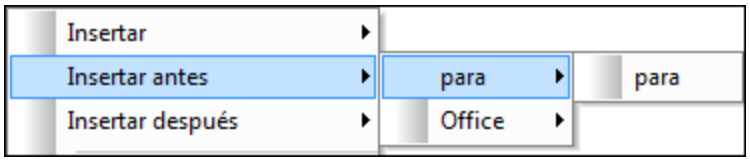
11.1.5 Menús contextuales de la vista Authentic

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`.



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 654 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 683.

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 654.

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:

© 2016-2022 Altova GmbH

- 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.

11.2 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:

- 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 681.
- 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 683.
- Qué formatos de imagen 685 pueden visualizarse en la vista Authentic.

Para aprender a usar todas las funciones de la vista Authentic, consulte y complete el *Tutorial de la vista Authentic* con Altova XMLSpy o con Altova Authentic Desktop. El *Tutorial de la vista Authentic* está disponible en la documentación de estos productos.

Sitio web de Altova: Edición de contenido XML, Creación de contenido XML

11.2.1 Edición básica

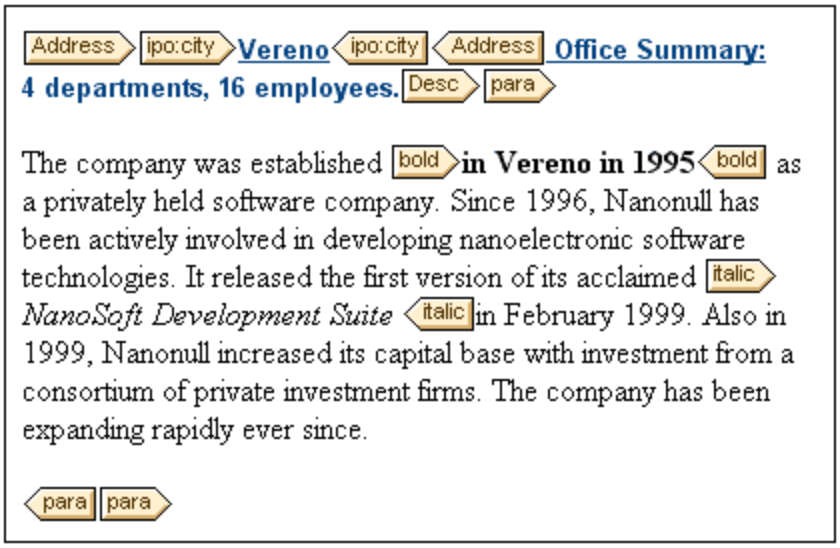
En la vista Authentic puede editar documentos XML. Sin embargo, la vista Authentic puede ocultar el marcado XML del documento y mostrar solamente el contenido del documento (*ver la primera imagen de este apartado*). De este modo, el usuario no entra en contacto con la tecnología XML y puede editar el 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*).

Vereno Office Summary: 4 departments, 16 employees.

The company was established **in Vereno in 1995** as a privately held software company. Since 1996, Nanonull has been actively involved in developing nanoelectronic software technologies. It released the first version of its acclaimed *NanoSoft Development Suite* in February 1999. Also in 1999, Nanonull increased its capital base with investment from a consortium of private investment firms. The company has been expanding rapidly ever since.

Un documento editable en la vista Authentic sin marcado XML.

© 2016-2022 Altova GmbH

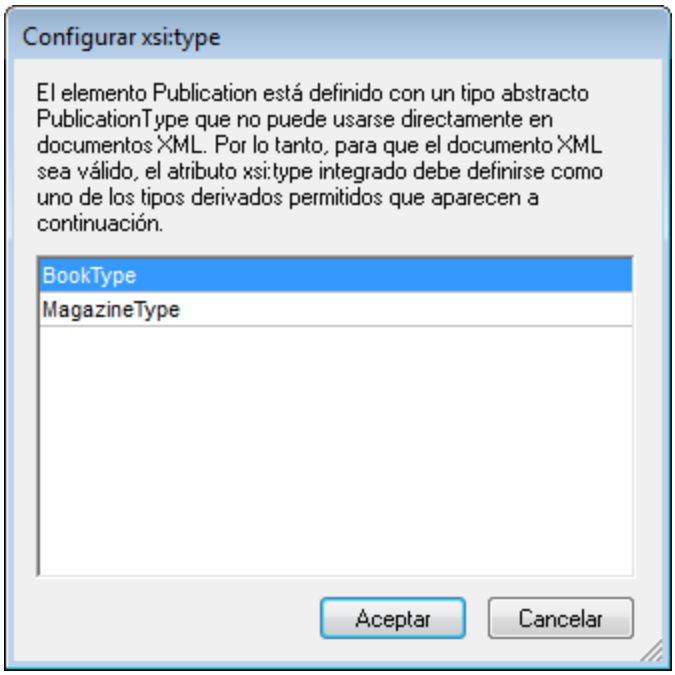


Un documento editable en la vista Authentic con las etiquetas de marcado XML.

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.



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.

 © 2016-2022 Altova GmbH

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 659.

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 655.

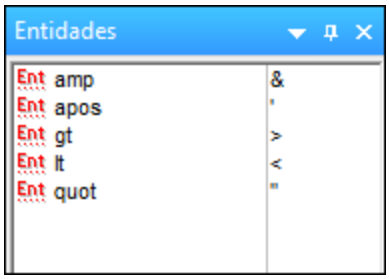
Usar RichEdit en la vista Authentic

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.

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 658 (*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 683 en la vista Authentic directamente. Después de definirlas, estas entidades aparecen en el ayudante de entrada Entidades 658 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 650. 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

 © 2016-2022 Altova GmbH

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.)

11.2.2 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 667 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 668 y de los iconos de edición para tablas CALS/HTML 672 se describen en los apartados siguientes.

11.2.2.1 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

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.

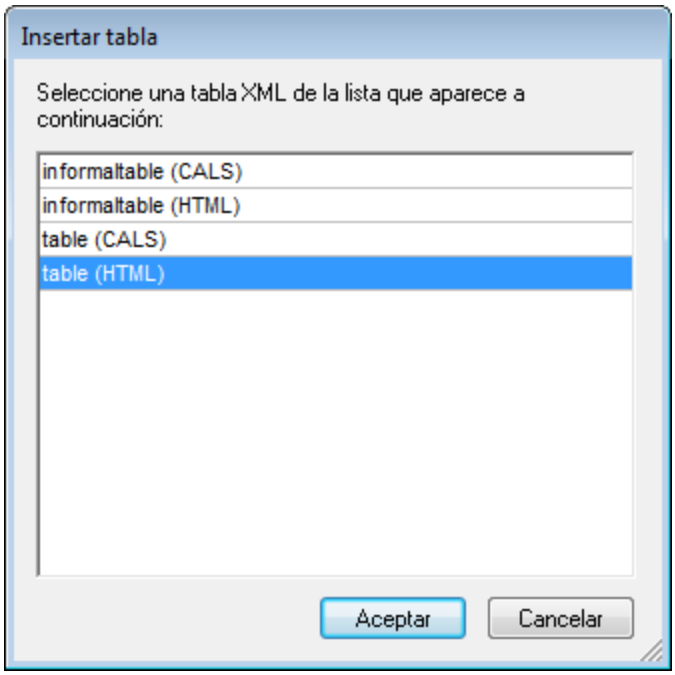
11.2.2.2 Tablas CALS/HTML

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 672).

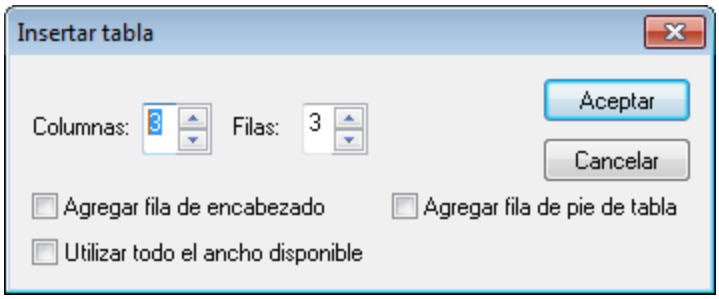
Insertar tablas

Para insertar una tabla CALS/HTML siga estos pasos:

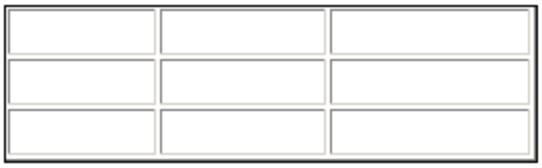
1. Ponga el cursor en la posición donde desea insertar la tabla y haga clic en el icono . (Recuerde que es el esquema el que determina dónde está permitido insertar una tabla). Aparece el cuadro de diálogo "Insertar tabla" (*imagen siguiente*). Este diálogo enumera todas las estructuras de datos de elementos 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.

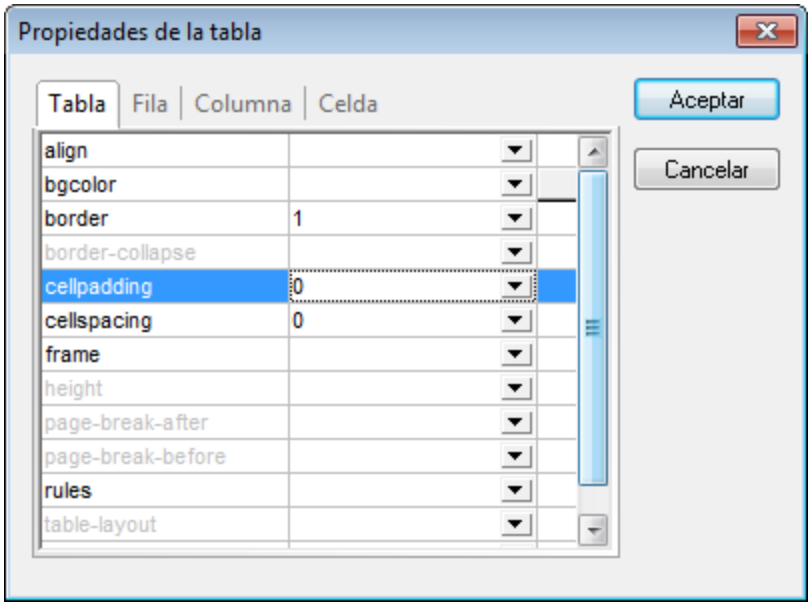
Cómo dar formato a las tablas e introducir datos

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:

- 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.

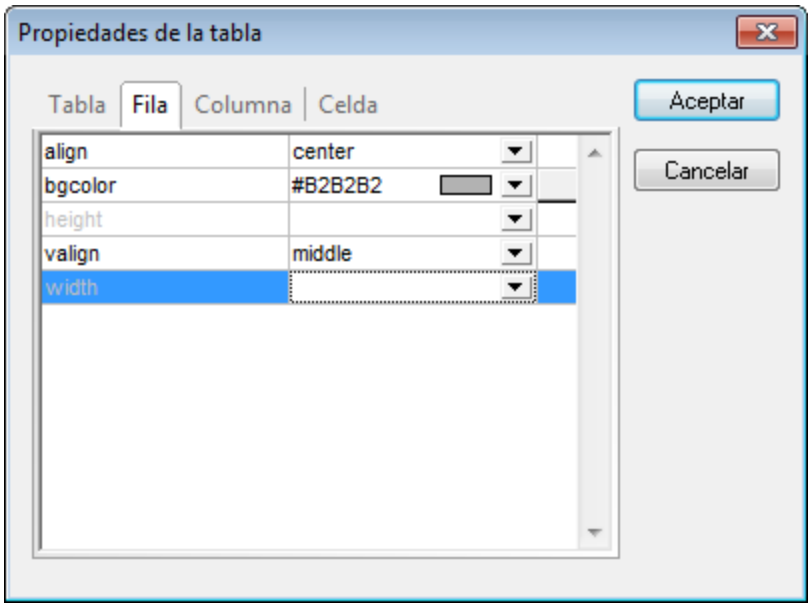
Siga estos pasos para dar formato a una tabla, fila, columna o celda:

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	

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 672.

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.

11.2.2.3 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 668.

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.

 © 2016-2022 Altova GmbH

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

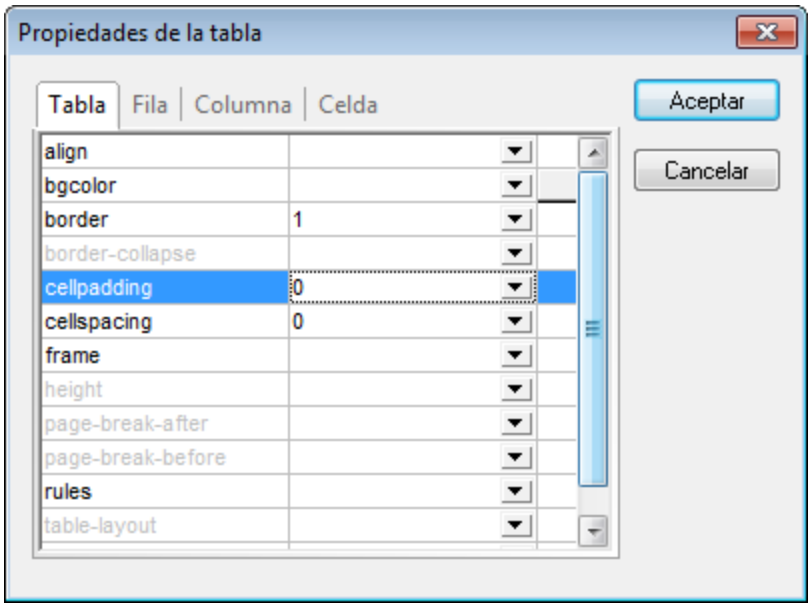
Este comando alinea el contenido en el centro de la celda.

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.



11.2.3 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:

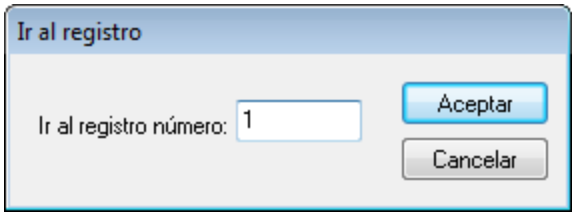
- El diseñador de la hoja de estilos SPS puede limitar el número de registros de una tabla de BD que se presentan en la vista Authentic para que el diseño tenga un aspecto más compacto. Si así fuera, en la vista Authentic se carga únicamente ese número de registros. El usuario puede cargar y visualizar los demás registros de la tabla de BD con ayuda de los iconos de navegación (véase *Navegar por una tabla de BD* 674).
- Puede consultar la BD 675 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 679.

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.

11.2.3.1 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** 856.

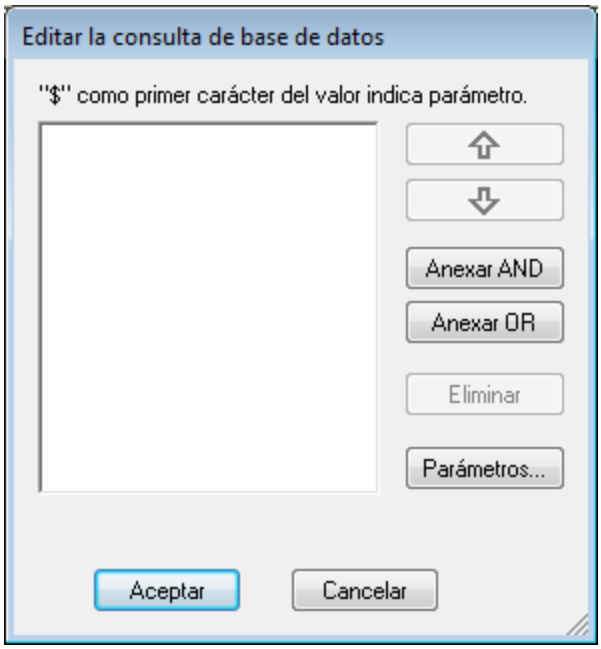
11.2.3.2 Consultas de BD

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.



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 677.
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 678.

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:

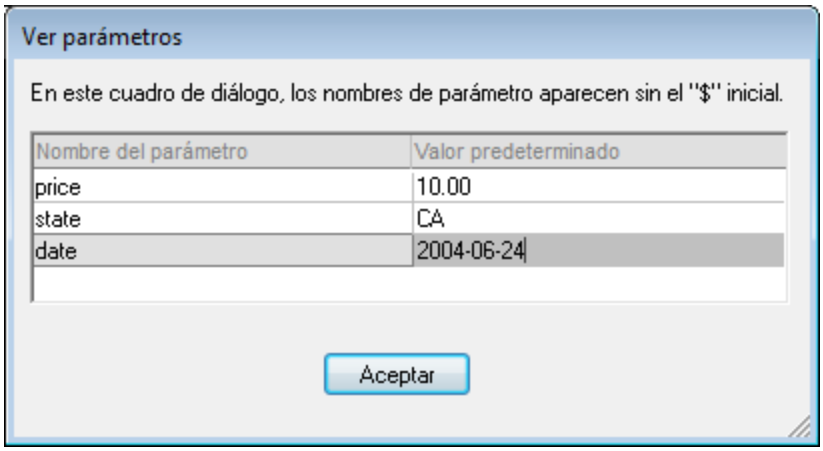
`=`	Igual a
`<>`	No igual
`<`	Menos que
`<=`	Menos o igual que
`>`	Más que
`>=`	Más o igual que
`LIKE`	Fonéticamente igual
`NOT LIKE`	Fonéticamente no igual
`IS NULL`	Está vacío
`NOT NULL`	No está vacío

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).

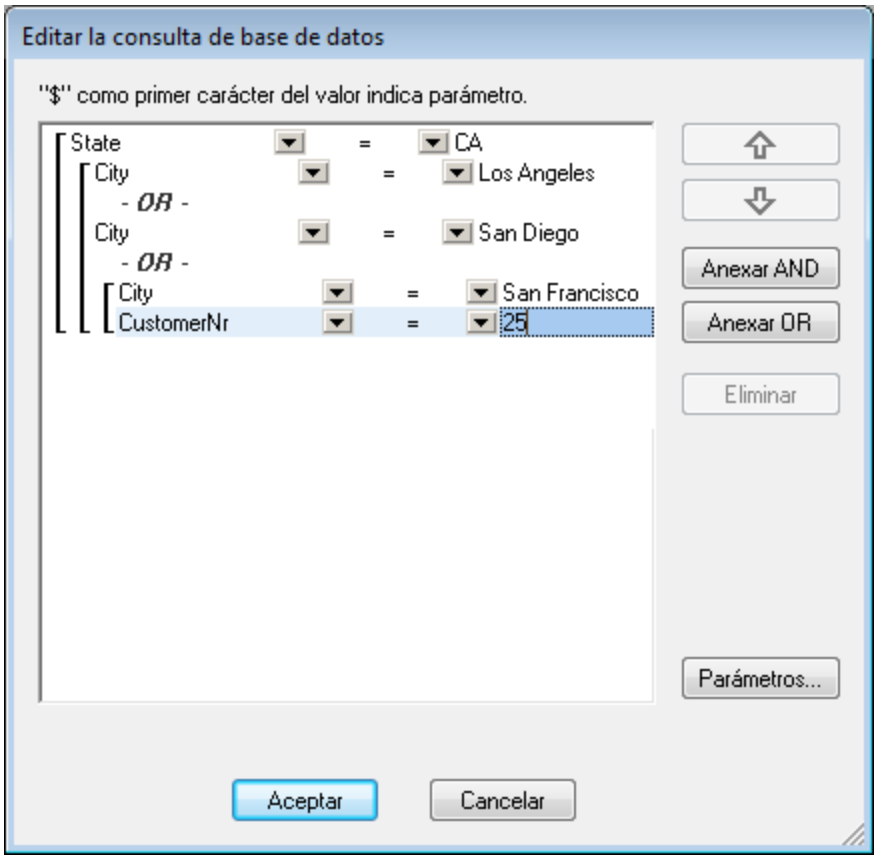
Para ver qué parámetros se definieron en el archivo SPS pulse el botón **Parámetros** del cuadro de diálogo "Editar la consulta de base de datos". Como resultado aparece el cuadro de diálogo "Ver parámetros" (*imagen siguiente*).



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.



 © 2016-2022 Altova GmbH

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.

11.2.3.3 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 los datos XML de Authentic...** 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.

El nuevo registro se agrega a la BD nada más hacer clic en **Archivo | Guardar los datos XML de Authentic...**. 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 674).

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 los datos XML de Authentic...**. 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 los datos XML de Authentic...** 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.

 © 2016-2022 Altova GmbH

11.2.4 Trabajar con fechas

En la vista Authentic las fechas se pueden editar de dos maneras diferentes:

- Con ayuda del selector de fecha 681.
- O escribiendo el valor 682 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

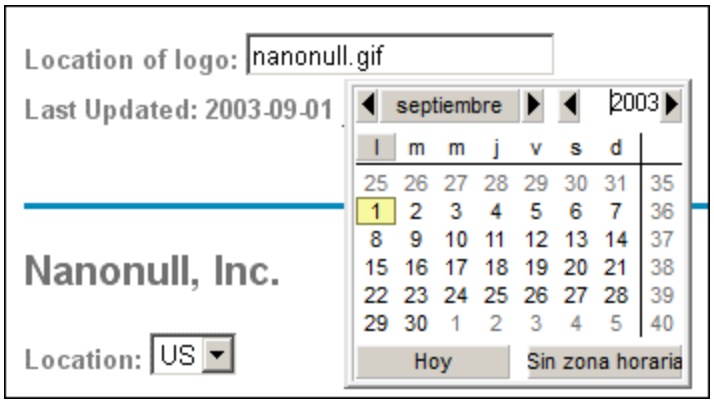
En el documento XML las fechas se almacenan en un tipo de datos de fecha. Para que el documento XML sea válido, la fecha debe almacenarse en un formato léxico concreto, dependiendo del tipo de datos. Por ejemplo, para el tipo de datos `xs:date` la fecha debe tener el formato léxico `YYYY-MM-DD`. Si en 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.

11.2.4.1 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*).

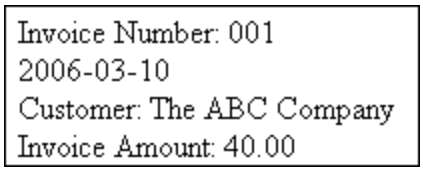


Para seleccionar una fecha, haga clic en la fecha, el mes y el año. Los datos introducidos en el documento XML y los datos de la vista Authentic se actualizan en función de la fecha seleccionada en el selector. Si lo desea también puede introducir la zona horaria.

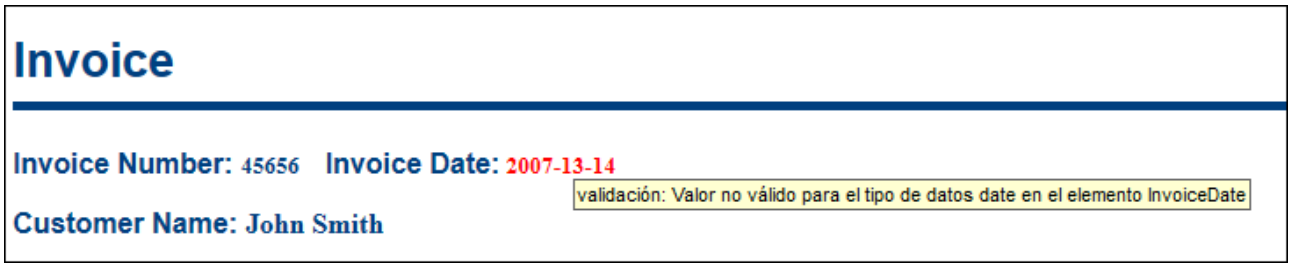
11.2.4.2 Entrada de texto

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*).

© 2016-2022 Altova GmbH

Invoice Number: 001
2006/03/10
Customer: The ABC Company
Invoice Amount: 40.00

11.2.5 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.

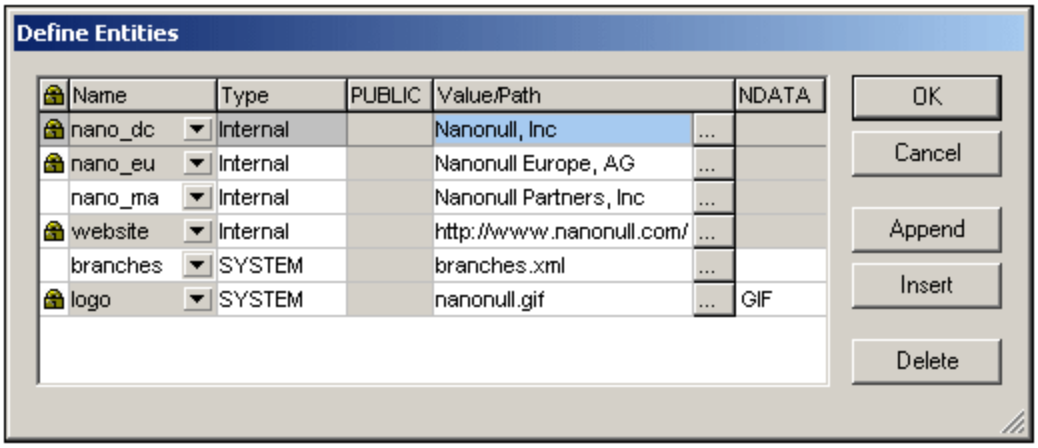
Puede usar entidades para las cadenas de texto, fragmentos XML y recursos externos que vaya a utilizar repetidas veces en el documento. Las entidades (que no son más que una abreviatura de los datos) se definen en el cuadro de diálogo "Definir entidades". Una vez definidas, puede usarlas tantas veces 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*).



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 `&`.

© 2016-2022 Altova GmbH

- 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).

11.2.6 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.

11.2.7 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

12 Scripting con Authentic

Los scripts de Authentic aportan mayor flexibilidad y más posibilidades de interacción en los diseños SPS. Estos diseños se crean y editan en StyleVision Enterprise y Professional Edition y se pueden ver en la vista Authentic de todos los productos de Altova de la edición Enterprise y Professional.

A continuación aparecen los productos de Altova que ofrecen una vista Authentic y que son compatibles con los scripts de Authentic. No obstante, en la versión de confianza del complemento de Authentic para exploradores web se desactivan los scripts internos por motivos de seguridad.

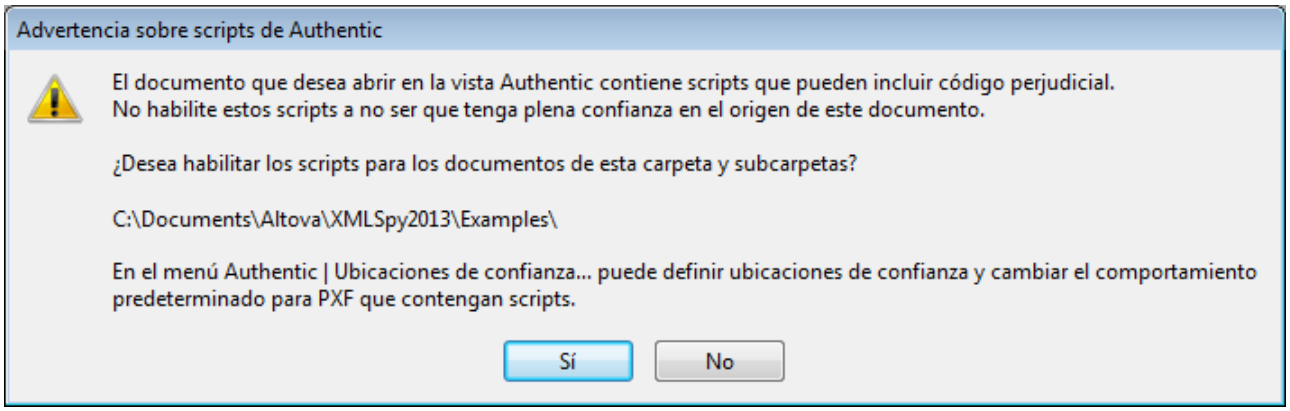
Producto de Altova	Creación de scripts de Authentic	Scripts de Authentic habilitados
StyleVision Enterprise	Sí	Sí
StyleVision Professional	Sí	Sí
StyleVision Basic *	No	No
XMLSpy Enterprise	No	Sí
XMLSpy Professional	No	Sí
Authentic Desktop Enterprise	No	Sí
Authentic Browser Ent Trusted **	No	Sí
Authentic Browser Ent 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 un SPS) incluye un script y este archivo se abre en la vista Authentic (de StyleVision o de otro producto de Altova), la aplicación emite un mensaje de advertencia (*imagen siguiente*).



El mensaje de advertencia ofrece 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**.

- 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".

Nota: cuando se accede a StyleVision desde su interfaz COM (consulte la Referencia del programador 906 para aprender a hacerlo), esta comprobación de seguridad no se hace y la advertencia sobre scripts de Authentic no aparece. El cuadro de diálogo solamente aparece en la vista Authentic de los productos de Altova.

Cómo funcionan los scripts de Authentic

El diseñador del documento SPS puede usar scripts de Authentic para añadir funciones interactivas a los documentos de Authentic. Esto puede hacerse de dos formas diferentes:

- Asignando scripts para acciones definidas por el usuario (macros) a elementos de diseño, botones de barras de herramientas y elementos de menús contextuales.
- Añadiendo al diseño controladores de eventos que reaccionan a eventos de la vista Authentic.

Los scripts necesarios para añadir funciones interactivas a los documentos de Authentic se pueden crear dentro de la interfaz gráfica de StyleVision (edición Enterprise o Professional). Los formularios, macros y controladores de eventos se crean dentro del Editor de scripts de StyleVision y los scripts se guardan con el SPS. Después, en la vista **Diseño** de StyleVision, los scripts guardados se asignan a a elementos del 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.

Apartados de esta sección

Esta sección explica el funcionamiento de los scripts de Authentic y se divide en varios apartados:

 © 2016-2022 Altova GmbH

- Editor de scripts 690: describe la interfaz donde se guarda el proyecto de scripting que contiene los scripts para el documento SPS.
- Macros 691: explica cómo asociar macros a elementos de diseño, a botones de la barra de herramientas de Authentic y a menús contextuales de Authentic.
- Controladores de eventos 697: explica cómo crear y usar scripts de controladores de eventos.
- Opciones de scripting 698: documenta todas las opciones de scripting.

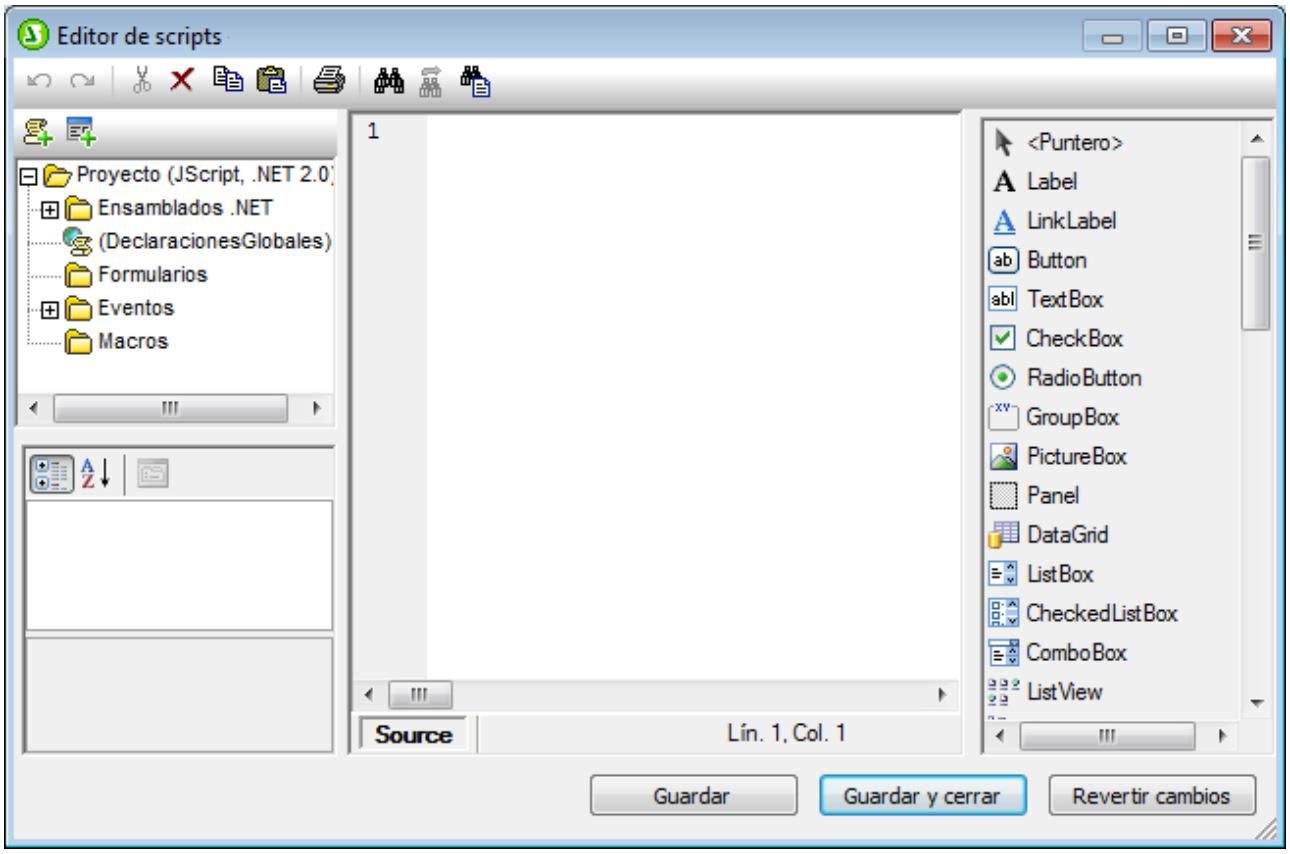
Archivos de ejemplo

En la carpeta `Authentic\Scripting` del proyecto `Examples` puede encontrar archivos SPS de ejemplo con scripts de Authentic.

12.1 Editor de scripts

En el Editor de scripts de StyleVision (*imagen siguiente*) puede crear scripts para formularios, macros y controladores de eventos al diseño SPS.

El Editor de scripts se inicia con el comando de menú **Authentic | Editar scripts de Authentic** o haciendo clic en la pestaña de la vista 37 **Diseño** 37 y seleccionado **Scripts de Authentic**.



Para cambiar el idioma de scripting:

1. Haga clic con el botón derecho en el nodo `Proyecto`, situado al principio de la lista del panel superior izquierdo (*imagen anterior*) del Editor de scripts.
2. Seleccione **Configuración del proyecto** en el menú contextual que aparece.
3. Se abre el cuadro de diálogo "Configuración del proyecto". Seleccione JScript o VBScript y indique el framework .NET de destino. A partir de ese momento el usuario puede acceder a clases y extensiones del Framework .NET seleccionado y usarlas para, por ejemplo, crear formularios.

La *Referencia del programador* 906 describe con más detalle el Editor de scripts 907 y sus características.

Cada documento SPS tiene un solo proyecto de scripting, que se guarda con el documento SPS. El botón **Guardar** del Editor de scripts no guarda el script en el diseño SPS, sino que simplemente guarda en memoria los cambios realizados en el proyecto de scripting. Los cambios en el proyecto de scripting no se guardan en el SPS hasta que se guarda el SPS propiamente dicho.

© 2016-2022 Altova GmbH

12.2 Macros

Las macros son programas que se ejecutan cuando tiene lugar una acción del usuario. Las macros no toman un parámetro, no devuelven un valor y no llaman a otras macros directamente. Sin embargo, si quiere volver a utilizar el mismo código en varias macros o en varios controladores de eventos, puede ponerlo en una función bajo las **`Declaraciones globales`** del Editor de scripts.

La macro que no esté asignada a una acción del usuario concreta no se ejecutará. Ejemplos de acciones de usuario válidas son una interacción con un elemento de diseño concreto 691 o con un botón de la barra de herramientas 694 o la selección de un elemento de menú contextual definido por el usuario 693.

La propiedad **`EventContext`**

Un elemento de diseño de un SPS puede aparecer en varios sitios diferentes. Para establecer en qué contexto se debe ejecutar una macro, la interfaz `AuthenticView` tiene una propiedad llamada `EventContext` que almacena el contexto. A través de la propiedad `EventContext` la macro puede consultar la posición XPath donde se inició la macro, evaluar expresiones XPath en el nodo actual y acceder a las variables definidas en el ámbito de la posición. Para más información consulte la documentación de la API de StyleVision 997.

Revisar referencias a macros

Puede darse el caso de que el diseñador cambie el nombre de la macro o que la elimine del proyecto de scripting, pero que se olvide de cambiar/eliminar la referencia a la macro en el diseño. Para evitar este tipo de errores, el comando **Authentic | Revisar referencias a macros** busca referencias no válidas en el diseño. Si se encuentran referencias no válidas, aparece un cuadro de diálogo donde puede modificar o eliminar las referencias no válidas encontradas.

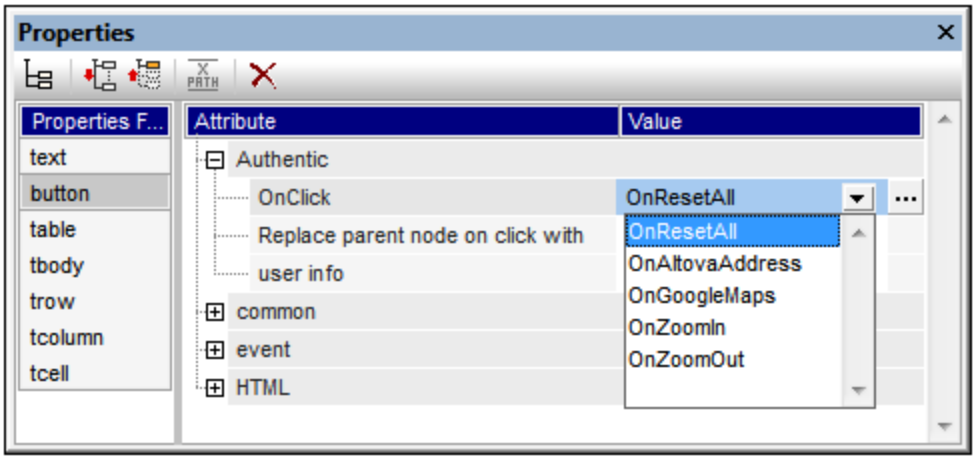
Temas de este apartado

Este apartado explica cómo asociar macros a acciones del usuario válidas:

- Macros en elementos de diseño 691
- Macros en botones de barras de herramientas 694
- Macros en elementos de menús contextuales 693

12.2.1 Macros en elementos de diseño

Puede asignar macros a elementos de diseño desde la ventana Propiedades. Por ejemplo, en la imagen siguiente, la macro llamada `OnResetAll` se seleccionó en la lista de macros disponibles y se asignó al evento `OnClick` de un botón:



Cada elemento de diseño es compatible con acciones de usuario diferentes. Esta tabla incluye todas las opciones posibles.

Acción del usuario	Compatible con estos elementos de diseño	Puede cancelar el evento
OnClick	Botón, Imagen	No
OnBeforeLinkClick	Enlace	Sí
OnBeforeChange	Contenido, Resto de contenido, Campo de entrada, Campo de entrada multilínea, Casilla de verificación, Botón de opción, Cuadro combinado	Sí
OnAfterChange	Contenido, Resto de contenido, Campo de entrada, Campo de entrada multilínea, Casilla de verificación, Botón de opción, Cuadro combinado	No
OnSetFocus	Contenido, Resto de contenido, Campo de entrada, Campo de entrada multilínea, Casilla de verificación, Botón de opción, Cuadro combinado	No
OnKillFocus	Contenido, Resto de contenido, Campo de entrada, Campo de entrada multilínea, Casilla de verificación, Botón de opción, Cuadro combinado	No

A las macros asignadas a estas acciones del usuario solo se les llama cuando se produce una entrada de datos. Es decir, no se les llama con acciones de deshacer ni cuando la acción se realiza fuera de la vista Authentic (por ejemplo, de una llamada desde la API de COM).

Si la macro debe utilizarse para cancelar un procesamiento de eventos posterior, el usuario debe poner una llamada de la API de COM `AuthenticView.DoNotPerformStandardAction()` en la macro correspondiente. Por ejemplo, se puede poner en la macro `BeforeChangeToEU` (*ver fragmento de código más abajo*) al que se asignó el evento `OnBeforeChange` para un botón de opción:

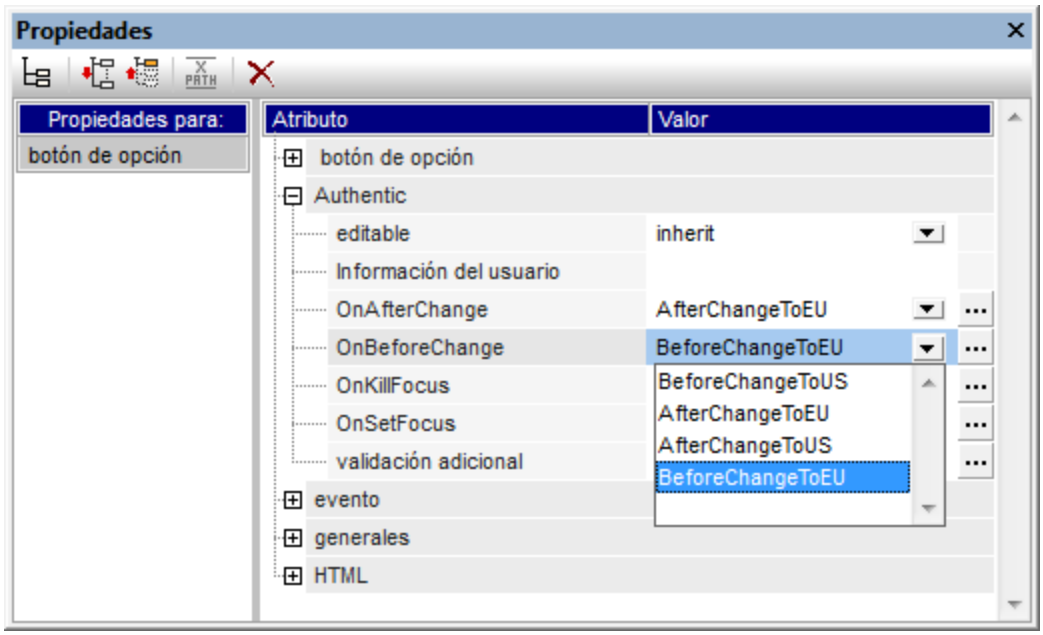
 © 2016-2022 Altova GmbH

```
if ( !confirm( "¿Seguro que desea cambiar los datos a una dirección europea?\nTodos los
datos actuales se eliminarán.") )
       AuthenticView.DoNotPerformStandardAction();
```

Esto evitará que el usuario cambie el valor del botón de opción si el usuario hace clic en **Cancelar** en el diálogo de confirmación. Este ejemplo está en el archivo `OnChange.sps` de la carpeta `Authentic\Scripting` del proyecto `Examples` de StyleVision. Abra el archivo y observe cómo funciona la macro.

Consejo práctico

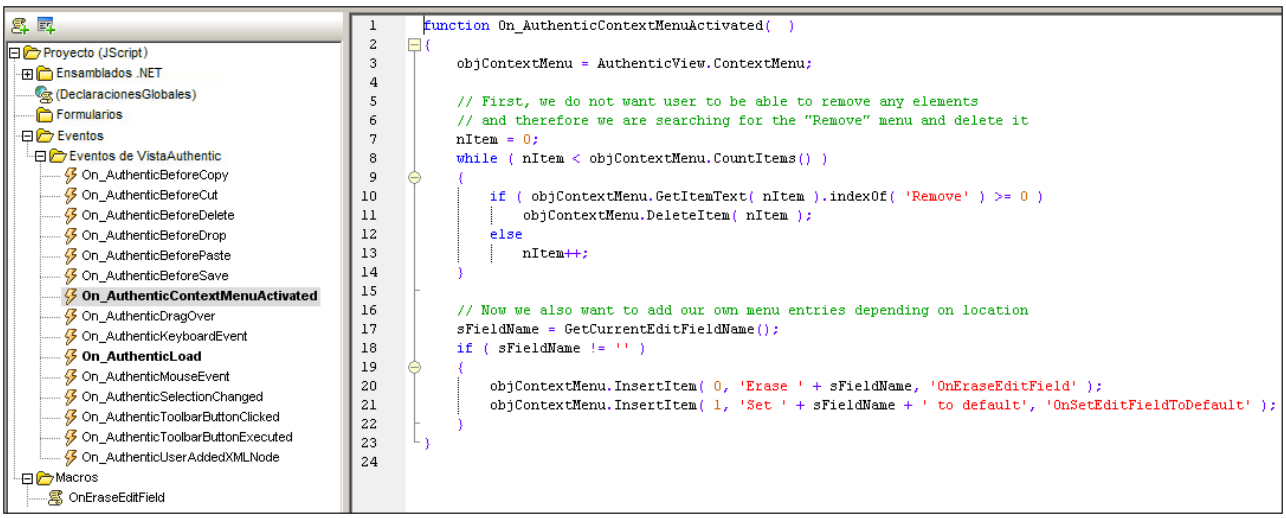
Tras asignar una macro en la ventana Propiedades (*imagen siguiente*), podrá abrirla en el Editor de scripts haciendo clic en el botón **Editar macro**, situado a la derecha de la macro (*imagen siguiente*).



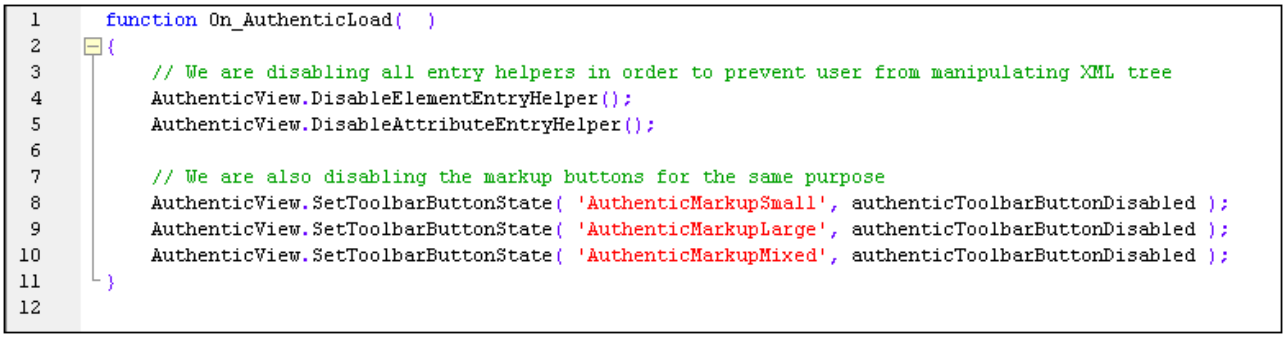
12.2.2 Macros en elementos de menús contextuales

También puede usar macros para personalizar los menús contextuales de Authentic. Es decir, como diseñador del SPS, puede añadir a los menús contextuales comandos personalizados que ejecutan macros definidas previamente.

Para ayudarle en este proceso no hay una interfaz gráfica concreta. El diseñador del SPS debe añadir un controlador de eventos para el evento `On_AuthenticContextMenuActivated` y después manipular el menú desde la API de AuthenticView[931]. Puede encontrar une ejemplo en el archivo `OnContextMenu.sps` de la carpeta `Authentic\Scripting` del proyecto `Examples` de StyleVision. La imagen siguiente muestra el script del controlador de eventos `On_AuthenticContextMenuActivated` en el Editor de scripts.



El controlador de eventos para el evento `On_AuthenticContextMenuActivated` crea dos elementos de menú y les asigna macros. Este archivo de ejemplo también contiene un controlador de eventos para el evento `On-AuthenticLoad`, que deshabilita los ayudantes de entrada y los botones de marcado cuando se carga la vista Authentic.



12.2.3 Botones personalizados

En el diseño SPS puede crear botones personalizados y añadirlos a la barra de herramientas de Authentic, especificando qué macro desencadenará cada botón. Para añadir un botón personalizado a la barra de herramientas de Authentic, primero debe seleccionar el comando **Authentic | Personalizar botones de la barra de herramientas**, que abre el cuadro de diálogo "Botones personalizados de Authentic" (*imagen siguiente*). Aquí haga clic en el botón **Agregar** situado en la esquina superior izquierda y después seleccione el tipo de botón personalizado que desea agregar.

Existen dos tipos de botones personalizados:

- Los **iconos de estado del texto** cambian el estilo del texto (p. ej. de negrita a cursiva) y están asociados a un elemento concreto. Estos elementos suelen aparecer dentro de elementos de tipo de contenido mixto (es decir, en elementos que pueden contener tanto elementos secundarios como texto). Por ejemplo, el elemento `para` es de contenido mixto porque puede contener texto y elementos secundarios como `bold` e `italic`. En casos así, puede asociar un icono de estado del texto (que es una imagen de mapa de bits) a un elemento secundario del elemento de contenido mixto. Por ejemplo,

 © 2016-2022 Altova GmbH

puede asociar el elemento `bold` a una imagen (es decir, a un icono de estado del texto). Después se define una plantilla global para el elemento `bold` para dar el formato del elemento `bold`. En la vista Authentic, el icono de estado del texto para el elemento bold aparecerá en la barra de herramientas, como un botón más. Cuando el usuario de la vista Authentic seleccione texto dentro de un elemento para como el de nuestro ejemplo y haga clic en el botón de la barra de herramientas que está asociado al elemento `bold`, alrededor del texto seleccionado se insertan las etiquetas del elemento `bold` y se aplica el formato negrita de la plantilla global al texto.

- Los **botones personalizados** de la barra de herramientas que están asociados a una macro.

Los botones personalizados toman estos parámetros:

- La ubicación de una imagen para el botón (en el campo *Mapa de bits*) o el texto para el botón (en el campo *Texto*).
- En el campo *Nombre del elemento* (campo disponible para iconos de estado del texto) escriba el nombre del elemento con el que desea asociar el icono de estado.
- En el cuadro combinado *Nombre de la macro* (campo disponible para botones personalizados) seleccione una macro de la lista desplegable. Las macros de esta lista son las que están guardadas con el SPS. Si hace clic en el botón **Crear**, se abre el Editor de scripts 907 de StyleVision en una ventana nueva, donde puede crear la macro con facilidad. Si hace clic en el botón **Editar**, la macro seleccionada se abre en el Editor de scripts 907, donde puede editarla.
- Si quiere, puede añadir un texto en el campo *Información rápida* que aparecerá cuando el usuario de la vista Authentic pase el puntero del ratón por el botón personalizado.

- En el campo *Identificador* debe escribir la cadena de texto que se usará para identificar el botón personalizado. Este identificador se puede usar en el código del script si el diseñador desea controlar el estado del botón a través de la API AuthenticView 931.

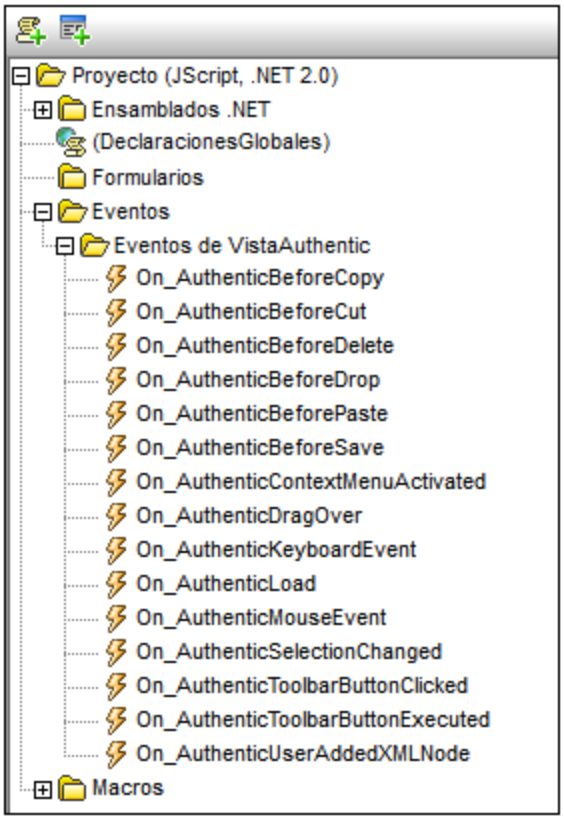
En el ejemplo de la imagen anterior se creó el botón personalizado **US**, que tiene asignada la macro `ChangeAddressToUS`. Este botón personalizdo usa una imagen llamada `US.bmp`. Si quiere puede escribir texto de reserva para el botón. En el ejemplo se introdujo un texto de información rápida y el botón tiene el identificador `Dirección_US`. Este ejemplo está en el archivo `ToolbarButtons.sps` de la carpeta `Authentic\Scripting` del proyecto `Examples` de StyleVision.

 © 2016-2022 Altova GmbH

12.3 Controladores de eventos

Gracias a los scripts de Authentic puede definir controladores de eventos que reaccionen a acciones del usuario generales (es decir, no relacionadas con un elemento de diseño concreto). Es un mecanismo similar al de añadir agentes de escucha de COM para eventos AuthenticView en aplicaciones externas, pero los controladores de eventos se pueden editar y almacenar en el diseño más fácilmente.

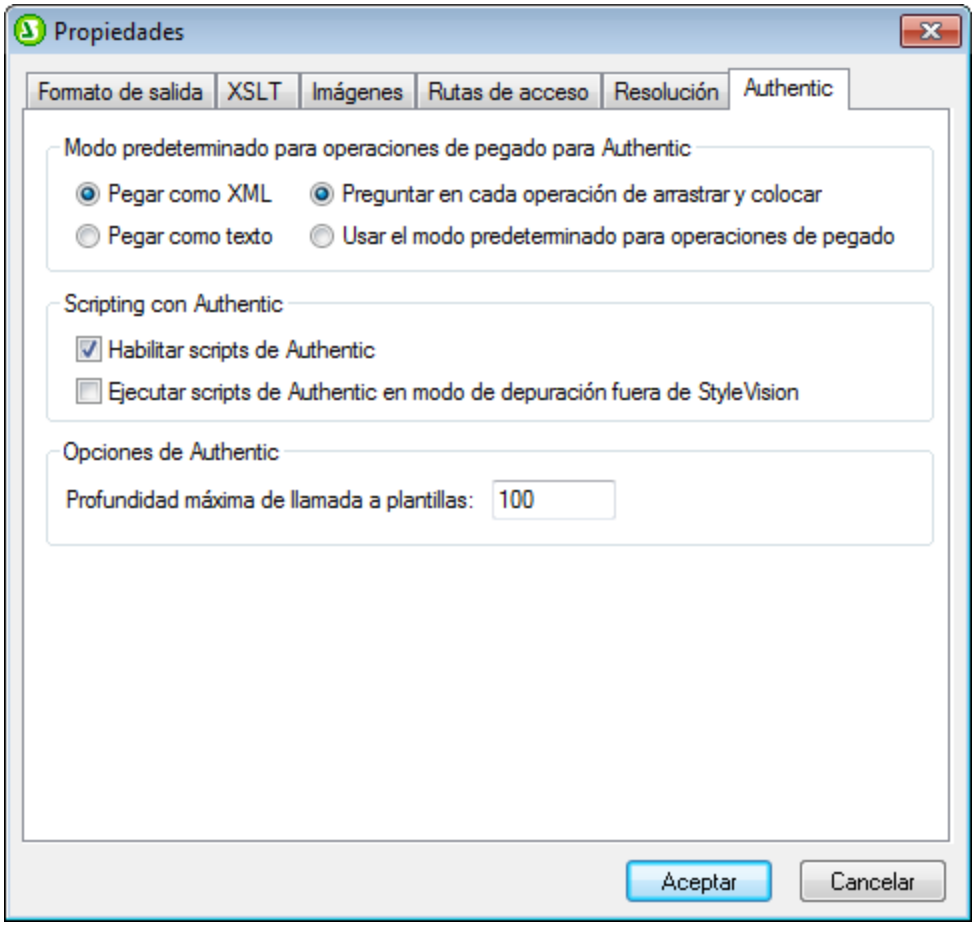
Para añadir un script de controlador de eventos, inicie el Editor de scripts (**Authentic | Editar scripts de Authentic**). Los controladores de eventos disponibles aparece en el panel superior izquierdo, bajo el nodo **`Eventos de VistaAuthentic`**.



Haga doble clic en un evento para abrir su plantilla de script en la ventana de scripting. Después edite la plantilla en la ventana de scripting. Para ver un ejemplo de uso de los controladores de eventos, consulte el apartado Macros en elementos de menús contextuales 693.

12.4 Opciones de scripting

Para poder mostrar documentos interactivos con scripts de Authentic, los scripts deben habilitarse en las propiedades de archivo del SPS. Para ello haga clic en el comando **Archivo | Propiedades**, que abre el cuadro de diálogo "Propiedades" (*imagen siguiente*). Seleccione la pestaña *Authentic* y marque la casilla *Habilitar scripts de Authentic*.



También puede indicar si las macros se ejecutan en modo de depuración o no desde fuera de StyleVision (es decir, en la vista Authentic de otros productos de Altova). Esta opción es muy práctica si quiere probar y depurar una macro fuera del entorno de StyleVision.

© 2016-2022 Altova GmbH

13 Procesamiento automatizado

StyleVision ofrece potentes posibilidades de automatización, gracias a sus diferentes funciones de generación de archivos y a los diferentes formatos de salida.

Función de generación de archivos de StyleVision

Cuando termine de crear su diseño SPS con StyleVision, puede generar varios tipos de archivos XSLT y de salida desde la interfaz gráfica de la aplicación. El tipo de archivos que puede generar depende de la edición de StyleVision que utilice (Enterprise, Professional o Basic). Estos archivos XSLT y de salida se generan con el comando **Archivo | Guardar archivos generados** 782. Estos son los archivos que puede generar con su edición de StyleVision:

- Archivos XSLT para documentos de salida HTML, texto y RTF.
- Documentos de salida en formato HTML, texto y RTF.

Como puede observar en la lista anterior, StyleVision puede generar dos tipos de archivos:

1. Archivos XSLT generados a partir del diseño SPS
2. Archivos de salida finales (como HTML).

Nota: si además usa bases de datos como documentos fuente, puede generar un esquema XML y archivos de datos XML basados en la estructura y en el contenido de la BD.

Los archivos de salida HTML, texto y RTF finales se crean en un solo paso, en el que StyleVision usa una hoja de estilos XSLT para transformar el documento XML en el formato de salida.

StyleVision Server y RaptorXML: generar archivos desde fuera de la interfaz gráfica

Además de generar hojas de estilos XSLT y los documentos de salida correspondientes desde la interfaz gráfica de StyleVision (con el comando **Archivo | Guardar archivos generados** 782) también puede generar archivos de salida de dos formas más:

1. Puede generar varios tipos de documentos de salida con StyleVision Server, que llama a la función de generación de archivos de StyleVision sin abrir su interfaz gráfica.
2. Con RaptorXML 703, una aplicación independiente de Altova que incluye el validador XML(+XBRL) y los motores XSLT y XQuery de Altova. Los motores XSLT de RaptorXML se pueden usar para transformar XML en diferentes formatos de salida procesando los documentos XML con hojas de estilos XSLT. El archivo XSLT debe crearse por adelantado para que RaptorXML pueda usarlo. Es decir, RaptorXML no toma el diseño SPS como parámetro de entrada. Las ventajas de usar RaptorXML para estas transformaciones son: (i) mayor velocidad porque permite transformar archivos de gran tamaño en un tiempo récord y (ii) el hecho que, además de ofrecer una interfaz de la línea de comandos, RaptorXML tiene interfaces para COM, Java y .NET. Esto significa que se le puede llamar desde estos entornos. Para más información consulte el apartado RaptorXML 703.
3. Puede realizar transformaciones múltiples en función de desencadenadores preconfigurados (una vez al día, por ejemplo) usando Altova StyleVision Server dentro de un flujo de trabajo de Altova FlowForce Server. Esto se describe en la sección Automatización con FlowForce Server 705.

13.1 Interfaz de la línea de comandos

Puede llamar a las funciones de StyleVision desde la línea de comandos de dos maneras diferentes:

- Llamando al ejecutable de StyleVision 700. Así conseguirá acceso a la función de generación de archivos XSLT de StyleVision. Los archivos XSLT se generan a partir del archivo SPS.
- Usando StyleVision Server 702 para generar archivos de salida (HTML, etc.) Los archivos de salida se generan a partir de un archivo PXF, que es un paquete formado por un archivo SPS y sus archivos relacionados (archivos XML, XSD, archivos de imagen, etc.). El archivo PXF se crea en StyleVision.

Uso de la línea de comandos

Puede usar la línea de comandos de dos maneras distintas:

- Puede introducir comandos en la línea de comandos y ejecutarlos inmediatamente. Por ejemplo, en la ventana de símbolo del sistema puede insertar un comando para StyleVision o StyleVision Server 702 y pulsar **Entrar** para ejecutar el comando.
- Puede introducir una serie de comandos en un archivo de procesamiento por lotes y ejecutarlo. Por ejemplo:

```
@ECHO OFF
CLS
StyleVision TestEN.sps -outxslt=HTML-EN.xslt
StyleVision TestDE.sps -outxslt=HTML-DE.xslt
StyleVision TestES.sps -outxslt=HTML-ES.xslt
```

  Cuando se procese el archivo por lotes, los comandos se ejecutarán y los archivos de salida se generarán.

Funciones de StyleVision en tareas programadas

Puede usar la herramienta **Tareas programadas** de Windows para definir la ejecución de comandos de StyleVision a una hora determinada y en una fecha determinadas. La tarea puede ser un solo comando o un archivo por lotes. Para aprender a crear este tipo de comandos consulte el apartado *Automatizar el procesamiento de datos* 707.

13.1.1 StyleVision

Esta es la sintaxis para la línea de comandos:

```
StyleVision [<Archivo SPS>] [<opciones>]
```

donde

`StyleVision`	llama al ejecutable de Altova StyleVision, ubicado en la carpeta de aplicación de StyleVision.
`<Archivo SPS>`	es el archivo SPS.
`<opciones>`	como mínimo una de las opciones que se describen más abajo.

© 2016-2022 Altova GmbH

Cuando se ejecuta un comando, StyleVision se ejecuta en silencio (es decir, sin abrir la interfaz gráfica), genera los archivos de salida necesarios y se cierra. Si se detecta un error o advertencia, se abre la interfaz gráfica del usuario y el mensaje correspondiente aparece en la ventana de mensajes.

Nota: para que el SPS se cargue correctamente en StyleVision es necesario que los archivos XSD y XML de trabajo que usa el SPS se encuentren en la posición indicada en el SPS.

Opciones de StyleVision

No hace falta insertar las opciones de StyleVision en ningún orden en particular. Algunas opciones relacionadas con los formatos de salida FO, PDF y Word 2007+ están disponibles solo en la edición Enterprise y otras en las ediciones Enterprise y Professional.

- ***Archivo XSLT de salida***

`-OutXSLT=<archivo>`	escribe código XSLT para HTML en el archivo especificado
`-OutXSLRTF=<archivo>`	escribe código XSLT para RTF en el archivo especificado
`-OutXSLFO=<archivo>`	escribe código XSLT para FO en el archivo especificado *(solo en la edición Enterprise)*
`-OutXSLWord2007=<archivo>`	escribe código XSLT para Word 2007+ en el archivo especificado *(solo en la edición Enterprise)*

- ***Resultado de los datos de BD***

`-OutDBXML=<archivo>`	escribe los datos XML generados con la BD en el archivo especificado. Si se trata de una BD XML, el esquema fuente se puede especificar con la marca `-DBWhere`.
`-OutDBSchema=<archivo>`	escribe el esquema XML generado a partir de la BD en el archivo especificado. Si se trata de una BD XML, el esquema fuente se puede especificar con la marca `-DBWhere`.

Ejemplos

```
StyleVision "QuickStart.sps" -outxslt="QuickStartHTML.xslt"
StyleVision "C:\Test\QuickStart.sps" -outxslt="C:\Test\QuickStartHTML.xslt"
```

Notas

- Las rutas de acceso pueden ser absolutas o relativas y deben usar barras diagonales invertidas.
- Si el nombre de archivo o su ruta de acceso contiene un espacio, es necesario poner toda la ruta de acceso entre comillas. Por ejemplo: `"c:\Mis Archivos\MiXML.xml" or "c:\MisArchivos\MiXML.xml"`.
- La línea de comandos distingue entre el uso de mayúsculas y minúsculas en los comandos, rutas de acceso y nombres de carpeta y archivo.

13.1.2 StyleVision Server

StyleVision Server se puede utilizar desde su interfaz de la línea de comandos en sistemas Windows, Linux y Mac OS para transformar archivos XML en documentos HTML, PDF, RTF y DOCX de salida. El comando `generate` de la interfaz de la línea de comandos de StyleVision Server toma dos argumentos (un archivo XML y un archivo PXF) 512 y varios parámetros (los formatos de salida deseados). Las hojas de estilos XSLT para la transformación se obtienen del archivo PXF 512 utilizado como entrada.

Una ventaja de usar la interfaz de la línea de comandos de StyleVision Server en lugar de la de RaptorXML Server es que StyleVision Server acepta archivos PXF como entrada (mientras que RaptorXML toma archivos XSLT como entrada). Sin embargo, StyleVision Server debería utilizarse junto con el sistema Altova FlowForce Server preferiblemente. Los flujos de trabajo de Altova FlowForce Server pueden iniciar trabajos de transformación en función de desencadenadores predefinidos. Por ejemplo, se pueden transformar automáticamente los archivos de una red cuando se desencadena el trabajo de FlowForce. Para más información consulte el apartado Automatización con FlowForce Server 705.

Si necesita más información sobre la interfaz de la línea de comandos de StyleVision Server, consulte los archivos de ayuda de StyleVision Server.

Archivos de salida

StyleVision Server puede generar estos archivos de salida a partir de archivos PXF:

- Archivos HTML (`.html`) usando los archivos XML y XSLT para HTML indicados en el archivo PXF o cualquier otro archivo XML
- Archivo de texto usando los archivos de texto y XSLT para texto indicados en el archivo PXF
- Archivo RTF usando los archivos XML y XSLT para RTF indicados en el archivo PXF o cualquier otro archivo XML

© 2016-2022 Altova GmbH

13.2 Trabajar con RaptorXML

Altova RaptorXML es el procesador XML y XBRL de tercera generación ultrarápido 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 Edition**, que se puede usar desde la red y puede transformar varios archivos a la vez.
- **RaptorXML+XBRL Server Edition**, que se puede usar desde la red, puede transformar varios archivos a la vez y ofrece funciones de validación de datos XBRL.

Para más información sobre RaptorXML visite el sitio web de Altova.

Casos de uso

Las funciones de RaptorXML más importantes para los usuarios de StyleVision son las de transformación XSLT. El uso más común de estas funciones sería el siguiente:

1. El usuario general una hoja de estilos XSLT a partir del diseño SPS con ayuda del comando **Archivo | Guardar archivos generados** 782. Es importante tener en cuenta que RaptorXML no se puede usar para generar hojas de estilos XSLT a partir de un archivo SPS.
2. La hoja de estilos XSLT generada se utiliza para transformar documentos XML con RaptorXML. RaptorXML puede generar documentos de salida en formato HTML, texto y RTF.

Ventajas de RaptorXML

Estas son las principales ventajas de usar RaptorXML:

- RaptorXML permite validar y transformar código XSLT en un tiempo récord y por tanto es ideal para trabajar con archivos de gran tamaño.
- Fácil utilización con la línea de comandos y con las interfaces COM, Java y .NET.
- Automatización y programación de tareas 707 mediante archivos de procesamiento por lotes y procesos de programación.

Para aprender a automatizar la producción de documentos de salida (p. ej. HTML) a partir de documentos XML con RaptorXML, consulte el apartado Automatizar el procesamiento de datos 707.

Para obtener información detallada sobre RaptorXML y sus interfaces COM, Java y .NET consulte la documentación de RaptorXML.

13.2.1 Documentos PDF de salida

Para generar un documento PDF a partir de un documento XML son necesarios dos pasos:

1. Primero se transforma el documento XML con una hoja de estilos XSLT. Para esta transformación se usa un motor de transformación XSLT (como RaptorXML, por ejemplo). El resultado de la transformación es un documento FO.
2. El documento FO se procesa con un procesador FO (como Apache FOP) para generar un documento PDF de salida. StyleVision se puede configurar para que envíe el resultado FO de una transformación XSLT a un procesador FO. En StyleVision, el resultado PDF se puede ver en la vista previa PDF o se puede guardar como archivo (con el comando **Archivo | Guardar archivos generados**782).

RaptorXML y PDF

RaptorXML no ofrece parámetros para dirigir el resultado PDF de una transformación XSLT a un procesador PDF. Por tanto, tras el primer paso de transformación XSLT solo tenemos un documento FO.

Ahora debemos pasar este documento FO a un procesador FO para que realice el procesamiento del segundo paso (convertir FO en PDF). Dependiendo del procesador utilizado, esto se puede hacer de varias maneras. Por ejemplo, si utilizamos el procesador Apache FOP, podemos usar este sencillo comando para identificar el documento FO de entrada y especificar el nombre y ubicación del documento PDF de salida:

```
fop -fo entrada.fo -pdf salida.pdf
```

FOP ofrece más parámetros, que se enumeran en la referencia del usuario de FOP.

FOP y XSLT

Hay una opción FOP que sirve para especificar un archivo XML de entrada, un archivo XSLT de entrada y un archivo PDF de salida:

```
fop -xml entrada.xml -xslt entrada.xslt -pdf salida.pdf
```

En este caso, FOP usa su motor XSLT integrado para llevar a cabo la transformación XML-FO del primer paso. Después pasa el documento FO resultante a FOP para el procesamiento FO-PDF del segundo paso.

Sin embargo, puede que el motor integrado de FOP no sea compatible con todas las características de StyleVision y de RaptorXML. Es decir, puede que se produzcan errores si una hoja de estilos XSLT generada con StyleVision se usa como entrada para una transformación XML con el motor XSLT integrado de FOP. En casos así, utilice el motor XSLT de RaptorXML+XBRL Server para transformar el XML en FO y pase el archivo FO al procesador FOP para crear el PDF.

Procesamiento por lotes en formato PDF

Hay una manera muy rápida y sencilla de generar documentos PDF usando RaptorXML para la transformación XSLT del primer paso y FOP para el procesamiento FO del segundo paso. Se trata de escribir un archivo de procesamiento por lotes que combine ambos comandos. Por ejemplo:

```
raptorxmlserver xslt --input=Prueba.xml --output=Prueba.fo Prueba.xslt
fop -fo Prueba.fo -pdf Prueba.pdf
```

El primer comando llama a RaptorXML y produce el documento de salida `prueba.fo`. El segundo comando pasa `prueba.fo` al procesador FOP, que genera el archivo PDF `prueba.pdf`. Para más información sobre el procesamiento de datos por lotes y cómo usar archivos por lotes para automatizar procesos, consulte el apartado Automatizar el procesamiento de datos707.

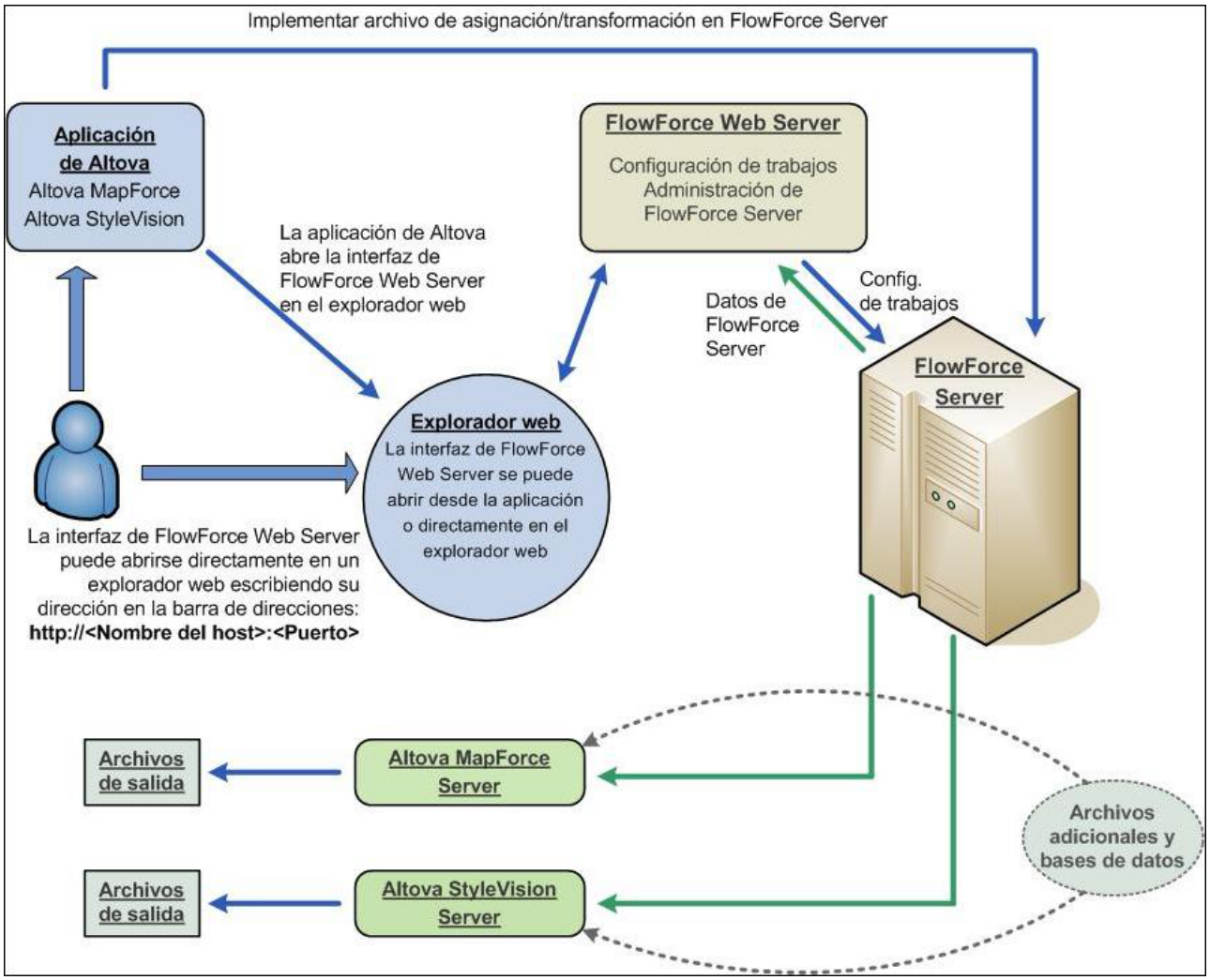
 © 2016-2022 Altova GmbH

13.3 Automatización con FlowForce Server

Las transformaciones se pueden automatizar en toda una red gracias a Altova FlowForce Server, un producto servidor para Windows, Linux y Mac OS. Así funciona el proceso de transformación en FlowForce Server:

1. Desde StyleVision, el usuario implementa en FlowForce Server 785 un archivo PXF 512 (con el comando **Archivo | Implementar en FlowForce** 785) como archivo de transformación con la extensión `.transformation`. El archivo `.transformation` contiene todos los archivos y toda la información necesaria para realizar las transformaciones definidas en el diseños SPS. En el diagrama que aparece más abajo, la línea que recorre la parte superior de la imagen representa este paso de implementación.
2. Tras implementar el archivo `.transformation` en FlowForce Server, puede crear trabajos en FlowForce que usen el archivo `.transformation` para realizar transformaciones según determinados desencadenadores creados en la definición del trabajo. Por ejemplo, el desencadenador puede desencadenar el trabajo a una hora concreta cada día. Los trabajos de FlowForce se crean en la interfaz web de FlowForce Server (que aparece en el centro del diagrama), a la que puede acceder desde StyleVision o a través de una dirección HTTP. Para más información sobre estos trabajos y desencadenadores, consulte la documentación de FlowForce Server.
3. En tiempo de ejecución FlowForce Server pasa las instrucciones de transformación y todos los archivos pertinentes a StyleVision Server, que se ocupa de realizar la transformación (*ver diagrama*).

El papel que desempeña StyleVision Server en el flujo de trabajo de FlowForce Server se ilustra en el diagrama que aparece a continuación. También aparece MapForce Server porque también se pueden crear trabajos de FlowForce que envíen asignaciones de datos de Altova MapForce a MapForce Server para que las ejecute.



Recuerde que además de invocarse desde un trabajo de FlowForce, StyleVision Server se puede invocar desde la línea de comandos. Para más información consulte la documentación de StyleVision Server.

© 2016-2022 Altova GmbH

13.4 Automatizar el procesamiento de datos

Un archivo de procesamiento por lotes (un archivo de texto guardado con la extensión `.bat`) incluye una secuencia de comandos que se ejecutan desde la línea de comandos. Cuando se ejecuta el archivo de procesamiento por lotes, se ejecutan uno por uno todos los comandos del archivo, empezando por el primero de ellos. Por tanto, los archivos de procesamiento por lotes son ideales:

- Para ejecutar una serie de comandos automáticamente (*ver ejemplo más abajo*).
- Para crear una cadena de comandos de procesamiento, cuando un comando necesita una entrada que es generada por un comando anterior. Por ejemplo, el archivo XML resultante de una transformación se puede usar como entrada para una transformación posterior (*ver ejemplo más abajo*).
- Para programar una secuencia de tareas que se debe ejecutar a una hora concreta.

Archivo de procesamiento por lotes con secuencia de comandos

La secuencia de comandos que se debe ejecutar debe introducirse así:

```
@ECHO OFF
CLS
StyleVision TestEN.sps -outxslt=HTML-EN.xslt
StyleVision TestDE.sps -outxslt=HTML-DE.xslt
StyleVision TestES.sps -outxslt=HTML-ES.xslt
```

Cuando se procesa el archivo por lotes, se ejecutan los comandos y se generan los archivos. El archivo por lotes del ejemplo anterior StyleVision para generar tres archivos XSLT a partir de un diseño SPS.

14 StyleVision en Visual Studio

StyleVision se puede integrar en el entorno IDE Microsoft Visual Studio 2012/2013/2015/2017/2019/2022, para que pueda combinar las avanzadas funciones de creación de archivos SPS con este avanzado entorno de desarrollo.

Esta sección describe:

- Cómo instalar 709 el complemento de StyleVision para Visual Studio.
- Diferencias 710 entre StyleVision para Visual Studio y la versión independiente.

© 2016-2022 Altova GmbH

14.1 Instalar el complemento de StyleVision

Siga estas instrucciones para instalar el complemento de StyleVision 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 StyleVision (ediciones Enterprise o Professional Edition). Si tiene instalada la versión 2022 o una más avanzada, debe instalar la versión de 64 bits de StyleVision.
3. Descargue y ejecute el paquete de integración de StyleVision. Este paquete está disponible en la página de descargas de StyleVision (ediciones Enterprise y Professional) de www.altova.com/es.

Tras instalar el paquete de integración podrá empezar a usar StyleVision en el entorno de Visual Studio.

Nota importante

Debe usar el paquete de integración correspondiente a su versión de StyleVision (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.

Temas relacionados

- Integración de StyleVision 1051

14.2 Diferencias entre StyleVision y StyleVision Visual Studio

Este apartado enumera las diferencias entre la versión de StyleVision para Visual Studio y la versión independiente de StyleVision.

Ayudantes de entradas (ventanas de herramientas en Visual Studio)

Los ayudantes de entrada de StyleVision están disponibles como ventanas de herramientas en Visual Studio. Para más información sobre los ayudantes de entrada y sobre la interfaz gráfica de StyleVision, consulte la sección Interfaz del usuario 35.) Tenga en cuenta que:

- puede arrastrar los ayudantes de entrada a cualquier posición del entorno de desarrollo.
- al hacer clic con el botón derecho en la pestaña de un ayudante de entrada aparece un menú con opciones para personalizar la interfaz. Con estas opciones puede convertir la ventana en flotante, acoplarla, ocultarla y ocultarla automáticamente.

Comandos de StyleVision como comandos de Visual Studio

Algunos comandos de StyleVision están presentes como comandos de Visual Studio en la interfaz gráfica de Visual Studio. Se trata de estos comandos:

- ***Deshacer, Rehacer:*** estos comandos de Visual Studio afectan a todas las acciones llevadas a cabo en el entorno de Visual Studio.
- ***Proyectos:*** los proyectos de StyleVision se tratan como proyectos de Visual Studio.
- ***Personalizar barras de herramientas y comandos:*** las pestañas *Barras de herramientas* y *Comandos* del cuadro de diálogo "Personalizar" (**Herramientas | Personalizar**) contienen tanto comandos de Visual Studio como de StyleVision.
- ***Vistas:*** en el menú **Vista**, el comando **StyleVision** contiene opciones para activar/desactivar los ayudantes de entrada y las ventanas laterales de StyleVision, así como para cambiar de una vista de edición a otra y activar/desactivar guías de edición.
- ***Ayuda de StyleVision:*** este menú de StyleVision aparece como submenú del menú **Ayuda** de Visual Studio.

Nota: en Visual Studio 2019 y versiones posteriores puede acceder a las funcionalidades de StyleVision desde el menú **Extensiones** de Visual Studio. En versiones anteriores estas funcionalidades se encuentran en los menús de nivel superior de Visual Studio.

 © 2016-2022 Altova GmbH

15 StyleVision en Eclipse

Eclipse es un marco de código abierto que integra diferentes tipos de aplicaciones distribuidas en forma de complementos. El complemento de StyleVision para Eclipse permite acceder a las funciones de StyleVision desde la plataforma Eclipse 2022-09, 2022-06, 2022-03, 2021-12 y está disponible para sistemas 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 para Eclipse e integrar StyleVision en Eclipse 712
- Perspectiva de StyleVision en Eclipse 719
- Otros puntos de entrada de StyleVision en Eclipse 722

15.1 Instalar el complemento de StyleVision en 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
- StyleVision 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 StyleVision 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 StyleVision 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 StyleVision.

Instalar el complemento de StyleVision para Eclipse

Puede integrar el complemento de StyleVision en Eclipse ante la instalación del paquete de integración de StyleVision. Para ello:

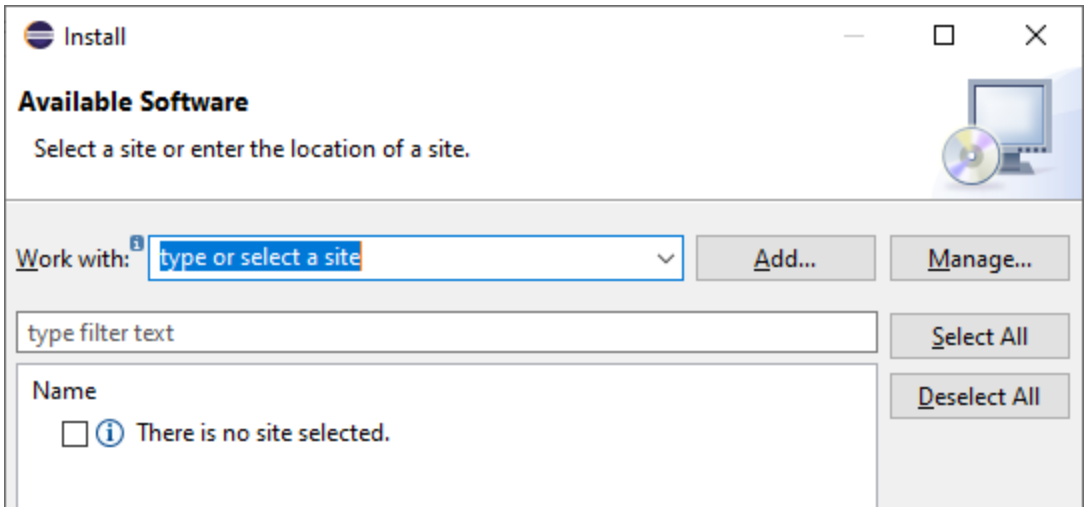
1. Ejecute el paquete de integración de StyleVision para iniciar el asistente de instalación.
2. Cuando el instalador se lo pida seleccione *Instalar el complemento para Eclipse* y haga clic en **Aceptar**.
3. Cuando le pida que escoja cómo quiere integrar el complemento de StyleVision en Eclipse seleccione *Permitir integración de StyleVision 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 StyleVision se habilitarán en Eclipse la siguiente vez que lo inicie.

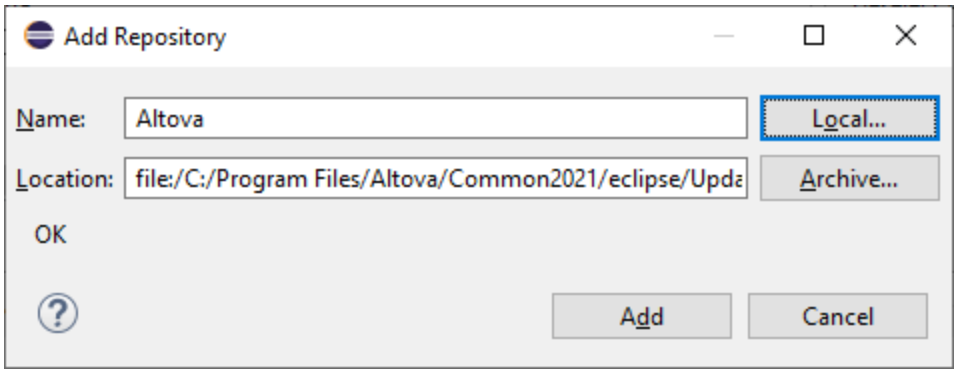
Integrar el complemento de StyleVision para Eclipse manualmente

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



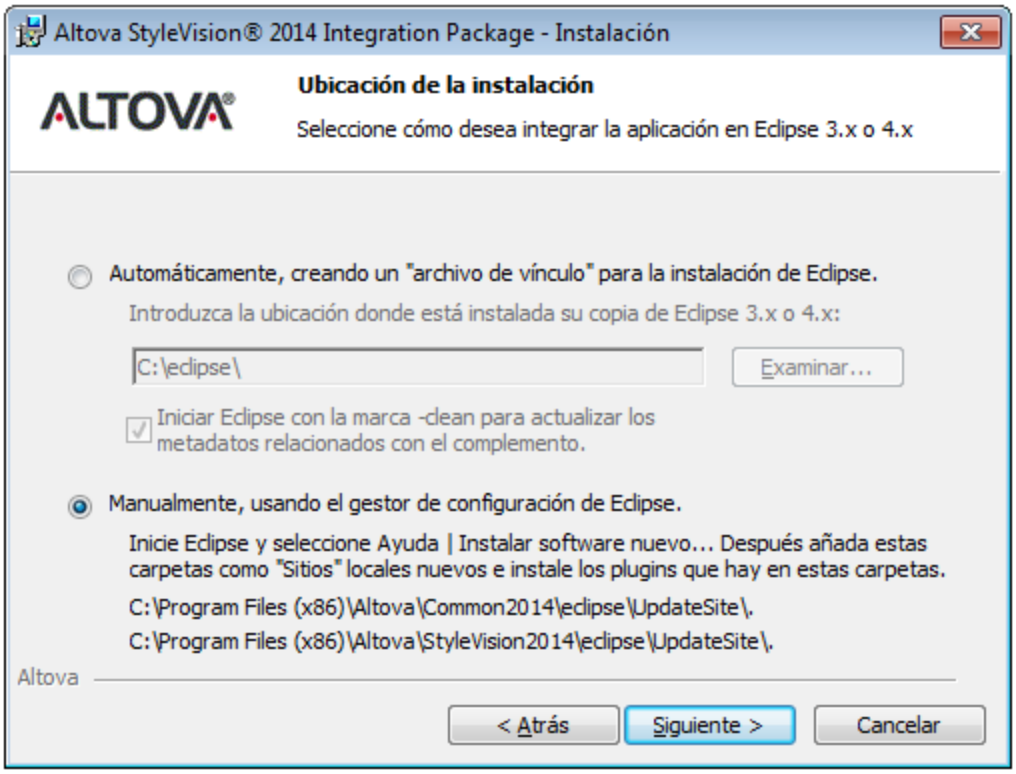
3. En el cuadro de diálogo "Agregar repositorio" haga clic en **Local**. Navegue hasta la carpeta `C:\Program Files\Altova\Common2023\eclipse\UpdateSite` y selecciónsela. Elija un nombre para el sitio (por ejemplo, "Altova").



4. Repita los pasos 2-3 del punto anterior pero esta vez elija la carpeta `C:\Program Files\Altova\StyleVision2023\eclipse\UpdateSite` y un nombre como "Altova StyleVision".
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`.

Recomendamos que elija la opción *Automáticamente* para que el asistente para la instalación se encargue de integrar el complemento. Seleccione la opción *Automáticamente* y después busque la carpeta donde se encuentra el ejecutable de Eclipse (`eclipse.exe`). Después haga clic en **Siguiente**. Si prefiere integrar el complemento de StyleVision para Eclipse de forma manual, seleccione la opción *De forma manual* (*imagen siguiente*). Más abajo explicamos cómo se integra el complemento de forma manual.



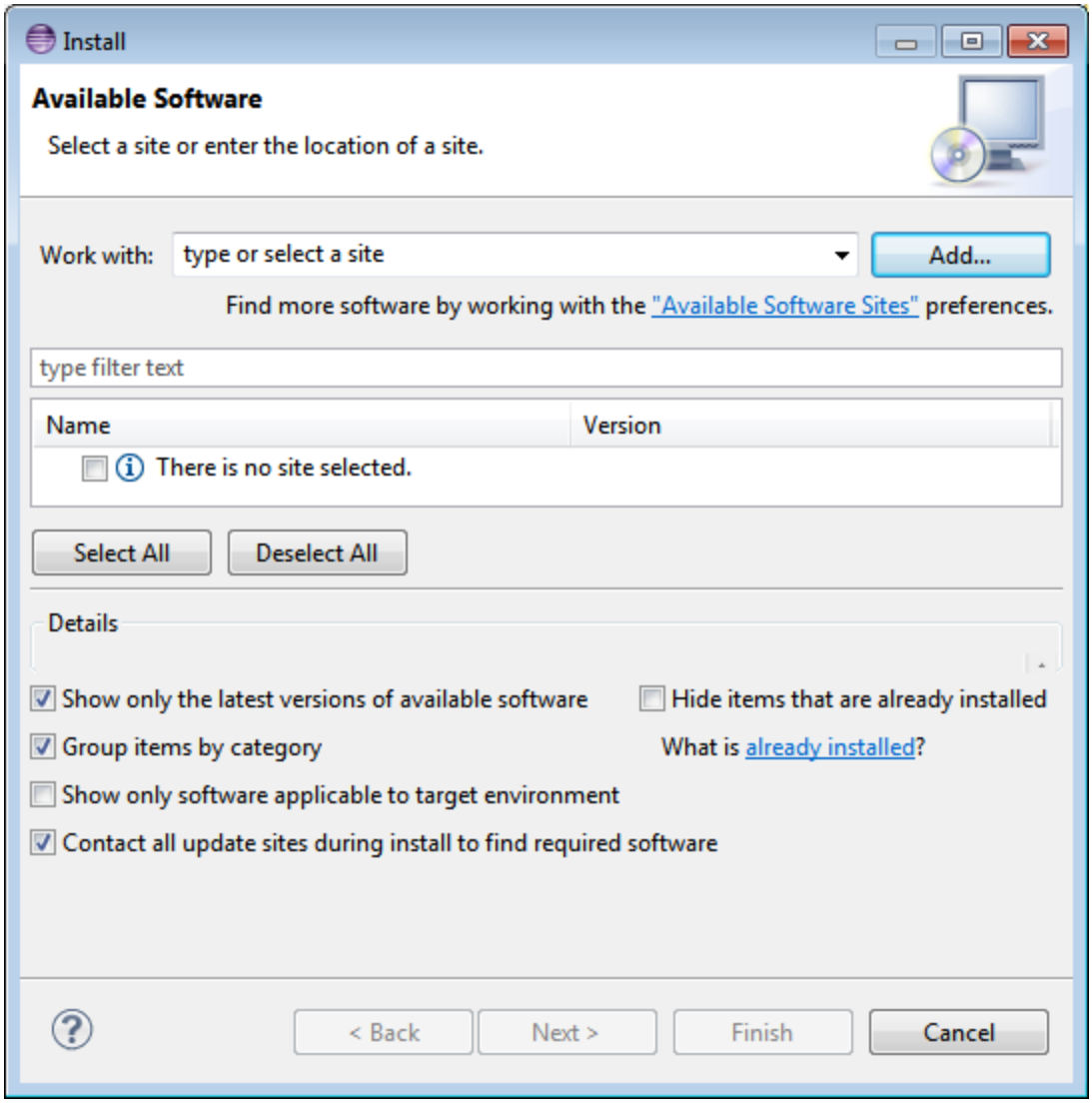
5. Finalice la instalación. Si seleccionó la integración automática, el complemento de StyleVision para Eclipse se integrará en Eclipse y estará disponible la próxima vez que inicie Eclipse.

Integrar el complemento de StyleVision en Eclipse de forma manual

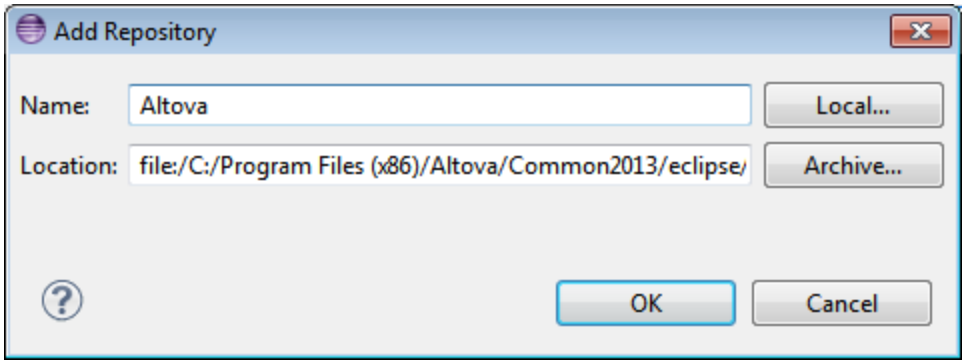
Para integrar el complemento de StyleVision en Eclipse de forma manual:

1. En Eclipse haga clic en el comando **Help | Install New Software** (Ayuda | Instalar software nuevo).
2. En el cuadro de diálogo que aparece (*imagen siguiente*) haga clic en el botón **Add** (Agregar).

© 2016-2022 Altova GmbH

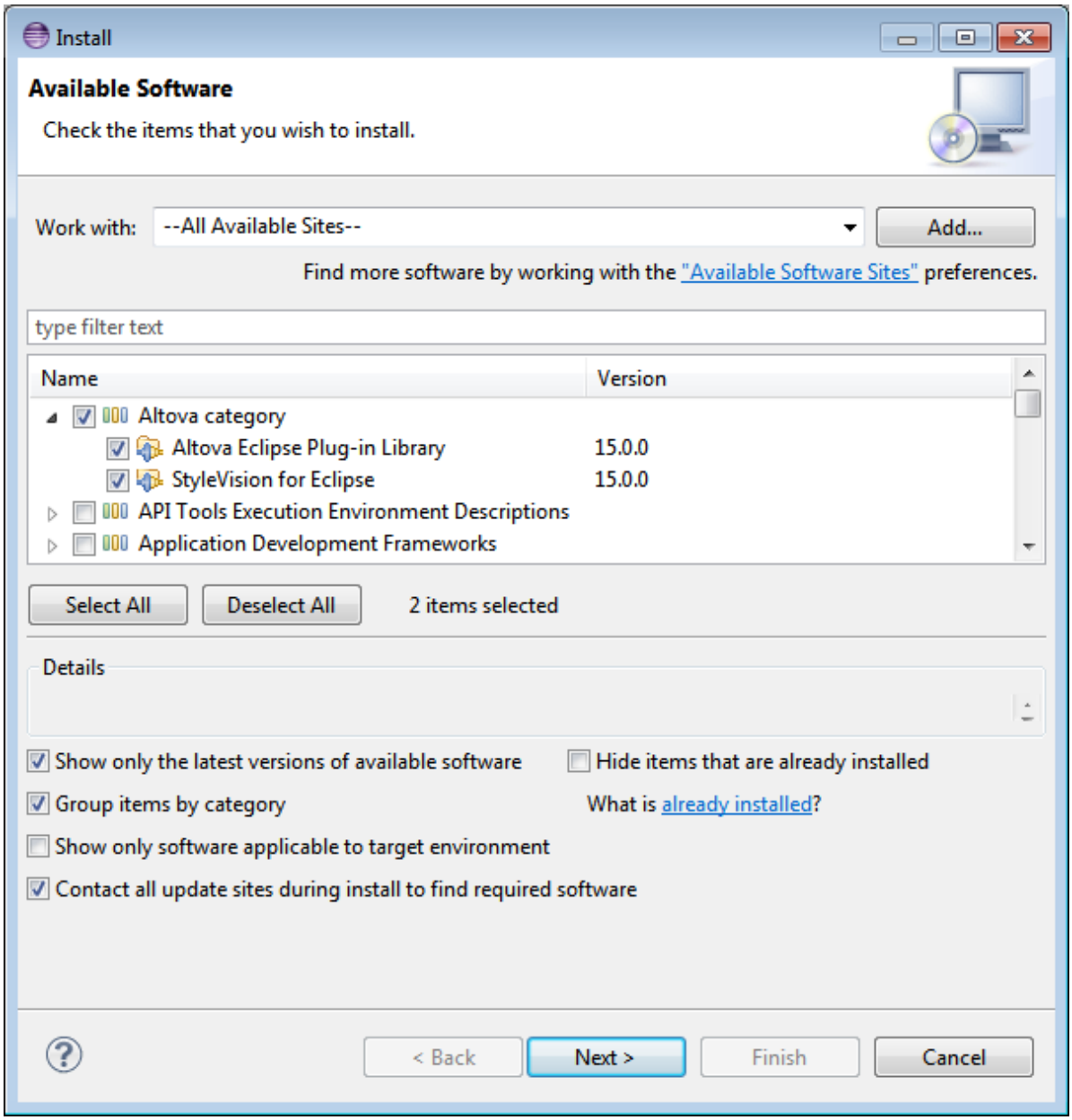


3. En el cuadro de diálogo que aparece, llamado "Add Repository" (Agregar repositorio, *imagen siguiente*) haga clic en el botón **Local**.
4. Busque la carpeta `c:\Archivos de programa\Altova\Common2023\eclipse\UpdateSite` y selecciónela. Escriba un nombre para el sitio (p. ej. Altova) y haga clic en **OK**.



5. Repita los pasos 2, 3 y 4 pero esta vez seleccione la carpeta `c:\Archivos de programa\Altova\StyleVision2023\eclipse\UpdateSite` y escriba un nombre como "Altova StyleVision".
6. En el cuadro combinado *Work With* (Trabajar con) del cuadro de diálogo "Install" seleccione la opción *All Available Sites* (Todos los sitios disponibles, *imagen siguiente*). Como resultado aparecen todos los complementos disponibles en el panel inferior. Marque la casilla de nivel superior de la carpeta *Altova category* (*imagen siguiente*). Después haga clic en **Next** (Siguiente).

© 2016-2022 Altova GmbH



7. Aparece una pantalla llamada "Install Details" (detalles de la instalación) en la que puede repasar los elementos que se van a instalar. Haga clic en **Next** (Siguiente) para continuar.
8. Aparece la pantalla "Review Licenses" (Repasar licencias). Seleccione la opción *I accept the terms of the license agreement* (Acepto el contrato de licencia). Para el complemento de StyleVision no es necesario otro contrato de licencia (basta con la clave de licencia de StyleVision Enterprise o Professional Edition). Para terminar haga clic en el botón **Finish** (Finalizar).

Si tiene problemas con el complemento (por ejemplo, si faltan iconos), inicie Eclipse desde la línea de comandos con la marca `-clean`.

Versión instalada actualmente

Para comprobar qué versión tiene instalada del complemento de StyleVision para Eclipse, en Eclipse seleccione la opción de menú **Help | About Eclipse**. Después seleccione el icono de StyleVision.

Temas relacionados

- Integración de StyleVision 1051

© 2016-2022 Altova GmbH

15.2 Puntos de entrada de StyleVision en Eclipse

En Eclipse una perspectiva es una vista de la interfaz del usuario configurada con función de operador. Cuando el complemento de StyleVision está integrado en Eclipse, se crea automáticamente una perspectiva StyleVision predeterminada. Esta perspectiva es una interfaz gráfica del usuario que incluye los elementos de la interfaz de StyleVision: sus vistas de edición, menús, ayudantes de entrada y otras barras laterales.

Al abrir un tipo de archivo asociado a StyleVision (por ejemplo, `.sps`), el archivo se puede editar en la perspectiva StyleVision. 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:

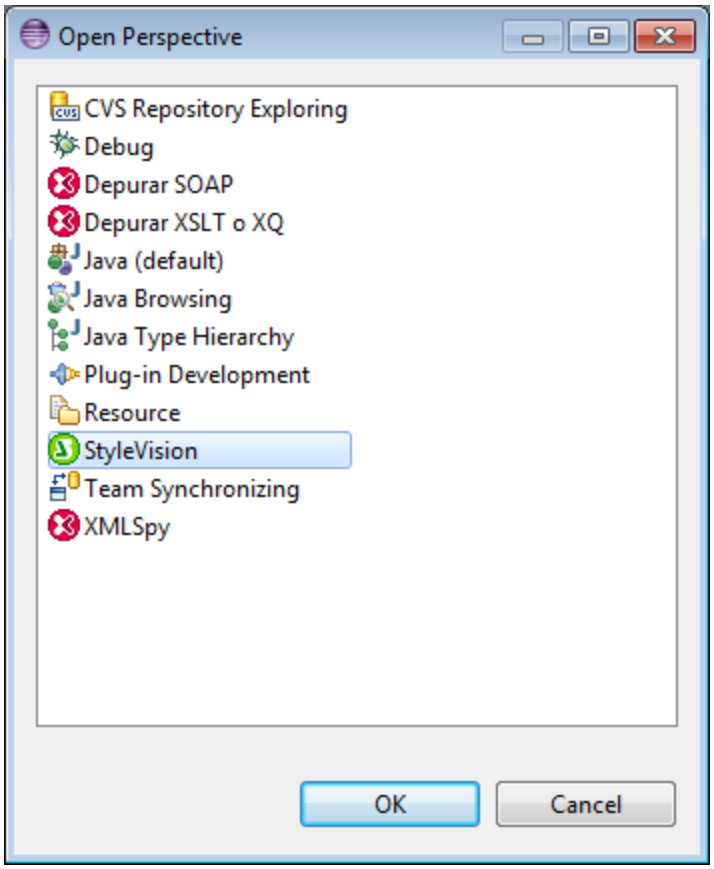
1. Permiten cambiar el entorno de trabajo para el archivo activo
2. Y permiten cambiar de un archivo a otro sin necesidad de abrir un entorno de desarrollo nuevo (el entorno asociado está disponible en la perspectiva)

Para trabajar con la perspectiva StyleVision es necesario:

- Cambiar a la perspectiva StyleVision.
- Configurar las preferencias de la perspectiva StyleVision.
- Personalizar la perspectiva StyleVision.

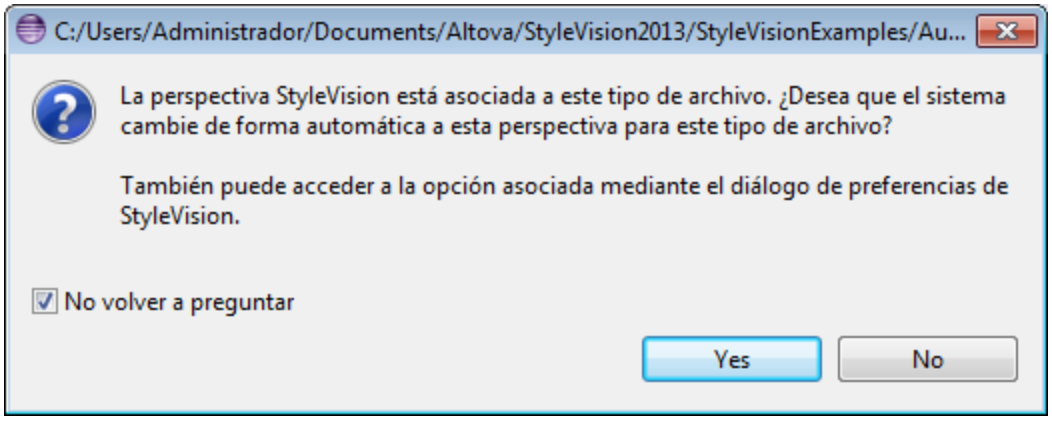
Cambiar a la perspectiva StyleVision

En Eclipse seleccione el comando **Window | Perspective | Open Perspective | Other**. En el diálogo que aparece (*imagen siguiente*) seleccione **StyleVision** y haga clic en **OK**.



La ventana vacía o el documento activo tendrán ahora la perspectiva StyleVision. 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.(*siguiente captura de pantalla*).



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 StyleVision

Para abrir el cuadro de diálogo de preferencias, seleccione el comando **Window | Preferences**. En la lista de perspectivas del panel de la izquierda, seleccione StyleVision 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 StyleVision
- Acceso a las opciones de StyleVision.

Personalizar la perspectiva StyleVision

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 StyleVision) 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.

15.3 Otros puntos de entrada de Stylevision en Eclipse

Además de la perspectiva de StyleVision existen otros dos puntos de entrada desde los que puede acceder a las funciones de StyleVision:

- El menú de StyleVision
- La barra de herramientas de StyleVision

El menú de StyleVision en Eclipse

El menú para Eclipse de **StyleVision** contiene comandos de StyleVision para acceder a las funciones correspondientes de StyleVision. Estos comandos aparecen en varios menús de la versión independiente de StyleVision.

Barra de herramientas de StyleVision en Eclipse

La barra de herramientas de StyleVision contiene estos botones:

Estos comandos sirven para:

- abrir la Ayuda de StyleVision
- acceder a los comandos de StyleVision (aunque también puede acceder a ellos desde el menú **StyleVision**)

Nota: los comandos de la barra de herramientas no son compatibles. Si ha configurado en StyleVision algún comando de la barra de herramientas que ejecute un comando o script, ese comando no estará disponible en el complemento.

© 2016-2022 Altova GmbH

16 Referencia del usuario

Esta sección de la documentación describe en detalle los menús, comandos y barras de herramientas de StyleVision, así como los iconos y símbolos de la vista **Diseño**. La referencia del usuario se divide en cuatro partes:

- La primera explica los símbolos e iconos utilizados en la vista 724 **Diseño** 724.
- La segunda describe el cuadro de diálogo Editar expresión XPath 729.
- La tercera describe todas las barras de herramientas y sus iconos 748 y explica cómo personalizarlas.
- La última parte describe uno a uno todos los menús de la aplicación y sus respectivos comandos.

El funcionamiento de las características de StyleVision no se describe en esta sección, sino en estos apartados del manual:

- Archivos SPS: contenido 134
- Archivos SPS: estructura 219
- Archivos SPS: características avanzadas 292
- Archivos SPS: presentación 374
- Archivos SPS: funciones adicionales 443
- Archivos SPS y bases de datos 518

16.1 Símbolos de la vista Diseño

Los diseños SPS suelen incluir varios tipos de componentes de diseño diferentes y en la vista **Diseño** cada componente se representa con un símbolo diferente. Hay símbolos para estos tipos de componentes:

- Nodos del documento XML 724
- Contenido del documento XML 725
- Dispositivos de entrada de datos 725
- Formatos predefinidos 726
- Objetos XPath 726
- Objetos URI 727

Todos estos componentes:

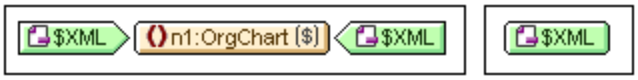
- se pueden mover mediante operaciones de arrastrar y colocar;
- se pueden cortar, copiar, pegar y eliminar usando (i) los comandos del menú **Edición** 794 o (ii) las teclas de acceso rápido estándar de Windows para estos comandos;
- se pueden modificar aplicando estilos y propiedades de formato;
- tienen un menú contextual que aparece al hacer clic con el botón derecho en el componente.

Nodos del documento XML

En la vista **Diseño** los nodos de elemento y atributo del documento XML se representan con etiquetas. Cada nodo tiene una etiqueta inicial y una etiqueta final. Al hacer doble clic en cualquiera de estas etiquetas, el nodo se contrae. Cuando el nodo está contraído su contenido se oculta. Al hacer doble clic en un nodo contraído, el nodo se expande y muestra su contenido.

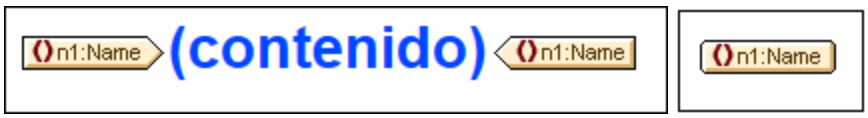
Los diferentes tipos de nodo que se representan en la vista **Diseño** son:

- ***Nodo de documento***



El **nodo de documento** (con la etiqueta `$XML`) representa el documento XML en su totalidad. Viene indicado con una etiqueta verde que dice `$XML` (si el esquema fuente está asociado a un documento XML) o `$DB` (si el esquema fuente está asociado a una BD). En la imagen anterior, el nodo de documento de la izquierda está expandido y contiene el elemento `OrgChart`, que está contraído. Por el contrario, el nodo de documento de la derecha está contraído y por eso no se puede ver su contenido.

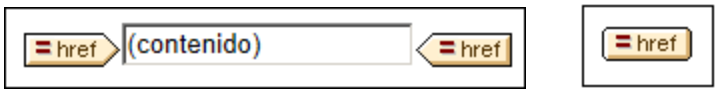
- ***Nodo de elemento***



Los **nodos de elemento** se insertan en el diseño junto con todos sus elementos antecesores (siempre que estos no están presentes ya en el punto de inserción). En la imagen anterior, por ejemplo, el nodo de elemento `Name` aparece expandido a la izquierda y contraído a la derecha.

© 2016-2022 Altova GmbH

- ***Nodo de atributo***



Los **nodos de atributo** se insertan en el diseño junto con todos sus elementos antecesores (siempre que estos no están presentes ya en el punto de inserción). Los nombres de atributo contienen el prefijo =. En la imagen anterior, por ejemplo, el nodo de atributo `href` aparece expandido a la izquierda y contraído a la derecha.

Los nodos se incluyen en el diseño como plantillas de nodo. Para más información sobre los diferentes tipos de plantillas que puede tener un diseño, consulte la sección Plantillas y fragmentos de diseño 265.

Contenido del documento XML

En la vista **Diseño** el contenido del documento XML se representa con dos marcadores de posición:

- `(contenido)`
- `(resto de contenido)`

El marcador de posición `(contenido)` representa el contenido de un solo nodo del documento XML. En el documento de salida se incluye todo el contenido de texto del nodo. Si el nodo es un atributo o un elemento de solo texto, en el documento de salida se incluye el valor del nodo. Si el nodo es un elemento que contiene contenido mixto o contenido de solo elemento, en el documento de salida se incluye el contenido de texto de todos los descendientes. Si hablamos en términos de XSLT, el marcador de posición `(contenido)` equivale al elemento `xsl:apply-templates` con su atributo `select` apuntando al nodo elegido.

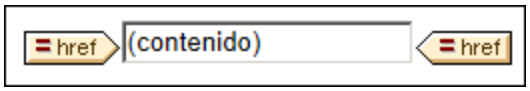
Nota: si se aplica a un nodo de elemento, el marcador de posición `(contenido)` no exporta los valores ni los atributos del elemento al documento de salida. Para incluir los nodos de atributo en el documento de salida, es necesario incluir explícitamente el atributo en la plantilla (principal o global).

El marcador de posición `(resto de contenido)` aplica plantillas al resto de los elementos secundarios del nodo actual. En este caso la plantilla que se aplica para cada elemento secundario será una plantilla global (si se definió alguna para el elemento) o la plantilla predeterminada para los elementos (que sencillamente exporta el texto de los elementos de solo texto y aplica plantillas a elementos secundarios). Por ejemplo, imagine que tiene un elemento llamado `libro` que tiene estos elementos secundarios: `título`, `autor`, `isbn` y `fechapublicación`. Si la definición de `libro` especifica que en el documento de salida solo se deben incluir los secundarios `título`, entonces cuando se procese esta definición no se exportarán los demás secundarios (`autor`, `isbn` ni `fechapublicación`). Si por el contrario, la definición de `libro` incluye el marcador de posición `(resto de contenido)` después de la definición del elemento `título`, entonces se aplicará una plantilla global (si existe para el elemento) o la plantilla predeterminada por cada uno de los elementos secundarios (`autor`, `isbn` y `fechapublicación`).

Dispositivos de entrada de datos

Si quiere, puede incluir dispositivos de entrada en el diseño para que el usuario de la vista Authentic pueda editar documentos XML correctamente. Para ello debe asignar uno de estos dispositivos de entrada a un nodo:

- ***Campos de entrada (de una sola línea o multilínea)***



- ***Cuadros combinados***

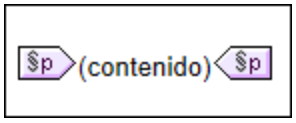
- ***Casillas de verificación***

- ***Botones de opción***

En la vista **Diseño** las etiquetas de los dispositivos de entrada de datos se pueden contraer o expandir haciendo doble clic en sus etiquetas. Para más información sobre el uso de estos componentes de diseño consulte el apartado Dispositivos de entrada de datos 190 del manual.

Formatos predefinidos

En la vista **Diseño** los formatos predefinidos se representan con etiquetas de color malva, que se pueden expandir o contraer haciendo doble clic en ellas.



Por ejemplo, la imagen anterior de la izquierda muestra el formato predefinido `p (párrafo)` expandido. La de la derecha muestra el mismo formato predefinido, esta vez contraído (con su contenido oculto). Para aplicar un formato predefinido, resalte la parte del diseño a la que desea aplicar el formato e inserte el formato predefinido 375.

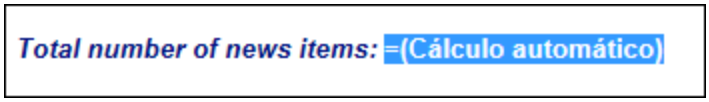
Objetos XPath

En StyleVision hay dos funciones que utilizan expresiones XPath:

- ***Plantillas condicionales***

En la vista **Diseño** las etiquetas de la **condición** son de color azul. La etiqueta inicial contiene algunas celdas: la primera celda (empezando por la izquierda) incluye un signo de interrogación. Las demás contienen (i) un número (empezando por el número 1) por cada condición `when` o (ii) un asterisco para la condición opcional `otherwise`. Para seleccionar una rama de la condición basta con hacer clic en su celda. El número de la rama condicional seleccionada aparece resaltado en la etiqueta inicial y la plantilla de la condición seleccionada (entre las etiquetas de la condición) aparece en la vista. La expresión XPath para la rama condicional seleccionada también aparece resaltada en la ventana Estructura del diseño. Las etiquetas de las condiciones no se pueden expandir ni contraer.

- ***Cálculos automáticos***



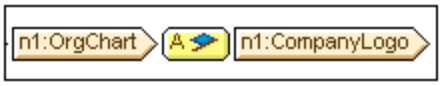
En la vista **Diseño** los **cálculos automáticos** se representan con el marcador de posición `=(Cálculo automático)` (*imagen anterior*). Al seleccionar un cálculo automático, su expresión XPath aparece resaltada en la ventana Estructura del diseño. Los cálculos automáticos se editan desde la ventana Propiedades 293.

Objetos URI

En los diseños SPS puede insertar tres tipos de objetos URI:

- ***Imágenes***
  Si inserta una imagen en el diseño SPS y StyleVision puede acceder a ella, entonces la imagen aparece en la vista **Diseño**. Si StyleVision no puede acceder a ella, la posición de la imagen en el diseño se señala con un marcador de posición.

- ***Marcadores (anclajes)***



En la vista **Diseño** los marcadores se representan con etiquetas de color amarillo que tienen la letra `A` (*imágenes anteriores*). Los marcadores se crean con el comando **Insertar | Insertar marcador** y pueden estar vacíos o tener contenido. El contenido debe insertarse una vez creado el anclaje. Las etiquetas de anclaje se pueden expandir (*imagen anterior, izquierda*) o contraer (*imagen anterior, derecha*).

- ***Vínculos***

En la vista **Diseño** los vínculos se representan con etiquetas de color amarillo que tienen la letra `A` (*imágenes anteriores*). Los vínculos se crean con el comando **Insertar | Insertar hipervínculo**. El objeto alrededor del cual se crea el vínculo puede insertarse en el diseño antes o después de crear el hipervínculo. Las etiquetas de un hipervínculo se pueden expandir (*imagen anterior, izquierda*) o contraer (*imagen anterior, derecha*).

Temas relacionados

- Barras de herramientas 748
- Barras laterales de la vista Diseño 41
- Edición de contenidos 134

© 2016-2022 Altova GmbH

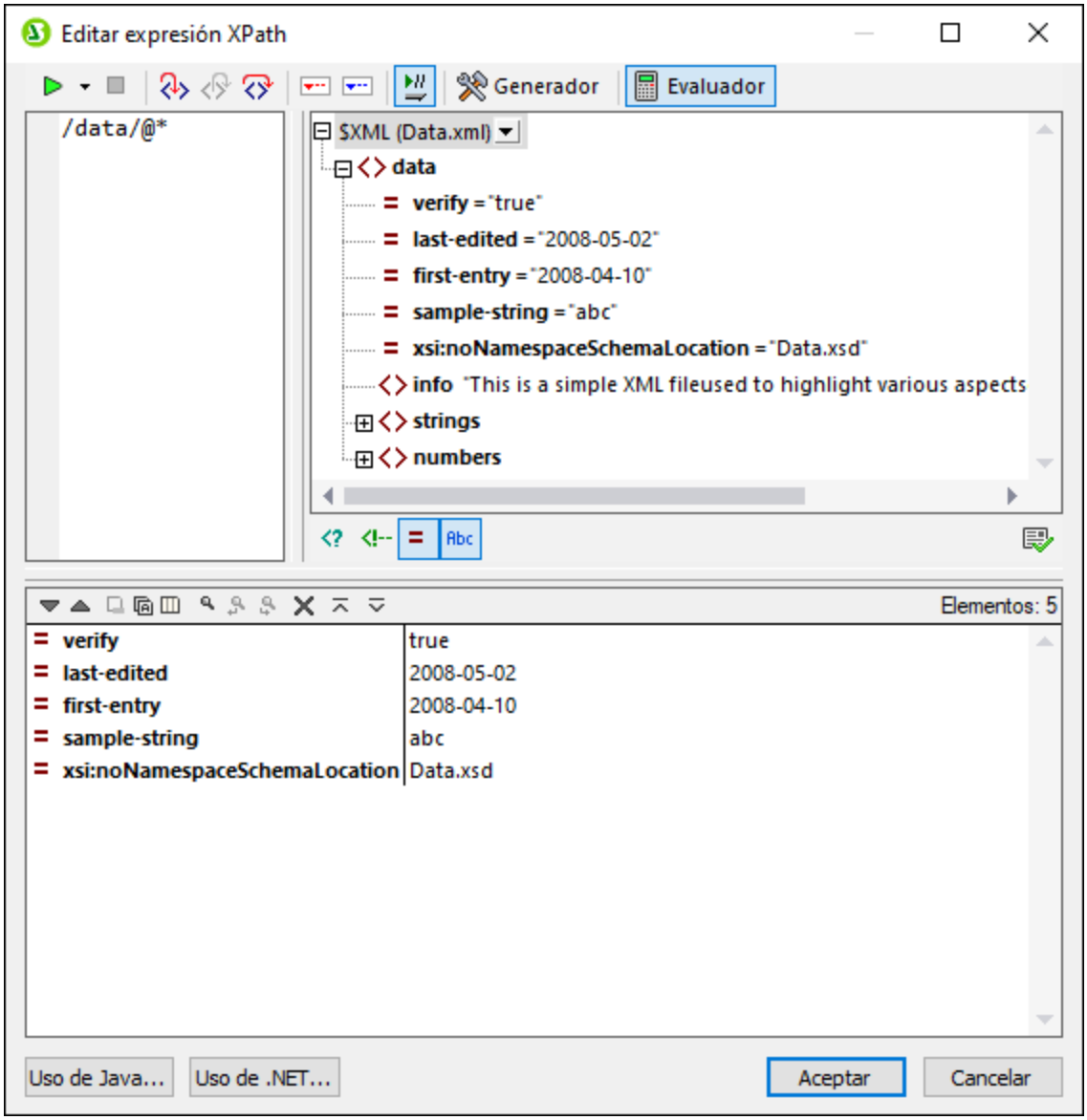
16.2 Diálogo Editar expresión XPath

El cuadro de diálogo "Editar expresión XPath" (*imagen siguiente*) sirve para construir, probar y editar las expresiones XPath. Se puede acceder a este cuadro de diálogo desde cualquier sitio de la Vista Diseño donde se pueda introducir una expresión XPath, como las que se introducen para el procesamiento condicional 303 o los valores de los Estilos 58 y Propiedades 59.

El cuadro de diálogo es compatible automáticamente con la versión XPath correspondiente a la versión XSLT 115 del diseño SPS (XPath 1.0 para XSLT 1.0, XPath 2.0 para XSLT 2.0 y XPath 3.1 para XSLT 3.0). Para cambiar a otra versión XPath debe cambiar la versión XSLT del diseño SPS 115.

Diseño del cuadro de diálogo

El cuadro de diálogo "Editar expresión XPath" contiene varios paneles (*imagen siguiente*): (i) un panel Expresión (*parte superior izquierda*), (ii) un panel *Fuentes* (*parte superior derecha*) y (iii) un panel *Resultados*. En el modo Generador el panel *Resultados* se amplía con otros paneles de ayudantes de entrada.



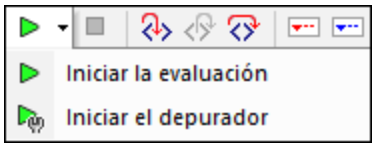
Modo Evaluador y modo Depurador

El cuadro de diálogo "Editar expresión XPath" tiene dos modos:

- Modo Evaluador 731, en el que se evalúa una una expresión XPath con respecto a los archivos XML de trabajo. Esta expresión se introduce en el panel *Expresión* y el resultado aparece en el panel *Resultados*. Puede hacer clic en nodos del resultado para ir a los nodos correspondientes en el panel *Fuentes* del cuadro de diálogo.
- Modo Depurador 735, en el que puede depurar una expresión XPath 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.

© 2016-2022 Altova GmbH

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*).



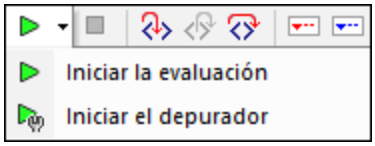
En los apartados de esta sección se explica cómo usar cada uno de estos dos modos.

Generador de expresiones XPath/XQuery

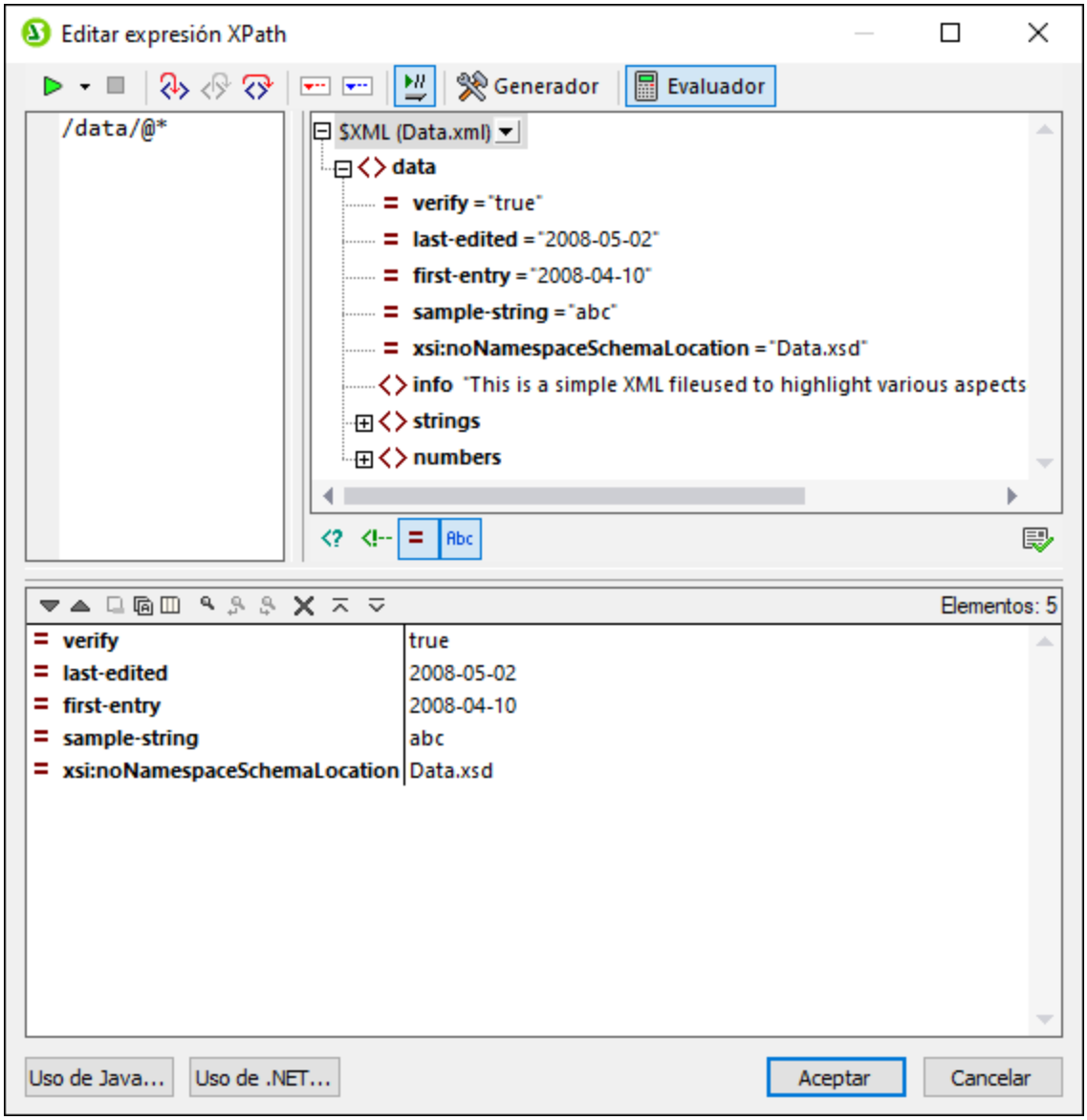
En los dos modos se puede usar el generador de expresiones 744 como ayuda para construir expresiones con una sintaxis correcta. Active o desactive el generador de expresiones 744 con el botón **Cambiar al modo generador** de la barra de herramientas principal .

16.2.1 Modo Evaluador

Seleccione el modo Evaluador; para ello seleccione **Iniciar evaluación** en el menú desplegable **Iniciar evaluación/depuración** (*imagen siguiente*).



En el modo Evaluador haga clic en el botón **Evaluador** (*imagen siguiente*). El evaluador se compone de varios paneles (*imagen siguiente*): (i) el panel *Expresión* (*parte superior izquierda*), (ii) el panel *Fuentes* y (iii) el panel *Resultados* (*parte inferior*).



La expresión XPath y su evaluación

La expresión XPath se introduce en el panel *Expresión*. Los resultados de la evaluación aparecen en el panel *Resultados* (*imagen anterior*).

Tenga en cuenta que:

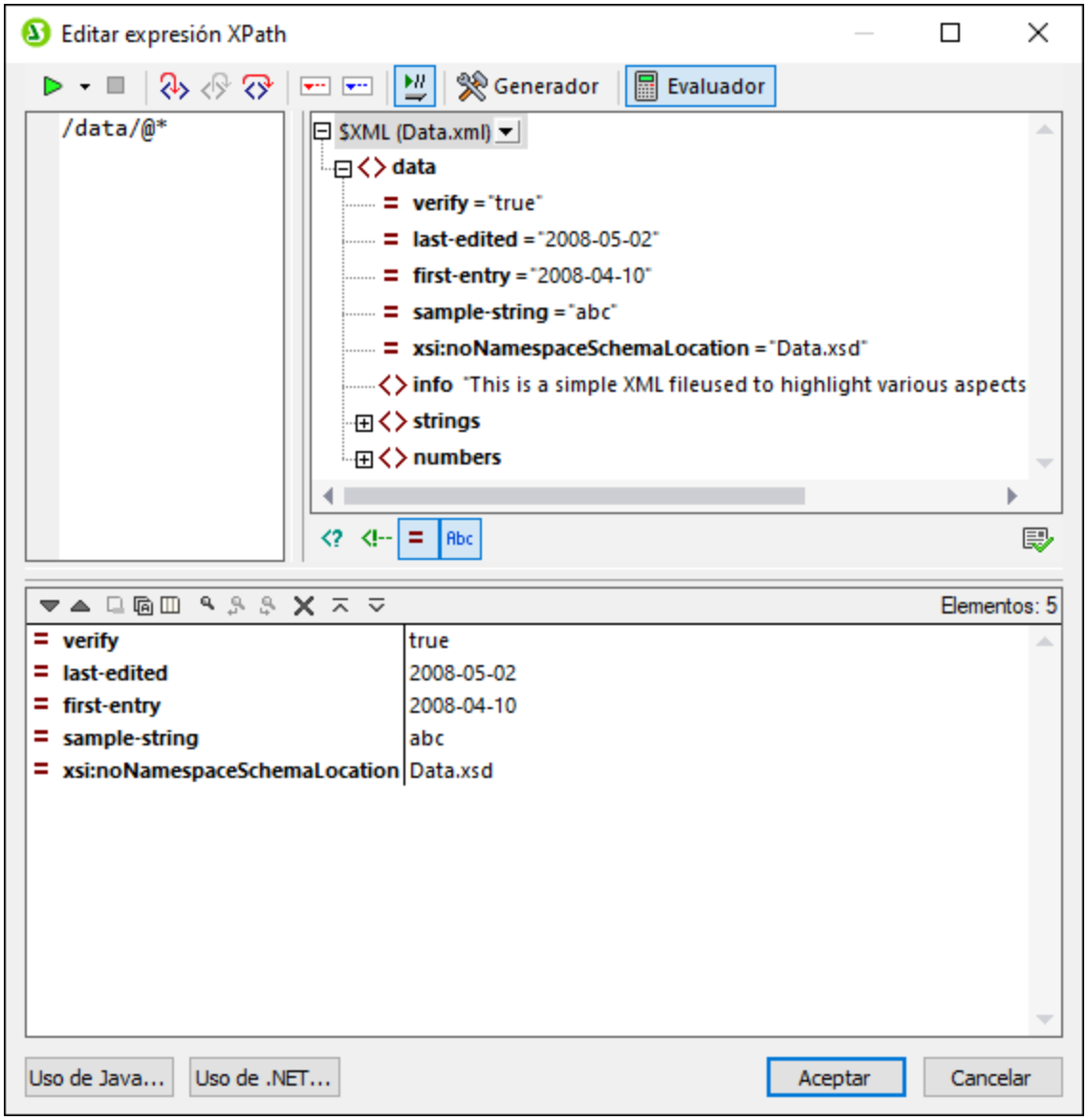
- para evaluar una expresión con respecto a un archivo XML, ese archivo debe estar asignado como Archivo XML de trabajo de uno de los orígenes del diseño SPS 113.
- puede ir viendo los resultados a medida que va escribiendo la expresión (use el icono **Evaluar mientras se teclea** de la barra de herramientas) o verlos cuando haga clic en el botón de la barra de herramientas **Iniciar evaluación/depuración (F5)**, en la parte superior izquierda de la barra de herramientas.

 © 2016-2022 Altova GmbH

- para introducir la expresión XPath de ruta en un árbol de fuentes de nodo haga doble clic en ese nodo en el panel *Fuentes*.
- en el panel *Fuentes* puede activar o desactivar que se muestren: (i) instrucciones de procesamiento, (ii) comentarios, (iii) atributos y (iv) elementos. Para ello use los botones que hay bajo el panel *Fuentes*.
- el nodo de contexto es el del componente del diseño dentro del cual se va a crear la expresión. Para definir otro nodo como el nodo de contexto de la expresión: (i) seleccione el nodo en el panel *Fuentes* y (ii) haga clic en **Definir contexto** (*bajo el panel Fuentes*). Para guardar este nodo de contexto para la expresión haga clic en el botón del comando **Recordar contexto de evaluación** (*bajo el panel Fuentes*). Tenga en cuenta que el nodo de contexto de la expresión es el nodo de contexto del componente de diseño actual y este es el nodo que se usará en tiempo de ejecución.
- puede usar las funciones de los lenguajes de programación Java y .NET en la expresión XPath. Los botones **Uso de Java** y **Uso de .NET**, en la parte inferior del cuadro de diálogo muestran información sobre cómo usar las funciones de extensión de Java y .NET en expresiones XPath. Para más información consulte la sección Funciones de extensión 1114 de esta documentación.
- para crear la expresión en varias líneas (para una lectura más cómoda) use la tecla **Entrar**.
- para aumentar/reducir el tamaño del texto en el campo de la expresión haga clic en él, mantenga pulsada la tecla **Ctrl** y accione la rueda de desplazamiento. ***Tenga en cuenta que este cambio afecta también al panel Resultados***.
- si no quiere introducir manualmente la ruta de localización de un nodo siga estas instrucciones: (i) colocar el cursor del ratón en el punto de la expresión donde quiere introducir la ruta de localización; (ii) en el árbol de fuentes haga doble clic en el nodo que quiere usar como destino. Esto introduce 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.

Panel Resultados

El panel *Resultados* (*panel inferior de la imagen siguiente*) tiene su propia barra de herramientas.



Estas son las características de este panel:

- la lista de resultados consiste en dos columnas: (i) un nombre o tipo de datos de nodo y (ii) el contenido del nodo.
- si la expresión XPath devuelve nodos (como elementos o atributos) puede elegir si quiere que se muestre todo el contenido del nodo como el valor del nodo. Para ello active la opción *Mostrar resultado completo*.
- cuando el resultado contiene un nodo (esto incluye nodos de texto), en lugar de contener literales generados por la expresión, al hacer clic en ese modo en el panel *Resultados* se resalta el nodo correspondiente en el documento XML del árbol de fuentes.
- puede copiar las dos columnas de una sublínea del resultado o copiar solamente la columna del valor. Para copiar todas las columnas haga clic con el botón derecho en una sublínea y active la opción **La copia incluye todas las columnas**. (También puede activar/desactivar este comando con el icono correspondiente del panel *Resultados*.) Después haga clic con el botón derecho en la sublínea que

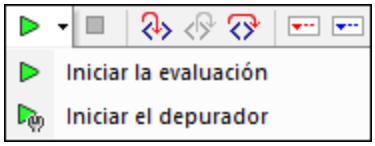
quiere copiar y seleccione **Copiar sublíneas** para copiar solamente esa o **Copiar todo** para copiarlas todas.

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 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.

16.2.2 Modo Depurador

Con el Depurador puede depurar expresiones XPath en el contexto de un archivo XML de trabajo 113. Para acceder al Depurador seleccione **Iniciar el depurador** en el menú desplegable **Iniciar evaluación/depuración** (*imagen siguiente*). Así se activa el modo Depurador. Ahora puede alternar entre el modo Generador (como ayuda para construir la expresión) y el Evaluador (para depurar la expresión). Para iniciar la depuración haga clic en **Iniciar evaluación/depuración (F5)**. Asegúrese primero de que el modo seleccionado es el Depurador.



Para iniciar la depuración haga clic en **Iniciar evaluación/depuración (F5)**. Asegúrese primero de que el modo seleccionado es el Depurador.

Botones de configuración del modo Depurador

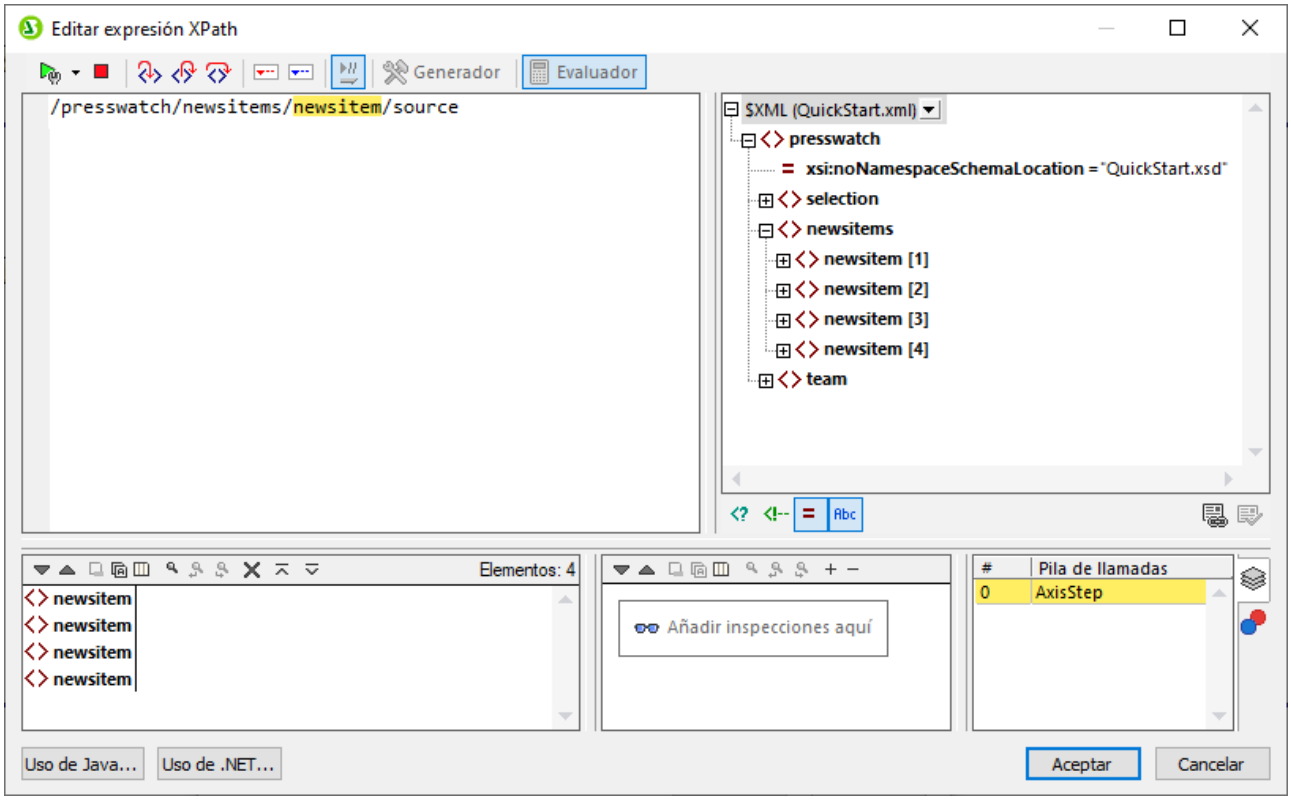
	Iniciar evaluación/depuración (F5)	Inicia el depurador
	Cambiar al Generador	Cambia al modo Generador, que cuenta con ayudantes de entrada contextuales para construir expresiones
	Evaluar al teclear	Evalúa las expresiones conforme las va escribiendo

Diseño del modo Depurador

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.

© 2016-2022 Altova GmbH



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.
- 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.

Ejecutar el depurador

A grandes rasgos, estos suelen ser los pasos que se siguen para depurar una expresión XPath:

1. Introducir la expresión XPath en el panel *Expresión*.
2. 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.
3. 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.
4. 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.

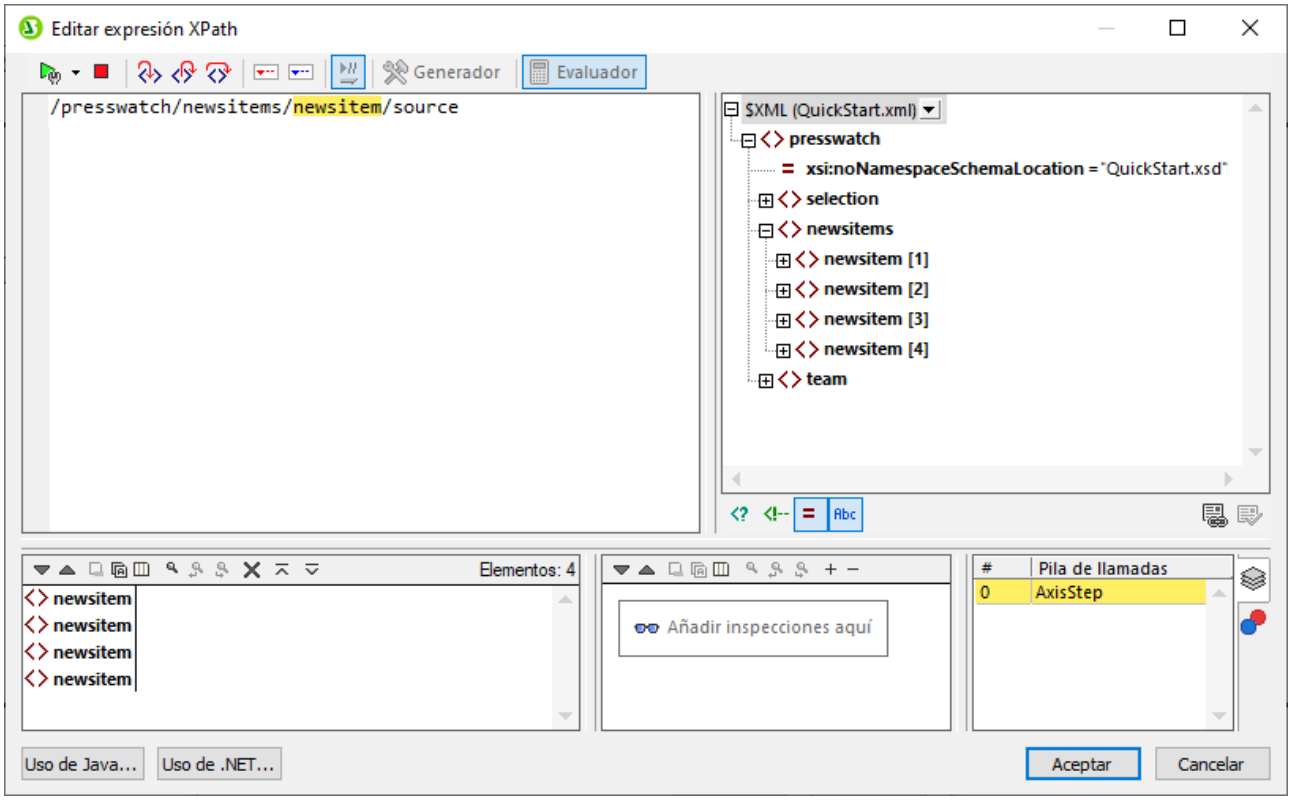
Botones para depurar

	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 usar la expresión de más abajo y hacer clic en los tres botones de este comando para ver cómo funcionan.

La imagen siguiente muestra la evaluación cuando el procesamiento se ha pausado al alcanzar el paso `newsitem`. En este paso el resultado muestra los cuatro nodos **`newsitem`**.



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)**.

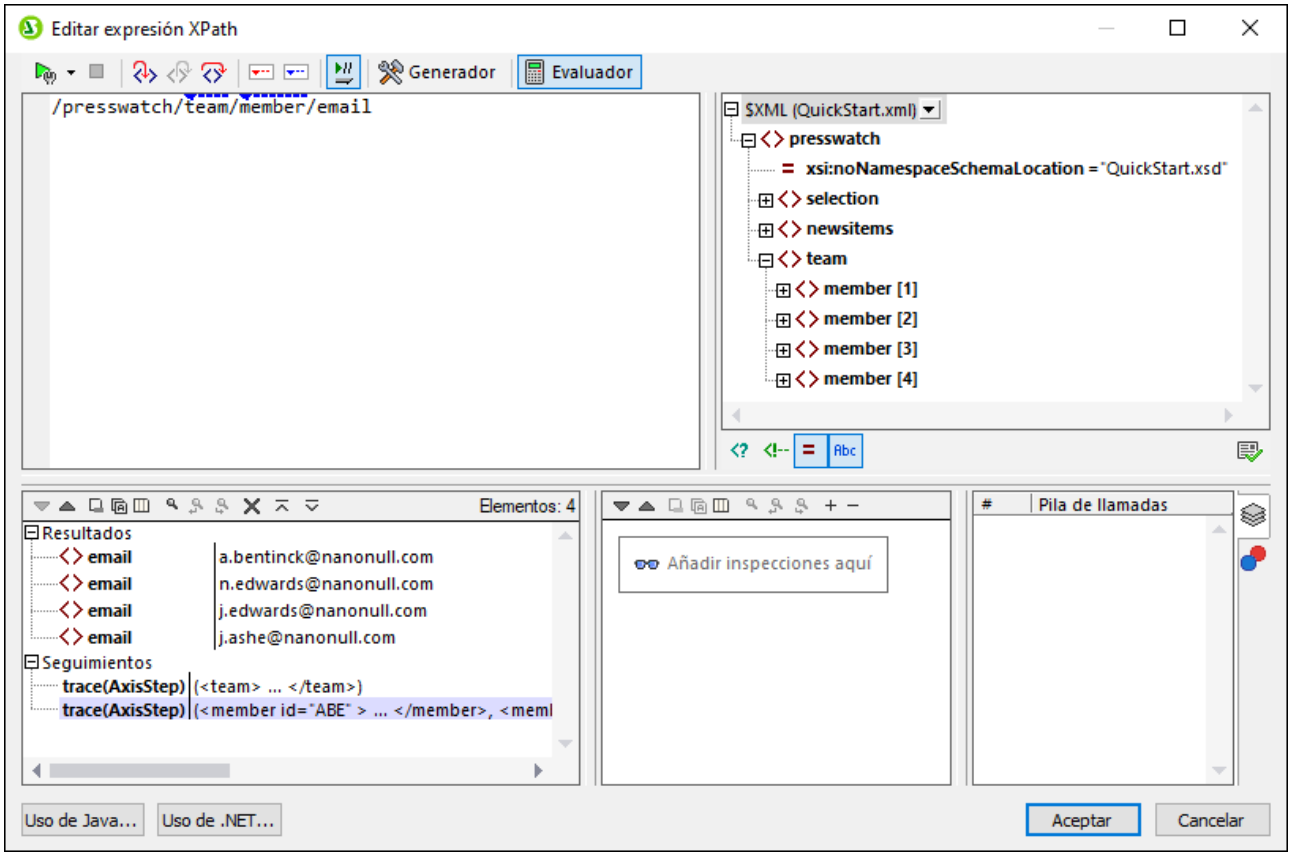
Véase también Puntos de depuración 742, más abajo.

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 los nodos `team` y `member`; los resultados que corresponden a estos puntos de seguimiento aparecen en la estructura *Seguimientos*.

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)**. Para marcar el punto de

seguimiento aparece una línea discontinua azul sobre esa parte de la expresión (*imagen siguiente*). Para quitar un punto de seguimiento basta con seleccionarlo y volver a hacer clic en **Insertar o quitar punto de seguimiento (Mayús+F9)**.



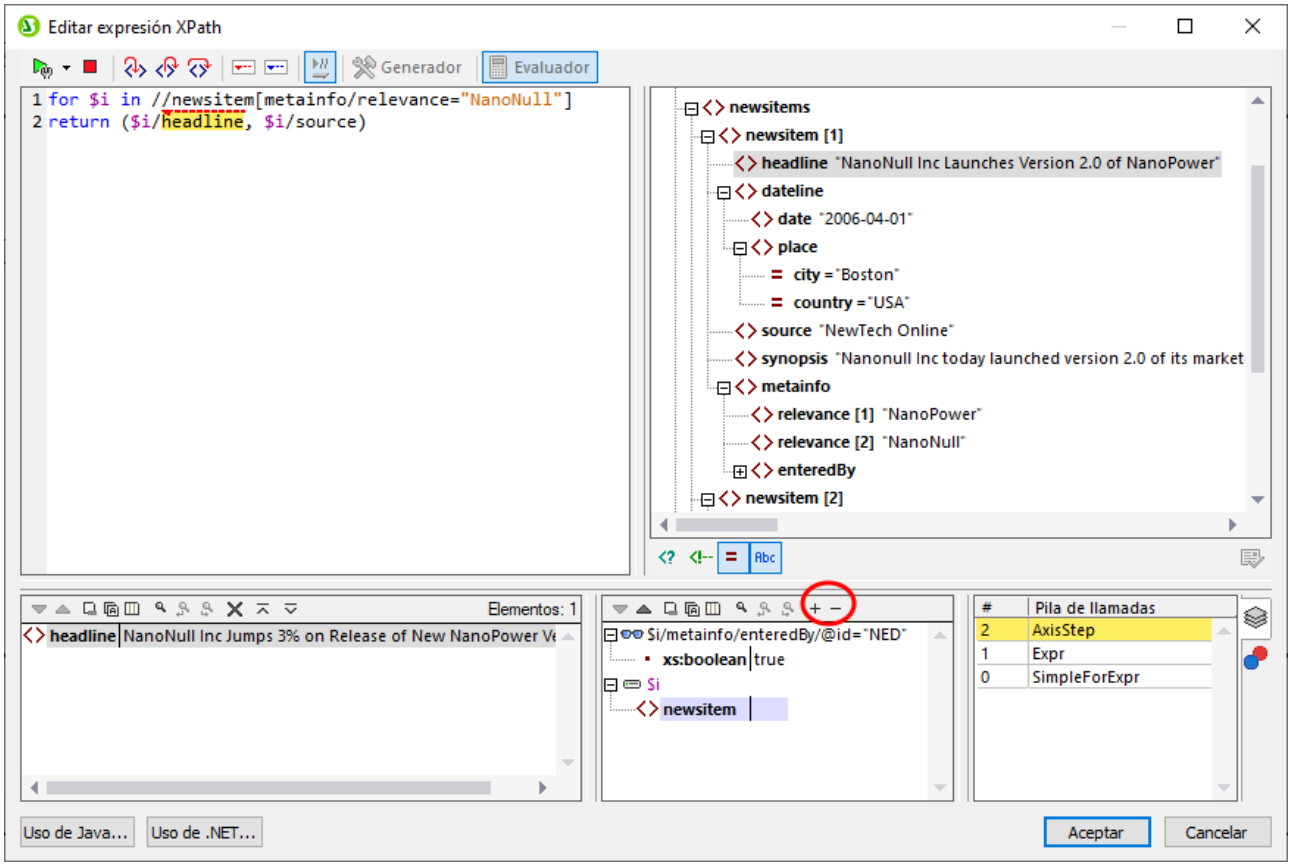
Nota: si en algún punto de la expresión hay un punto de interrupción y un punto se seguimiento, entonces la línea discontinua alterna los colores rojo y azul.

Véase también Puntos de depuración 742, 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 actuales correspondientes. Por ejemplo, en la imagen anterior el procesamiento se ha pausado en el punto de interrupción que hay en el elemento **`headline`**. La variable **`$i`** está dentro del alcance en este paso de la evaluación, por lo que **`$i`** se muestra con su valor actual, que en la imagen anterior es el primer nodo **`newsitem`**.

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/metainfo/enteredBy/@id="NED"`** se usa para probar si este elemento lo introdujo el miembro del equipo que tiene el identificador **`NED`**. El resultado **`true`** en el caso del primer elemento de las noticias nos comunica que ese elemento cumple esta condición.
- 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, se puede generar una cadena de texto apropiada para indicar el número total de elementos nuevos.

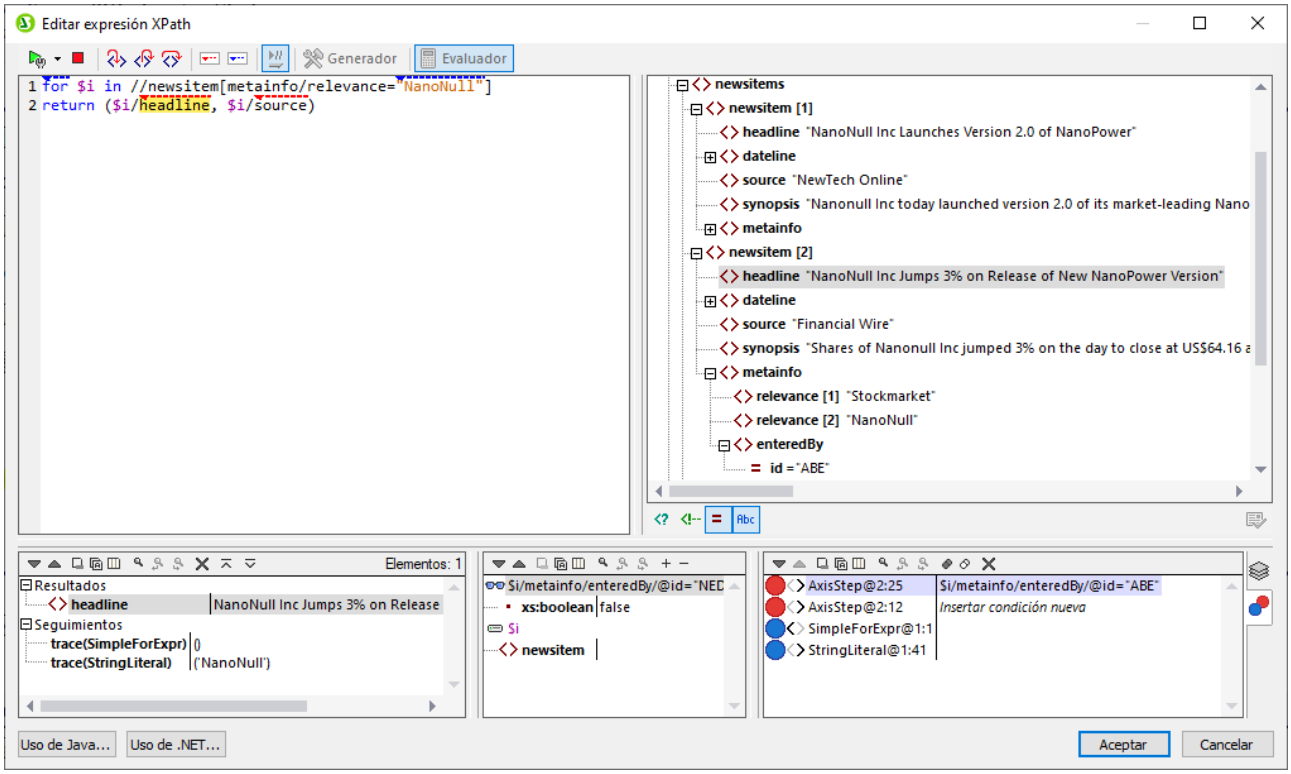
Para introducir una expresión de inspección haga clic en **Agregar entrada de inspección** en la barra de herramientas del panel (*dentro del círculo rojo en la imagen anterior*), introduzca la expresión y pulse la tecla

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.

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 (puntos 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, **`AxisStep@2:12`** significa que hay un punto de depuración en la línea 2, carácter 12 de la expresión del panel *Expresión*.



Algunas de las funciones que ofrece este modo son:

- En el caso de los puntos de interrupción, puede introducir una condición de interrupción (i) haciendo doble clic en *Insertar condición de interrupción* en el panel *Puntos de depuración*, (ii) introduciendo la expresión de la condición y (iii) pulsando 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/metainfo/enteredBy/@id="ABE"`** habilita el punto de interrupción en el encabezado de los elementos de las noticias que haya introducido el miembro del equipo que tiene el identificador **`ABE`**. La

imagen muestra la evaluación pausada en este punto. (Tenga en cuenta que la expresión de inspección de este punto de interrupción devuelve `false`.)

- 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.

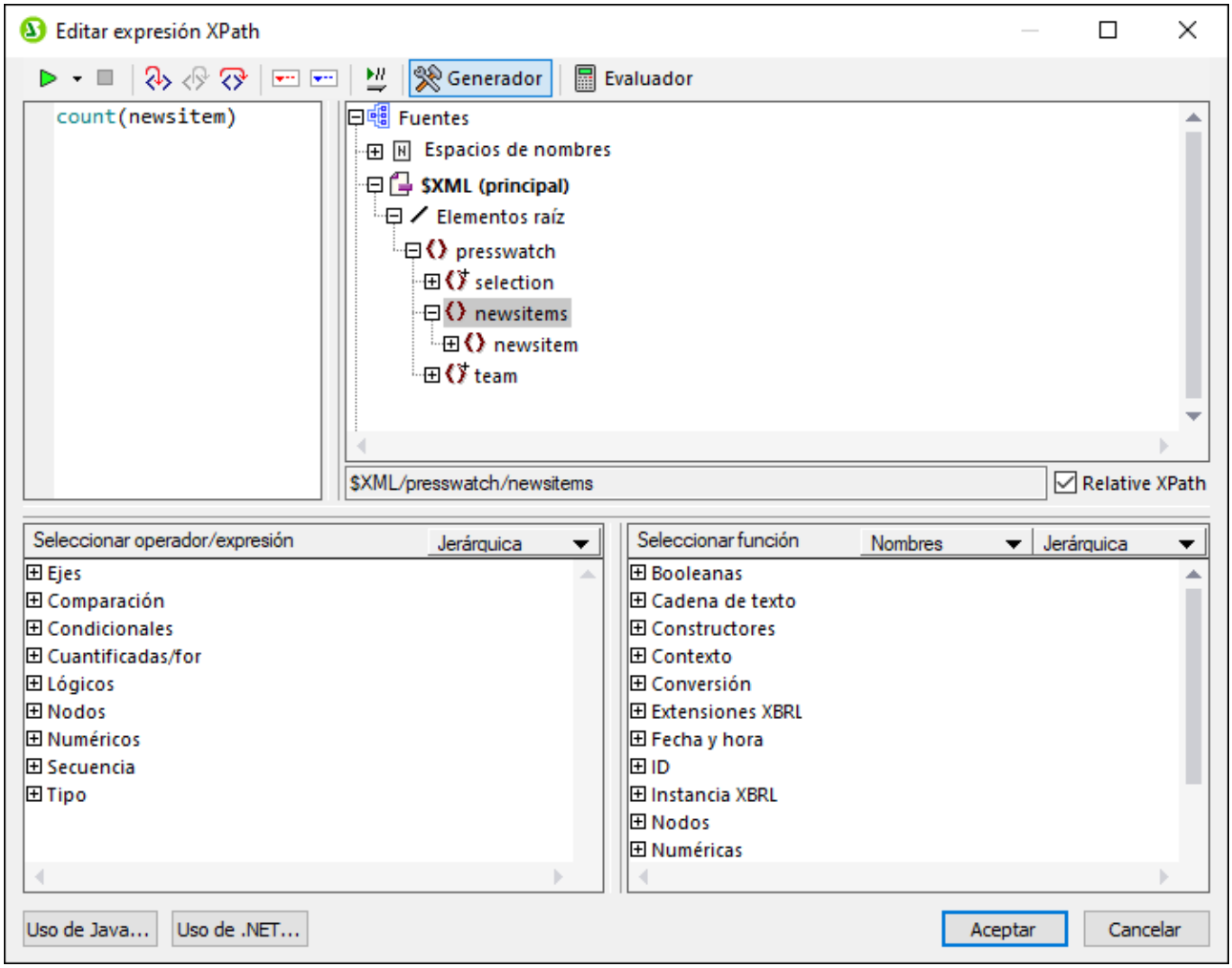
Comandos de la barra de herramientas en los paneles

Los paneles del cuadro de diálogo "Editar expresión XPath" del modo Depurador (*imagen anterior*) contiene botones 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 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.

16.2.3 Generador de expresiones XPath

Al activar el Generador de expresiones XPath (*imagen siguiente*) se habilitan varios paneles con ayudantes de entrada que puede usar para generar expresiones XPath. Haga doble clic en una entrada en cualquiera de estos ayudantes de entrada para introducirla en el punto de la expresión XPath en que se encuentre el cursor del ratón.



Existen tres paneles de ayudantes de entrada:

- Un esquema en forma de árbol que permite introducir nodos de elementos y atributos en la expresión XPath. Si se marca la casilla Relative XPath (XPath relativa), entonces la ruta de acceso del nodo seleccionado se entiende como relativa al nodo de contexto (el nodo del diseño en el que se genera la expresión XPath). El nodo de contexto se muestra más abajo en el panel del esquema. Si no se marca esta casilla, entonces se usa una expresión XPath absoluta para el nodo seleccionado; esta expresión XPath absoluta empieza en la raíz del documento.
- Un panel de ayudante de entrada para operadores y expresiones, que incluyen: (i) ejes (`ancestor::`, `parent::`, etc.), (ii) operadores (por ejemplo, `eq` y `div`) y (iii) expresiones (`for # in # return #`, etc.). Estos elementos del panel se pueden ordenar alfabéticamente en una lista o agrupar por

categoría funcional. Para seleccionar la opción que prefiera elija *Jerárquica* o *Nivelada* en el menú desplegable de la barra del título del panel.
- Un ayudante de entrada en el que las funciones de la versión XPath activa aparecen en orden alfabético en una lista o agrupadas por categoría funcional. Para elegir cómo quiere que aparezcan debe elegir *Jerárquica* o *Nivelada* en el menú desplegable de la barra del título del panel. La opción *Nombres/Tipos* permite elegir si los argumentos de las funciones se ven como nombres o como tipos de datos.

Características del Generador

- Para ver la descripción del texto de un elemento en cualquiera de los paneles, pase el cursor del ratón por encima del elemento.
- Cada una de las funciones aparece con su firma (es decir, con sus argumentos, los tipos de datos de los argumentos y los tipos de datos del resultado de la función).
- Las firmas aparecen con los nombres o los tipos de datos de los argumentos y el resultado de las funciones. Seleccione *Nombres* o *Tipos* en el menú desplegable de la barra del título del panel.
- Si hace doble clic en un elemento en cualquiera de los paneles (operador, expresión o función), entonces ese elemento se inserta donde se encuentre el cursor del ratón en la expresión. Las funciones se insertan con sus argumentos indicados por marcadores de posición (que representa el símbolo **#**).
- Si (i) selecciona texto en el campo de edición de la expresión XPath y (ii) hace doble clic en una expresión o en una función que contenga un marcador de posición para insertarla, entonces se insertará el texto seleccionado en lugar del marcador.

Una vez haya introducido una función en la expresión, si pasa el cursor del ratón por encima del nombre de esa función podrá ver la firma correspondiente, así como un texto con la descripción de la función. Si una función tiene más de una firma con el mismo nombre, esto de indica con un factor de sobrecarga en la parte inferior de la pantalla. Si coloca el cursor del ratón dentro de los paréntesis de la función y pulsa **Ctrl+Mayús+Barra espaciadora** puede ver las firmas de los distintos factores de sobrecarga del nombre de esa función.

Generar expresiones XPath

El cuadro de diálogo "Editar expresión XPath" cuenta con varias funciones que sirven de ayuda para generar expresiones XPath.

- *Nodo de contexto y esquema en árbol*
  La casilla *Selección* del panel *Fuentes* muestra de inmediato el nodo de contexto. La expresión se inserta dentro del nodo de contexto y se evalúa con él como su contexto.

- *Insertar un nodo del esquema en árbol*
  En el panel *Fuentes* se muestra todo el esquema. Haga doble clic en un nodo del esquema para insertarlo en la expresión XPath. Si marca la casilla XPath relativa, entonces el nodo seleccionado se inserte lo hará con una ruta de acceso a la expresión relativa al nodo de contexto.

- *Información del espacio de nombres*
  El esquema en árbol del panel Fuentes contiene un elemento Espacio de nombres. Si expande este elemento podrá ver todos los espacios de nombres declarados en esa hoja de estilos. Esta información puede ser útil para comprobar los prefijos de los espacio de nombres que esté pensando usar en una expresión XPath.

- *Insertar ejes, operadores y expresiones XPath*

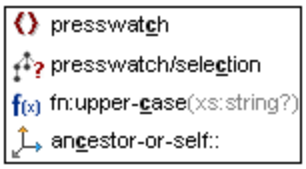
El panel Seleccionar operador/expresión contiene los ejes XPath (`ancestor::`, `parent::`, etc.), los operadores XPath (por ejemplo, `eq` y `div`) y las expresiones XPath (`for # in # return #`, etc.) para la versión XPath que se haya seleccionado como versión XSLT para el SPS. Puede ver estos elementos ordenados alfabéticamente o de forma jerárquica (que los agrupa según su función). Para insertar un eje o un operador en la expresión XPath, haga doble clic en el elemento correspondiente.

- *Insertar funciones XPath*
  En el panel Seleccionar función también puede ver las funciones XPath ordenadas alfabéticamente o agrupadas según su función (para cambiar de una vista a otra solo tiene que hacer clic en el icono correspondiente, en la parte superior del panel). Cada una de las funciones aparece con su firma. Si una función tiene más de una firma aparece tantas veces como firmas tenga. Los argumentos de una firma están separados por comas y cada argumento puede tener un indicador de frecuencia (`?` indica una secuencia de ninguno o de un elemento del tipo indicado; `*` indica una secuencia de cero o más elementos del tipo indicado). Los argumentos Los argumentos pueden mostrarse como nombres o como tipos de datos; seleccione *Nombres o tipos* en la barra del título del panel. Cada función indica también su tipo de retorno. Por ejemplo: `=> date ?` indica que el tipo de retorno esperado es una secuencia de ninguno o de un elemento `date`. Si coloca el cursor del ratón sobre una función aparece una breve descripción de la misma. Para insertar una función en la expresión XPath haga doble clic en la función que necesite.

- Las funciones de extensión Java y .NET se pueden usar en expresiones XPath, lo que permite acceder a las funciones de estos lenguajes de programación. Los botones **Java** y **.NET** que hay en la parte inferior del cuadro de diálogo sirven para activar textos informativos sobre cómo usar las funciones de extensión Java y .NET. Para más información al respecto consulte el apartado Funciones de extensión varias 1197 de esta documentación.

Edición inteligente mientras se teclea

Si escribe la expresión directamente en la caja de texto *Expresión*, entonces las opciones que estén disponibles en ese momento aparecerán en un mensaje emergente (*imagen siguiente*).

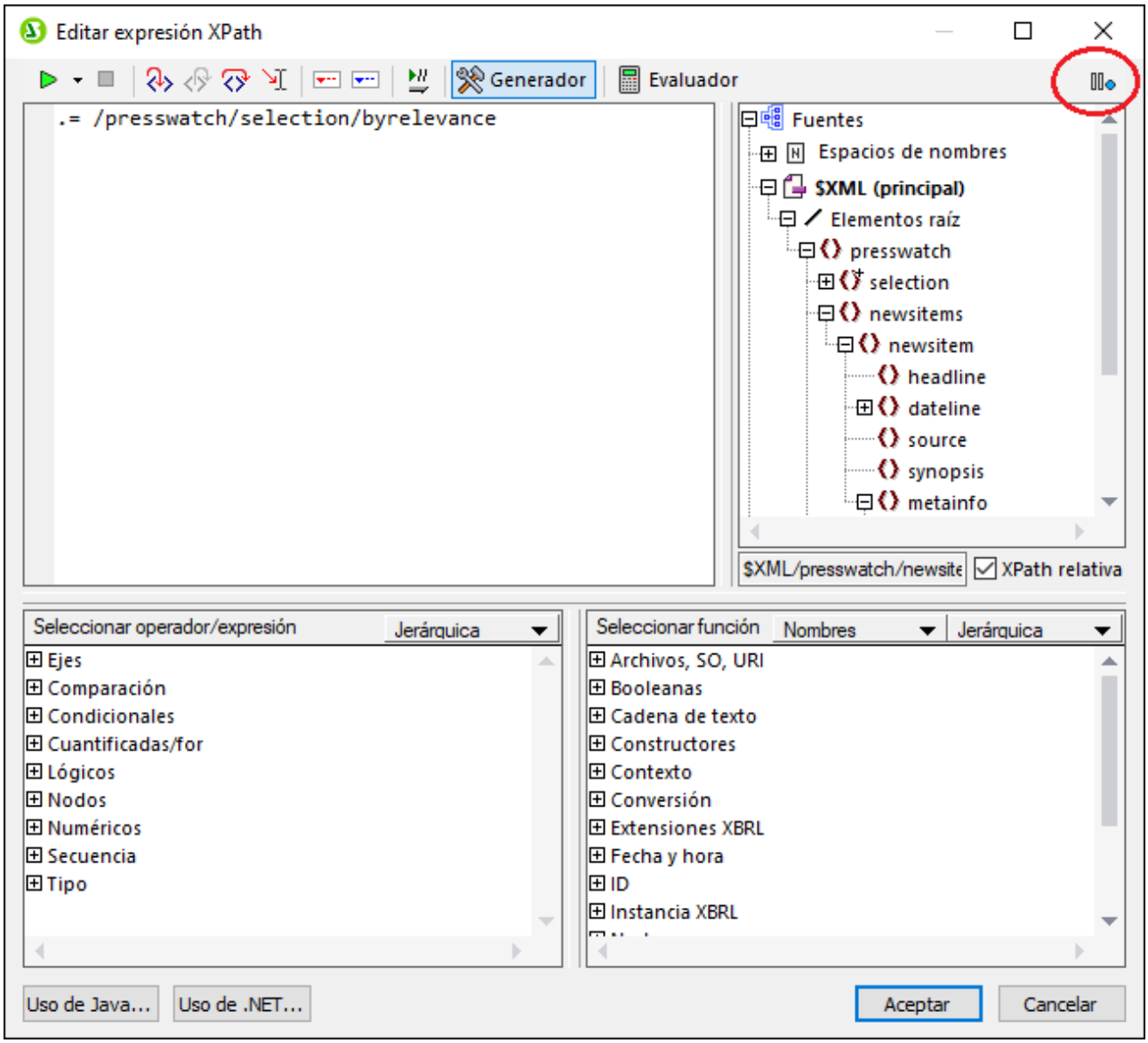


Esas opciones incluyen elementos, funciones XPath y ejes XPath. Puede usar las teclas **Arriba** y **Abajo** para moverse de una opción de la lista a otra. Cuando una opción esté resaltada, pulse la tecla **Entrar** si quiere introducirla en la expresión.

La casilla Otherwise

La casilla *Otherwise* (*marcada con una flecha roja en la imagen siguiente*) añade una rama `Otherwise` a una plantilla condicional como última rama. Cada plantilla condicional puede contener una rama `Otherwise` solamente. Al evaluar una plantilla condicional, la primera rama que se evalúa en `true` se ejecuta. Si ninguna de las ramas se evalúa en `true`, entonces se ejecuta la rama `Otherwise`, en caso de que exista; de lo contrario, la plantilla condicional se cierra sin ejecutar ninguna de sus ramas. La rama `Otherwise` se activa solamente si ninguna de las ramas anteriores se evalúa en `true`, por lo que no necesita definir una condición para ella. Por lo tanto, cuando se marca la casilla `Otherwise` se deshabilita el campo de entrada de la expresión XPath.

 © 2016-2022 Altova GmbH



Para más detalles sobre cómo usar la condición `Otherwise` consulte el apartado Condiciones 303.

16.3 Barras de herramientas

StyleVision ofrece acceso directo a una gran variedad de comandos mediante estas prácticas barras de herramientas:

- Formato 750
- Tabla 751
- Authentic 753
- RichEdit
- Insertar elementos de diseño 755
- Filtro para diseños 758
- Recursos globales 759
- Estándar 759

Los iconos de estas barras de herramientas y los comandos a los que corresponden se describen en los diferentes apartados de esta sección.

Posición de las barras de herramientas

Las barras de herramientas pueden flotar libremente por la pantalla o pueden colocarse en los bordes de la interfaz gráfica del usuario de StyleVision. Las barras de herramientas suelen colocarse en el borde superior, justo debajo de la barra de menús. Sin embargo, puede colocarlas en los bordes laterales e inferiores de la interfaz si lo prefiere.

Para colocar una barra de herramientas en los bordes de la interfaz gráfica:

1. Sujete la barra de herramientas por su controlador (si la barra ya está acoplada en los bordes de la interfaz) o por la barra de título (si la barra de herramientas está flotando).
2. Arrastre la barra de herramientas hasta la posición deseada y suéltela. Si todavía no existe una zona de barras de herramientas en el borde de la interfaz donde desea colocar la barra, se creará una automáticamente.

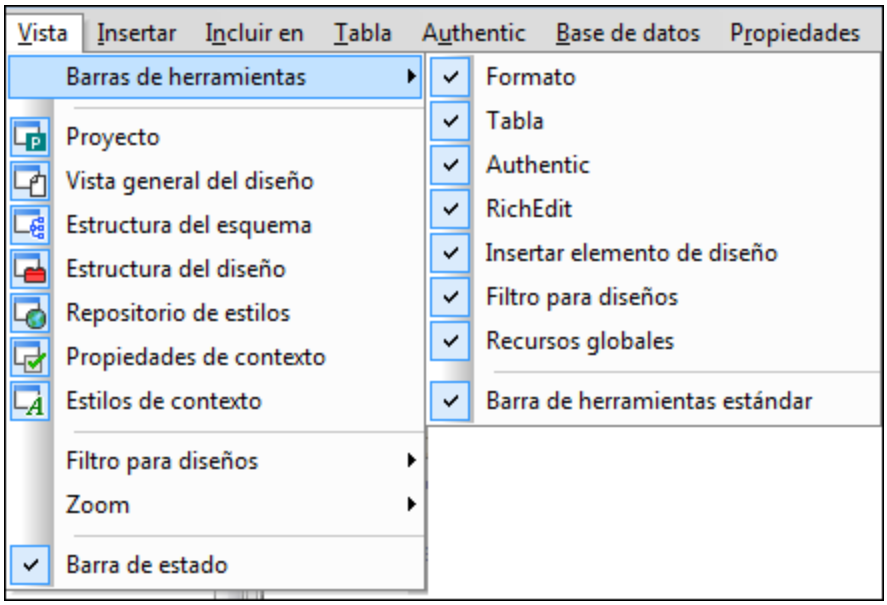
Para convertir una barra de herramientas en flotante, haga clic en su controlador, arrástrela fuera de la zona de barras de herramientas y suéltela en cualquier parte de la pantalla (excepto en un borde de la interfaz).

Mostrar u ocultar las barras de herramientas

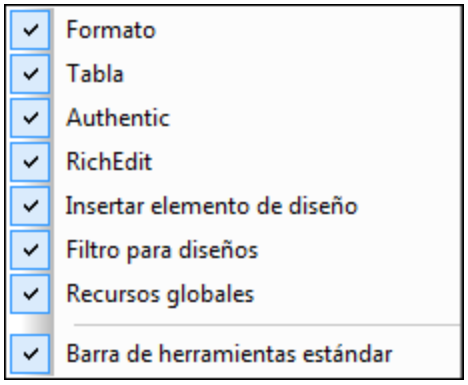
En StyleVision puede activar/desactivar (mostrar/ocultar) las barras de herramientas una por una. Esto se puede hacer de tres formas diferentes:

- En el menú **Vista | Barras de herramientas** (*imagen siguiente*), seleccione qué barras de herramientas se deben mostrar.

© 2016-2022 Altova GmbH



- Haga clic con el botón derecho en una zona de barras de herramientas y en el menú contextual que aparece (*imagen siguiente*) seleccione qué barras de herramientas se deben mostrar.

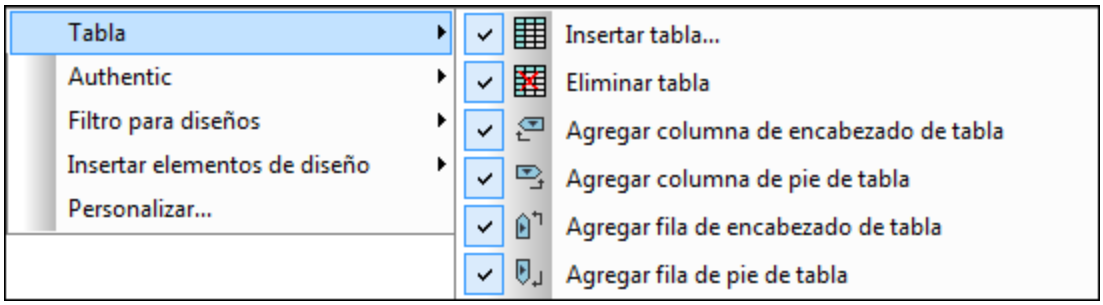


- En la pestaña *Barras de herramientas* del cuadro de diálogo Personalizar 870 (**Herramientas | Personalizar** 887), marque la casilla de las barras de herramientas que se deben mostrar. Cuando termine haga clic en **Cerrar**.

Agregar y quitar iconos de las barras de herramientas

En cada barra de herramientas puede añadir o eliminar iconos (es decir, puede mostrar/ocultar los iconos de las barras de herramientas). Para ello debe seguir estas instrucciones:

1. Elija la barra de herramientas que desea modificar y haga clic en la flecha que aparece al final de la barra.
2. Aparece el menú contextual **Agregar o quitar botones**, pase el cursor por este menú y aparece otro menú contextual (*imagen siguiente*) que enumera las barras de herramientas situadas en esa zona más la opción de menú **Personalizar...** (*imagen siguiente*).



3. Pase el cursor por la barra de herramientas que incluye el botón que desea añadir o quitar (*imagen siguiente*).
4. En el nuevo menú que aparece (*imagen anterior*), haga clic en el nombre del botón que desea añadir o quitar.
5. Si hace clic en la opción **Personalizar...** se abre el cuadro de diálogo Personalizar 887.

Al final del menú que incluye todos los botones de cada barra de herramientas hay una opción llamada **Restaurar la barra de herramientas**. Haga clic en esta opción para devolver la barra de herramientas a su estado de instalación.

Nota: los botones están predefinidos y no se pueden desvincular de su barra de herramientas.

16.3.1 Formato

La barra de herramientas **Formato** (*imagen siguiente*) se habilita cuando la vista activa es la vista **Diseño**. Sus comandos sirven para asignar propiedades de formato de bloque y línea al elemento seleccionado en la vista **Diseño**.

Formatos HTML predefinidos

El formato HTML seleccionado en la lista desplegable se aplica a la selección de la vista **Diseño**. Por ejemplo: el formato `div` aplica el elemento HTML `bloque (div)` a la selección actual de la vista **Diseño**. El formato HTML se convierte en las propiedades RTF correspondientes si el formato de salida elegido es RTF.

Propiedades de texto

Puede aplicar las propiedades de texto negrita, cursiva, subrayado y tachado a la selección actual de la vista **Diseño**. También puede seleccionar el estilo y tamaño de la fuente y el color de fondo y de primer plano.

Alineación

En esta barra de herramientas también puede definir la alineación del texto seleccionado en la vista **Diseño**: a la izquierda, en el centro, a la derecha o justificado.

 © 2016-2022 Altova GmbH

Listas

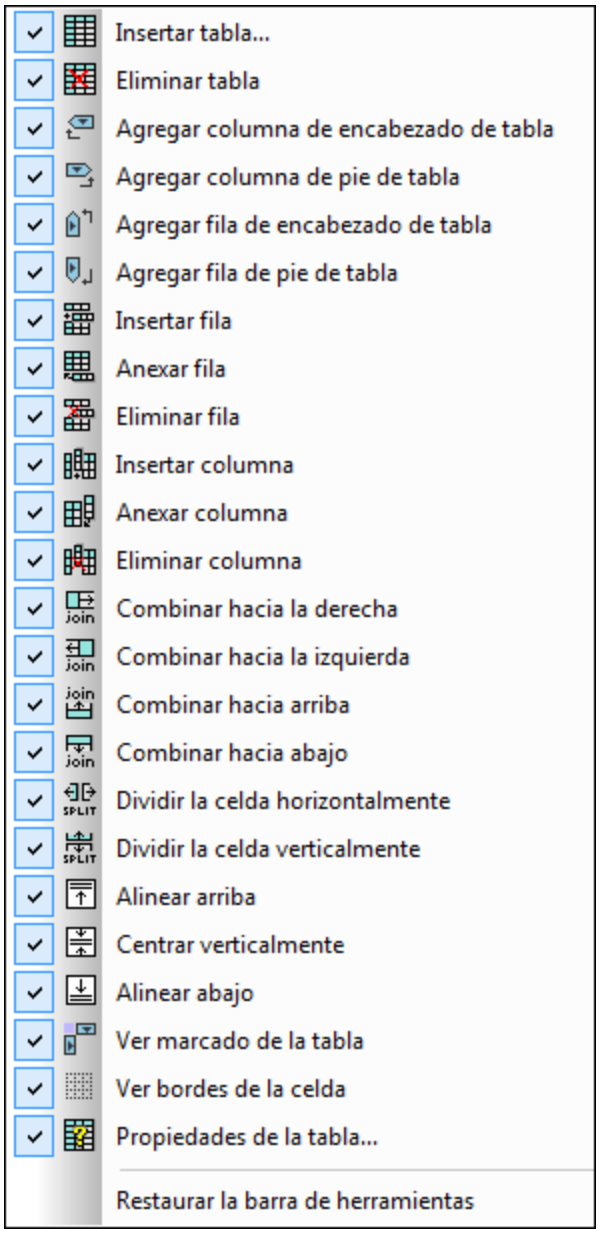
Puede insertar listas en el punto de inserción donde se encuentra el cursor o puede convertir el texto seleccionado en una lista.

Hipervínculos

Puede insertar un hipervínculo en el punto de inserción donde se encuentra el cursor. Para aprender a usar este comando consulte el apartado Hipervínculo 826.

16.3.2 Tabla

Los comandos de la barra de herramientas **Tabla** sirven para dar estructura y formato a las tablas dinámicas y estáticas seleccionadas en la vista **Diseño** (*imagen siguiente*).



Nota: la imagen anterior muestra el menú que aparece al hacer clic en el botón **Personalizar** situado a la derecha de la barra de herramientas Tabla.

Operaciones en filas y columnas

Puede insertar, anexar o eliminar filas y columnas en cualquier tabla estática o dinámica del diseño, en relación a la posición del cursor. Las filas y columnas se insertan antes de la posición del cursor o se anexan después de la última fila o columna. La fila/columna donde está el cursor también se puede eliminar. Estas operaciones se ejecutan con los comandos **Insertar fila/columna**, **Anexar fila/columna** o **Eliminar fila/columna**. También puede añadir encabezados y pies de tabla con los comandos **Agregar fila/columna de encabezado/pie de tabla**.

 © 2016-2022 Altova GmbH

Operaciones en celdas

La celda de la tabla donde está el cursor se puede unir a cualquiera de las celdas adyacentes. Esta operación es similar a expandir celdas de tabla en HTML. Los comandos que debe utilizar para estas operaciones son **Combinar hacia la derecha/izquierda/arriba/abajo**. Además la celda de la tabla donde está el cursor se puede dividir horizontal o verticalmente con los comandos **Dividir la celda horizontalmente** y **Dividir la celda verticalmente**. Asimismo puede alinear el contenido de la celda en la parte superior, central o inferior de la celda. Además puede ocultar o mostrar los bordes de la celda con el comando **Ver bordes de la celda**.

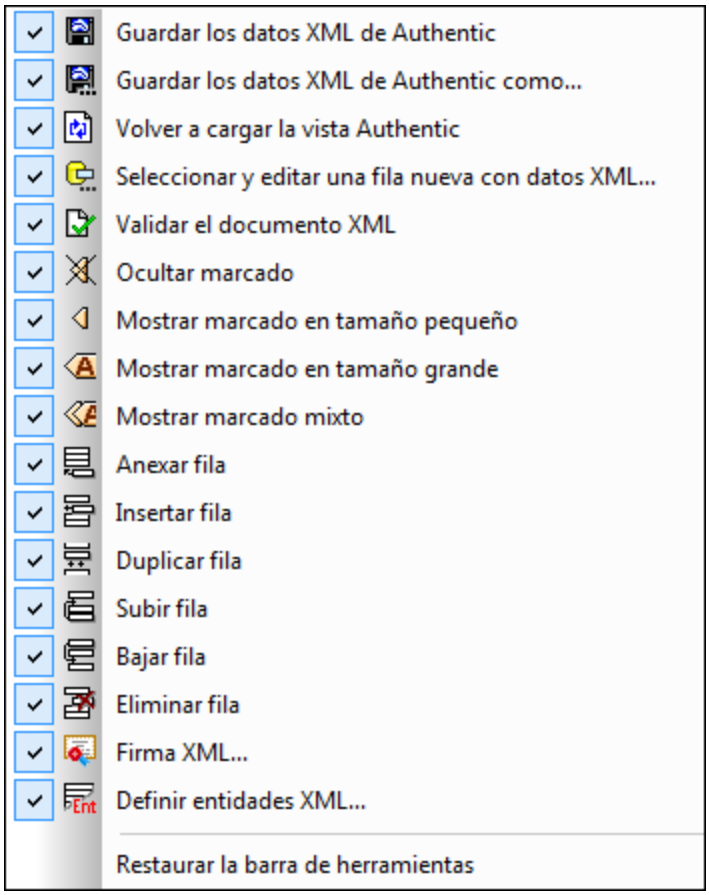
Operaciones en la tabla, propiedades y presentación

Si pone el cursor en una tabla estática o dinámica y hace clic en **Eliminar tabla** 844, la tabla se elimina. El marcado de la tabla se puede habilitar o deshabilitar con el comando **Ver marcado de la tabla**. Por último, con el comando **Propiedades de la tabla** puede abrir el cuadro de diálogo donde se definen las propiedades de la tabla.

16.3.3 Authentic

Los comandos de la barra de herramientas **Authentic** sirven para personalizar la vista Authentic y editar documentos XML en la vista Authentic (*imagen siguiente*).

Estos comandos permiten al diseñador del SPS probar todas las características que estarán a disposición del usuario de la vista Authentic.



Nota: la imagen anterior muestra el menú que aparece al hacer clic en el botón **Personalizar** situado a la derecha de la barra de herramientas Authentic.

Validar, guardar y volver a cargar documentos XML

Mientras edita un documento XML en la vista Authentic, puede comprobar la validez del archivo XML de trabajo haciendo clic en el comando **Validar el documento XML**. Los cambios realizados se pueden guardar en el archivo XML de trabajo con el comando **Guardar los datos XML de Authentic**. La última versión del documento XML se puede volver a cargar en cualquier momento.

Seleccionar y editar una fila nueva con datos XML

Este comando solamente se habilita en diseños SPS basados en bases de datos XML y permite cargar una fila nueva de la columna XML en la vista Authentic para poder editarla. Consulte la descripción del comando 856 para obtener más información.

Etiquetas de marcado en la vista Authentic

En la vista Authentic puede elegir cómo aparecen las etiquetas de marcado: puede ocultarlas (**Ocultar marcado**), mostrarlas con los nombres de nodo (**Mostrar marcado en tamaño grande**), sin nombres de nodo (**Mostrar marcado en tamaño pequeño**) o cada nodo de una forma (**Mostrar marcado mixto**).

 © 2016-2022 Altova GmbH

Edición de tablas dinámicas en la vista Authentic

En la vista Authentic pueden ejecutarse operaciones en filas de tablas dinámicas. Es decir, se pueden insertar, anexar, duplicar, subir y bajar filas haciendo clic en los iconos pertinentes de la barra de herramientas: **Insertar fila**, **Anexar fila**, **Duplicar fila**, **Subir fila**, **Bajar fila** y **Eliminar fila**.

Definir entidades

Mientras edita el archivo XML de trabajo en la vista Authentic puede definir entidades. Haga clic en el botón **Definir entidades XML** para abrir el cuadro de diálogo "Definir entidades XML". Para más información consulte el apartado Authentic | Definir entidades 857.

16.3.4 RichEdit

Los comandos de la barra de herramientas **RichEdit** sirven para marcar el texto en la vista Authentic con propiedades de estilo.

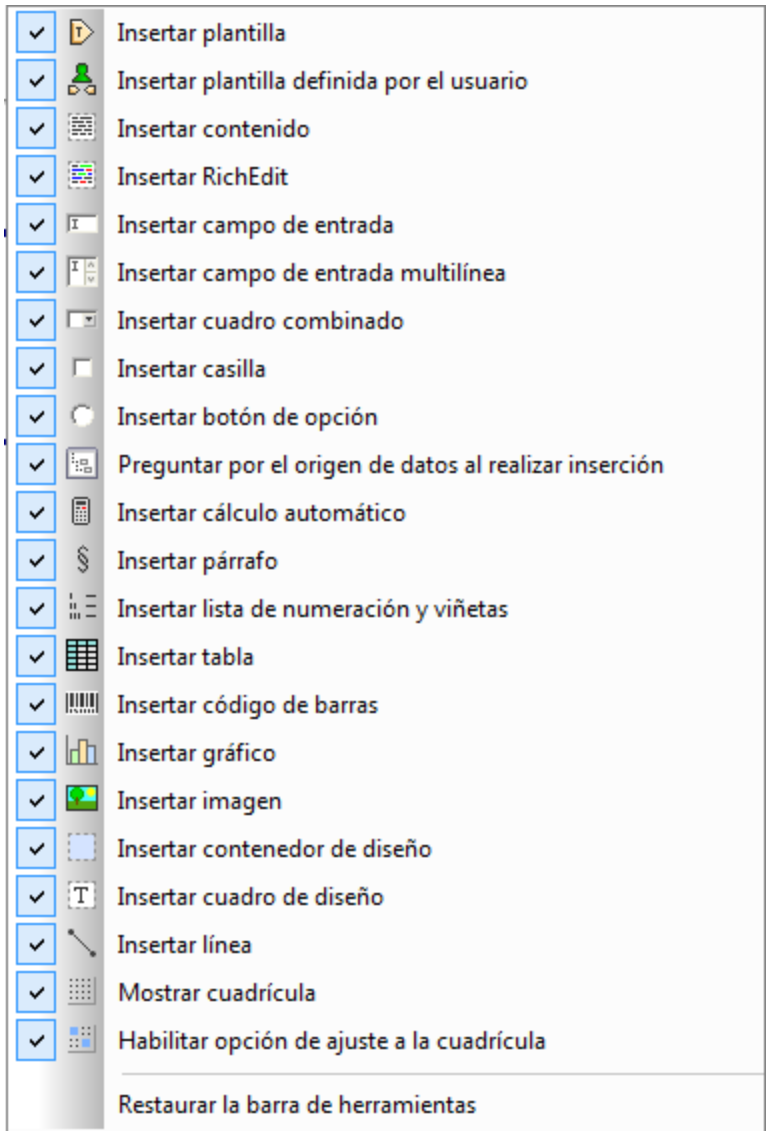
Cuando el usuario de la vista Authentic pone el cursor dentro de un elemento que se creó como componente RichEdit 410, los botones y controles de la barra de herramientas **RichEdit** se habilitan (*imagen siguiente*). De lo contrario, estos botones están siempre deshabilitados.

Gracias a la función RichEdit el usuario de la vista Authentic puede especificar la fuente del texto, su tamaño, su estilo, su decoración, su color, su color de fondo y su alineación. Seleccione el texto al que desea aplicar estilos en la vista Authentic y seleccione el estilo con los botones y controles de la barra de herramientas. El texto al que se aplica el estilo aparece entre las etiquetas del elemento de estilo 410.

16.3.5 Insertar elementos de diseño

Los comandos de la barra de herramientas **Insertar elementos de diseño** sirven para insertar elementos de diseño en el SPS. En la imagen siguiente puede ver todos los elementos de diseño que se pueden insertar con ayuda de esta barra de herramientas. La barra de herramientas incluye comandos de tres tipos:

1. Comandos para insertar componentes de diseño 756, que dependen del nodo de contexto (la mayoría de los comandos de la barra de herramientas).
2. Comandos para insertar componentes de distribución 757, que no dependen del nodo de contexto.
3. Comandos relacionados con la cuadrícula del diseño 758, que ayudan a trabajar en la planificación gráfica del SPS.

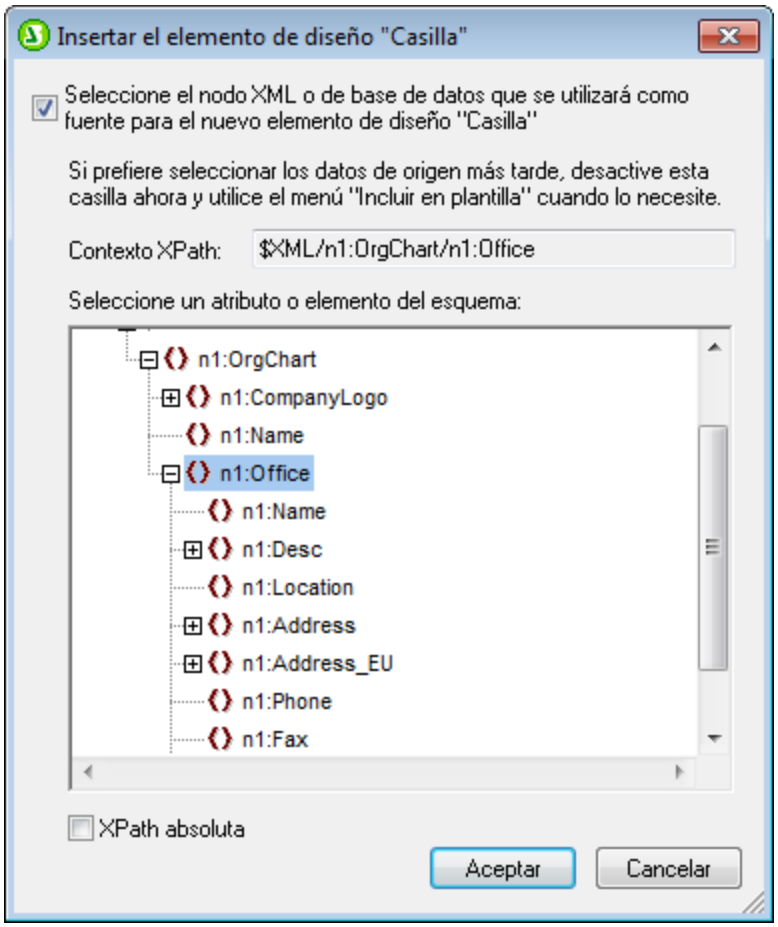


Comandos para insertar componentes de diseño

Los componentes de diseño son elementos que dependen del nodo de contexto y se insertan desde el menú **Insertar**. Siga estos pasos para insertar un elemento de diseño con la barra de herramientas **Insertar**:

1. En la barra de herramientas haga clic en el icono del elemento que desea insertar.
2. Ahora haga clic en la posición de la vista **Diseño** donde desea insertar el elemento. Acto seguido aparece un cuadro de diálogo (*imagen siguiente*) que muestra la estructura del esquema y el nodo de contexto. El nodo de contexto es el nodo dentro del que se insertará el elemento de diseño.

 © 2016-2022 Altova GmbH



3. Si quiere insertar el elemento de diseño dentro del nodo de contexto actual, haga clic en **Aceptar**. Si quiere usar un contexto diferente, seleccione otro nodo de contexto en la estructura del esquema y haga clic en **Aceptar**.
4. En el caso de algunos elementos de diseño (como los cálculos automáticos, por ejemplo) se necesita un paso más. En otros casos, como cuando se trabaja con plantillas definidas por el usuario, este cuadro de diálogo ni siquiera aparece. En estos casos aparece el diálogo Editar expresión XPath 729. Elija las opciones necesarias y haga clic en **Aceptar**.

El elemento de diseño se inserta al final del paso 3 o 4, dependiendo del tipo de elemento de diseño.

Comandos para insertar componentes de distribución

La barra de herramientas **Insertar elementos de diseño** incluye comandos para insertar tres elementos de distribución diferentes: (i) contenedores de diseño, (ii) cuadros de diseño y (iii) líneas. No olvide que las líneas y los cuadros de diseño solamente se pueden insertar dentro de un contenedor de diseño.

Para insertar un contenedor de diseño, haga clic en el icono **Insertar contenedor de diseño** y después haga clic en la posición de la vista **Diseño** donde desea insertar el contenedor. Aparece un aviso donde debe indicar el tamaño del contenedor. Para insertar un cuadro de diseño, haga clic en el icono **Insertar cuadro de diseño** y después haga clic en la posición del contenedor de diseño donde desea insertar el cuadro. Ahora haga clic dentro del cuadro y empiece a escribir. Para insertar una línea, haga clic en el icono **Insertar línea** y después haga clic en la posición del contenedor de diseño donde quiere empezar a dibujar la línea. Arrastre el cursor hasta el punto donde desea que finalice la línea y suéltelo.

Para cambiar el tamaño de los contenedores y cuadros de diseño debe pasar el cursor por el contorno de estos elementos y arrastrarlo hasta obtener el tamaño deseado. Para mover un cuadro de diseño, pase el cursor por el borde superior o izquierdo del cuadro y cuando el cursor tome forma de cruz, arrastre el cuadro hasta su nueva posición.

Comandos relacionados con la cuadrícula

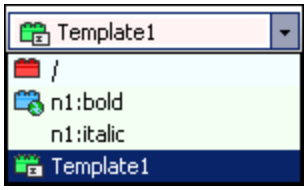
El comando **Mostrar cuadrícula** muestra/oculta la cuadrícula de diseño. Si activa el comando **Ajustar a la cuadrícula**, los elementos creados dentro del contenedor de diseño (como cuadros de diseño y líneas) se ajustan a las líneas y a las intersecciones de las líneas. Las propiedades de la cuadrícula pueden definirse en la pestaña *Diseño* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).

16.3.6 Filtro para diseños

Los comandos de la barra de herramientas **Filtro para diseños** (*imagen siguiente*) sirven para filtrar el tipo de plantillas que aparecen en la vista **Diseño** y el tipo de plantillas que se ocultan.

Icono	Comando	Descripción
	Mostrar solo una plantilla	Solo muestra la plantilla seleccionada. Ponga el cursor en una plantilla y haga clic en este icono para ver solo la plantilla seleccionada.
	Mostrar todos los tipos de plantilla	Muestra todas las plantillas del SPS (plantilla principal, global, con nombre y de diseño).
	Mostrar plantillas importadas	Haga clic para mostrar/ocultar las plantillas importadas.
	Mostrar/ocultar la plantilla principal	Haga clic para mostrar/ocultar la plantilla principal.
	Mostrar/ocultar las plantillas globales	Haga clic para mostrar/ocultar las plantillas globales.
	Mostrar/ocultar los fragmentos de diseño	Haga clic para mostrar/ocultar los fragmentos de diseño.

El cuadro combinado *Filtro para diseños* (*imagen siguiente*) incluye una lista con todas las plantillas existentes en el diseño SPS.

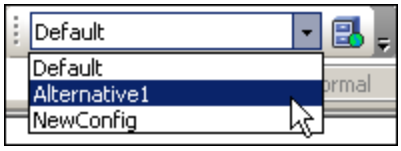


 © 2016-2022 Altova GmbH

Si selecciona una plantilla en el cuadro combinado, la plantilla se selecciona en el diseño. Es decir, con este cuadro combinado puede navegar rápidamente hasta la plantilla deseada en el diseño. Esto es muy práctico si su diseño tiene varias plantillas y si algunas están ocultas.

16.3.7 Recursos globales

La barra de herramientas **Recursos globales** (*imagen siguiente*) sirve para (i) seleccionar la configuración activa para la aplicación y (ii) abrir el cuadro de diálogo Recursos globales de Altova 444.



Seleccione la configuración activa en la lista desplegable del cuadro combinado. Haga clic en el icono **Administrar recursos globales** para abrir el cuadro de diálogo "Recursos globales de Altova".

16.3.8 Estándar

Los comandos de la barra de herramientas Estándar ejecutan importantes funciones de edición y otras operaciones en el archivo. Todos estos comandos se enumeran y describen a continuación. Para obtener una descripción más completa de un comando, consulte el apartado correspondiente de la Referencia del usuario.

Icono	Comando	Tecla de acceso rápido	Descripción
	Diseño nuevo a partir de un esquema XML, DTD o archivo XML 761	**Ctrl+N**	Crea un documento SPS nuevo basado en un esquema. Haga clic en la punta de flecha para crear un SPS basado en una BD o en un HTML o para crear un SPS en blanco.
	Abrir 768	**Ctrl+O**	Abre un documento SPS ya existente.
	Volver a cargar		Vuelve a cargar la última versión guardada del SPS.
	Guardar diseño 773	**Ctrl+S**	Guarda el documento SPS activo.
	Guardar todos 773	**Ctrl+Mayús+S**	Guarda todos los documentos SPS que están abiertos.
	Imprimir 791	**Ctrl+P**	Imprime la vista Authentic del archivo XML de trabajo.

Icono	Comando	Tecla de acceso rápido	Descripción
	Vista previa de impresión 791		Muestra una vista previa de impresión de la vista Authentic del archivo XML de trabajo.
	Cortar 794	**Mayús+Supr**	Corta la selección y la pone en el portapapeles.
	Copiar 794	**Ctrl+C**	Copia la selección en el portapapeles.
	Pegar 794	**Ctrl+P**	Pega el contenido del portapapeles en la posición del cursor.
	Eliminar 794	**Supr**	Elimina la selección
	Deshacer 794	**Alt+ Retroceso**	Deshace el cambio realizado. Puede utilizar la función Deshacer un número ilimitado de veces.
	Rehacer 794	**Ctrl+Y**	Rehace la acción deshacer.
	Buscar 794	**Ctrl+F**	Busca texto en la vista Authentic y en las vistas de resultados.
	Buscar siguiente 794	**F3**	Busca la siguiente aparición del término de búsqueda.
	Zoom 810		Establece el nivel de zoom de la vista Diseño.
	Mostrar marcado del diseño en tamaño pequeño		Convierte las etiquetas de marcado en etiquetas pequeñas.
	Mostrar marcado del diseño en tamaño grande		Convierte las etiquetas de marcado en etiquetas grandes.
	XSLT 1.0 115		Establece XSLT 1.0 como lenguaje de la hoja de estilos.
	XSLT 2.0 115		Establece XSLT 2.0 como lenguaje de la hoja de estilos.
	XSLT 3.0 115		Establece XSLT 3.0 como lenguaje de la hoja de estilos.
	Ortografía 866		Revisa la ortografía del documento SPS.
	Mostrar/ocultar "Consulta de la BD" 860		Muestra/oculta el cuadro de diálogo "Consulta de la BD".

© 2016-2022 Altova GmbH

16.4 Menú Archivo

Los comandos del menú **Archivo** sirven para trabajar con documentos SPS y archivos relacionados. Estos son los comandos del menú **Archivo**:

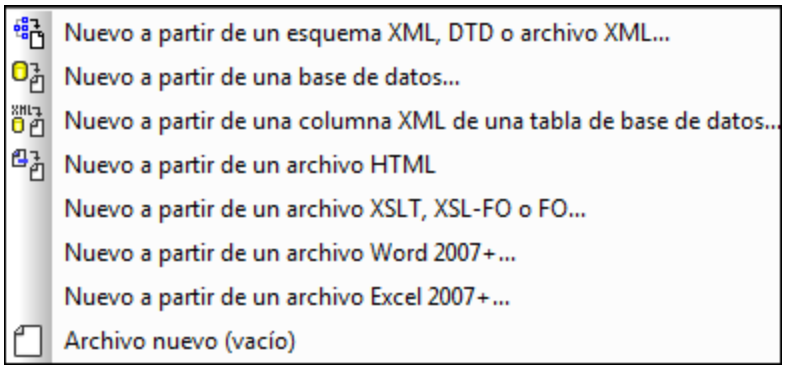
- **Nuevo** 761: para crear un diseño SPS nuevo a partir de diferentes documentos fuente.
- **Abrir, Volver a cargar, Cerrar, Cerrar todos** 768 ; 768 para abrir, cerrar y volver a cargar el archivo activo.
- **Guardar el diseño, Guardar como, Guardar todos** 773: para guardar el archivo SPS activo y todos los archivos SPS que están abiertos.
- **Exportar como archivo de diseño de MobileTogether** 782: para generar un diseño de MobileTogether a partir del archivo SPS activo.
- **Guardar los datos XML de Authentic, Guardar los datos XML de Authentic como** 782: se habilitan si la vista activa es la vista Authentic. Sirven para guardar los cambios en el archivo XML de trabajo 30.
- **Guardar archivos generados** 782: para guardar los archivos de salida que se pueden generar a partir del SPS.
- **Diseño web** 787: genera todos los archivos necesarios para ejecutar una aplicación ASPX en la carpeta de destino indicada.
- **Propiedades** 787: para definir la codificación de los documentos de salida, el modo de compatibilidad CSS del explorador web, cómo se deben resolver las rutas de acceso relativas de las imágenes en la vista Authentic y si las imágenes deben incrustarse o enlazarse en los documentos RTF (*ediciones Enterprise y Professional*) y Word 2007+ (*edición Enterprise*).
- **Vista previa de impresión, Imprimir** 791: se habilitan en la vista Authentic y en las vistas previas. Estos comandos imprimen la vista.
- **Archivos usados recientemente, Salir** 792: para abrir un archivo utilizado recientemente y para salir del programa.

Temas relacionados

- Barras de herramientas 748
- Barra de herramientas Estándar 759
- Barra de herramientas Authentic 753

16.4.1 Nuevo

Al pasar el cursor sobre el comando **Archivo | Nuevo** aparece un submenú (*imagen siguiente*) con opciones para crear un documento SPS nuevo de varios tipos diferentes:



- Un archivo SPS nuevo basado en un esquema XML, un documento DTD o un esquema XML generado a partir de un archivo XML (**Nuevo a partir de un esquema XML, DTD o archivo XML...**). El esquema seleccionado se añade a la ventana Vista general del diseño 45 y se crea una representación gráfica de la estructura del esquema en la ventana Estructura del diseño 48. En la vista **Diseño** 37 se abre el documento SPS nuevo, que aparece en blanco con una plantilla principal vacía. También puede crear un archivo SPS nuevo a partir de un archivo (esquema o XML) almacenado en una URL o en un recurso global (*ver más abajo*).
- Un archivo SPS nuevo basado en un esquema XML generado a partir de una BD (**Nuevo a partir de una base de datos** o **Nuevo a partir de una columna XML de una tabla de base de datos**). El proceso de conexión se describe en la sección Conectarse a una BD y configurar el diseño SPS 523. En la vista **Diseño** 37 se abre el documento SPS nuevo, que aparece en blanco con una plantilla principal vacía.
- Un archivo SPS nuevo basado en un esquema definido por el usuario que se crea nodo a nodo a partir de un archivo HTML 502 (**Nuevo a partir de un archivo HTML**). El esquema definido por el usuario se añade a la ventana Vista general del diseño 45 y Estructura del esquema 48. En la ventana Estructura del esquema, el esquema tiene un solo elemento de documento (elemento raíz) y el archivo HTML se carga en la vista **Diseño** 37.
- Un archivo SPS nuevo basado en un archivo XSLT, XSL-FO o FO. La estructura y el estilo de la plantilla del archivo XSLT se crea en el SPS. Después puede modificar los componentes de diseño y añadir contenido y formato al SPS. Para más información consulte el apartado Diseño nuevo a partir de archivos XSLT, XSL-FO y FO 471.
- Un archivo SPS nuevo que incluye el contenido de un documento MS Word como texto estático del diseño 143.
- Un archivo SPS nuevo que incluye el contenido de un documento MS Excel como texto estático del diseño 146.
- Un archivo SPS nuevo vacío (**Archivo nuevo (vacío)**). Este comando no añade ningún esquema a las ventanas Vista general del diseño y Estructura del esquema. En la vista **Diseño** 37 se crea una plantilla principal vacía.

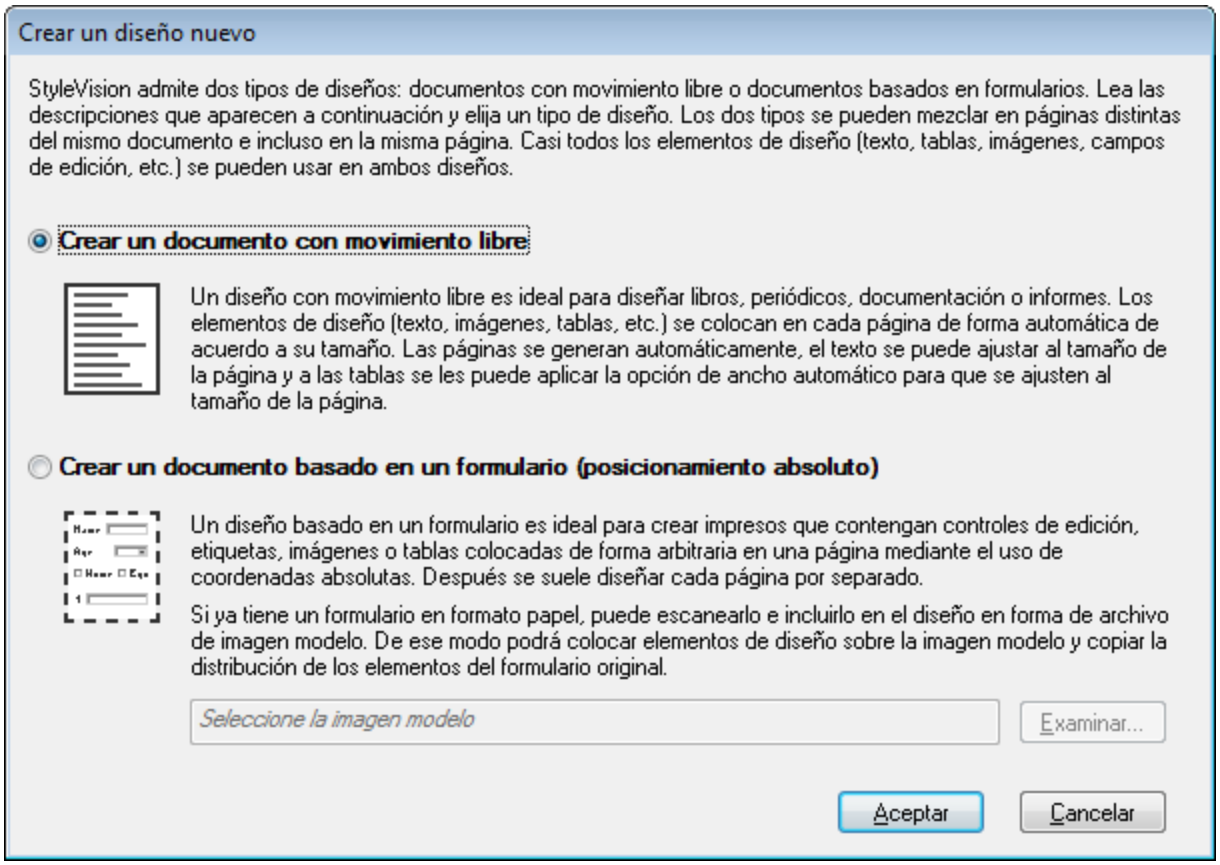
Nota: si quiere puede usar un recurso global 444 para encontrar un archivo o BD.

Seleccionar el tipo de diseño

Una vez seleccionados los archivos fuente (XSD y XML), aparece el cuadro de diálogo "Crear un diseño nuevo".

El cuadro de diálogo "Crear un diseño nuevo" (*imagen siguiente*) ofrece dos opciones. Puede crear: (i) un diseño de *documento con movimiento libre* o (ii) un diseño de *documento basado en un formulario*, en el que los componentes tienen posiciones absolutas (como en un programa de maquetación).

 © 2016-2022 Altova GmbH

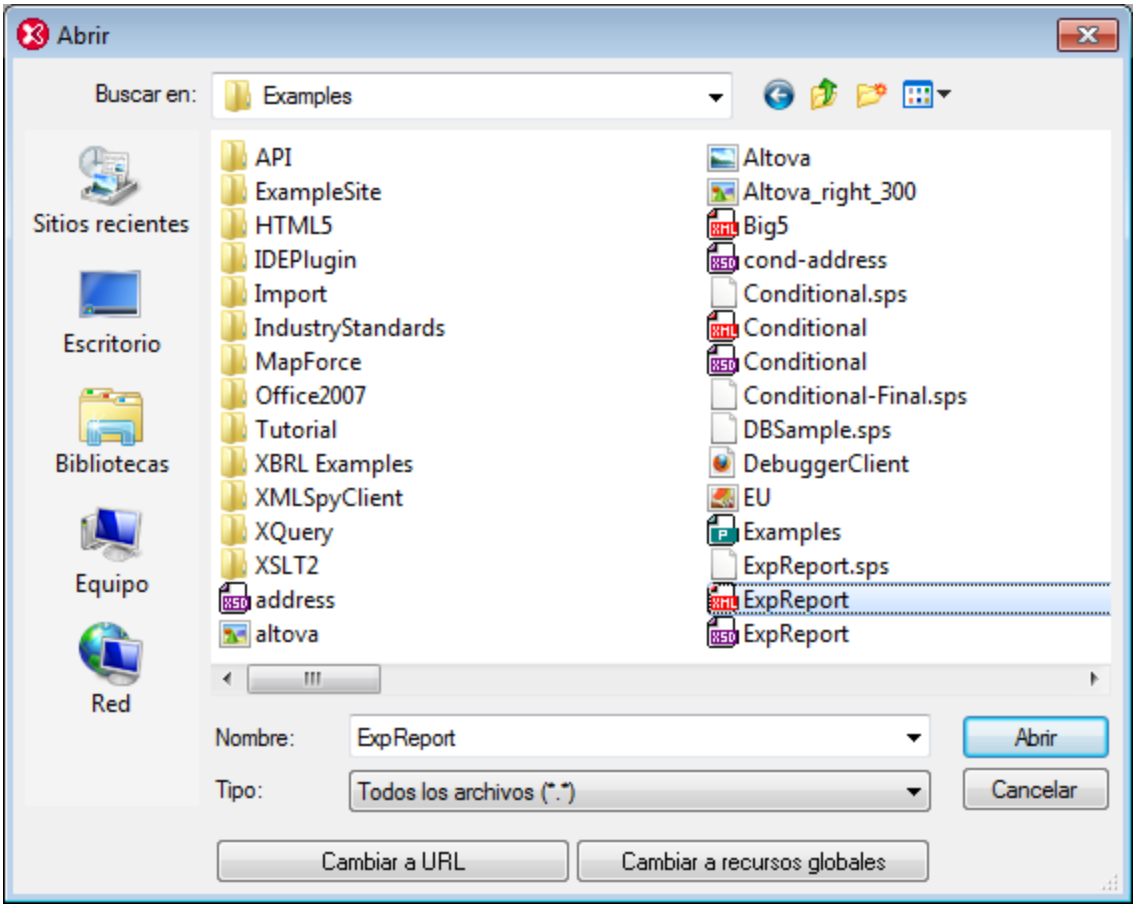


En los *documentos con movimiento libre*, el contenido se ajusta al formato de salida. Los elementos de contenido solamente se pueden colocar en relación a los demás y no con posición absoluta. Este tipo de diseño es ideal para la creación de informes, artículos y libros.

En los *documentos basados en formularios* se crea un solo contenedor de diseño 204, donde los componentes se pueden colocar con posición absoluta. Las dimensiones del contenedor de diseño las define el usuario, que puede colocar cuadros de diseño con posición absoluta dentro del contenedor de diseño. Después puede colocar el contenido del documento dentro de los cuadros de diseño. Si desea que el SPS sea igual que un formulario concreto, puede usar una imagen del formulario original como imagen modelo 204. La imagen modelo se puede insertar como imagen de fondo en el contenedor de diseño. La imagen modelo le ayudará a diseñar su formulario pero no se incluirá en los documentos de salida.

- Seleccionar y guardar archivos a través de direcciones URL y recursos globales

  En varios cuadros de diálogo "Abrir archivo" y "Guardar archivo" de StyleVision 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.

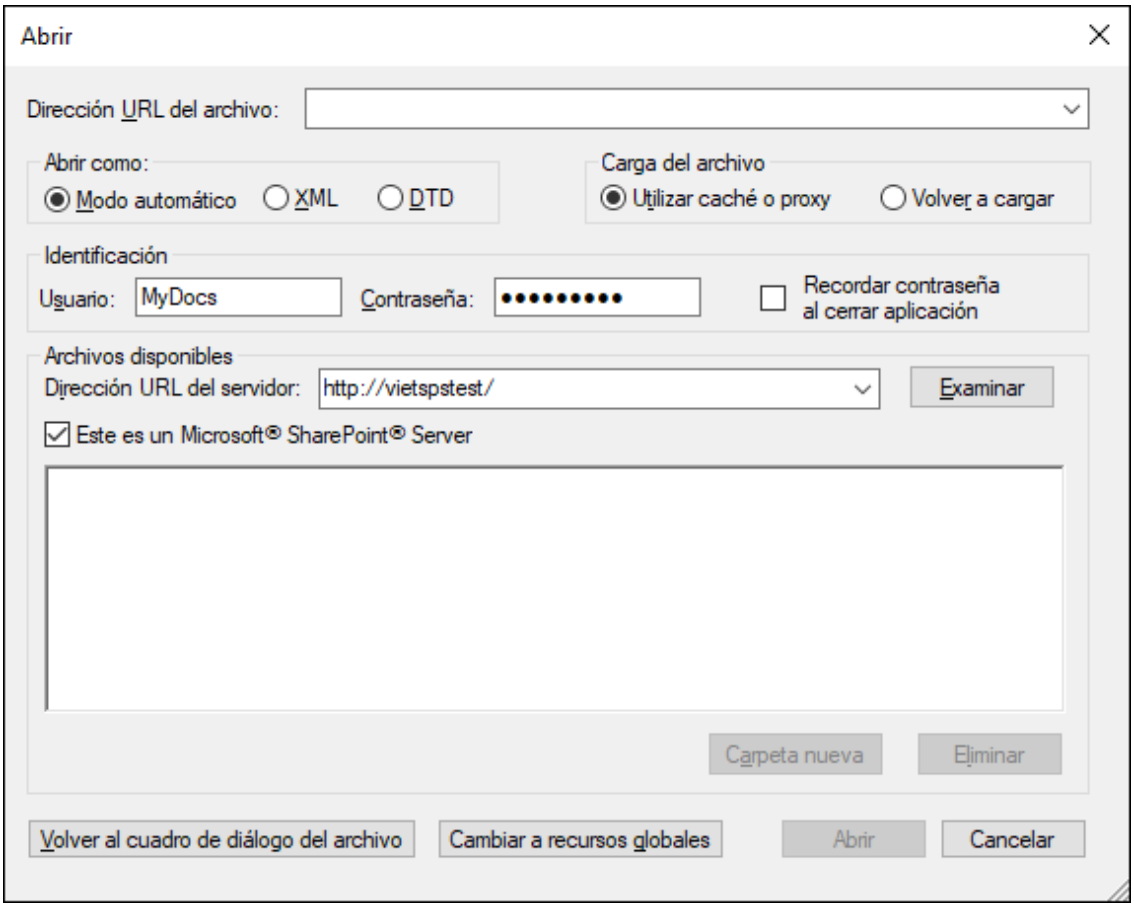


Seleccionar archivos a través de direcciones URL

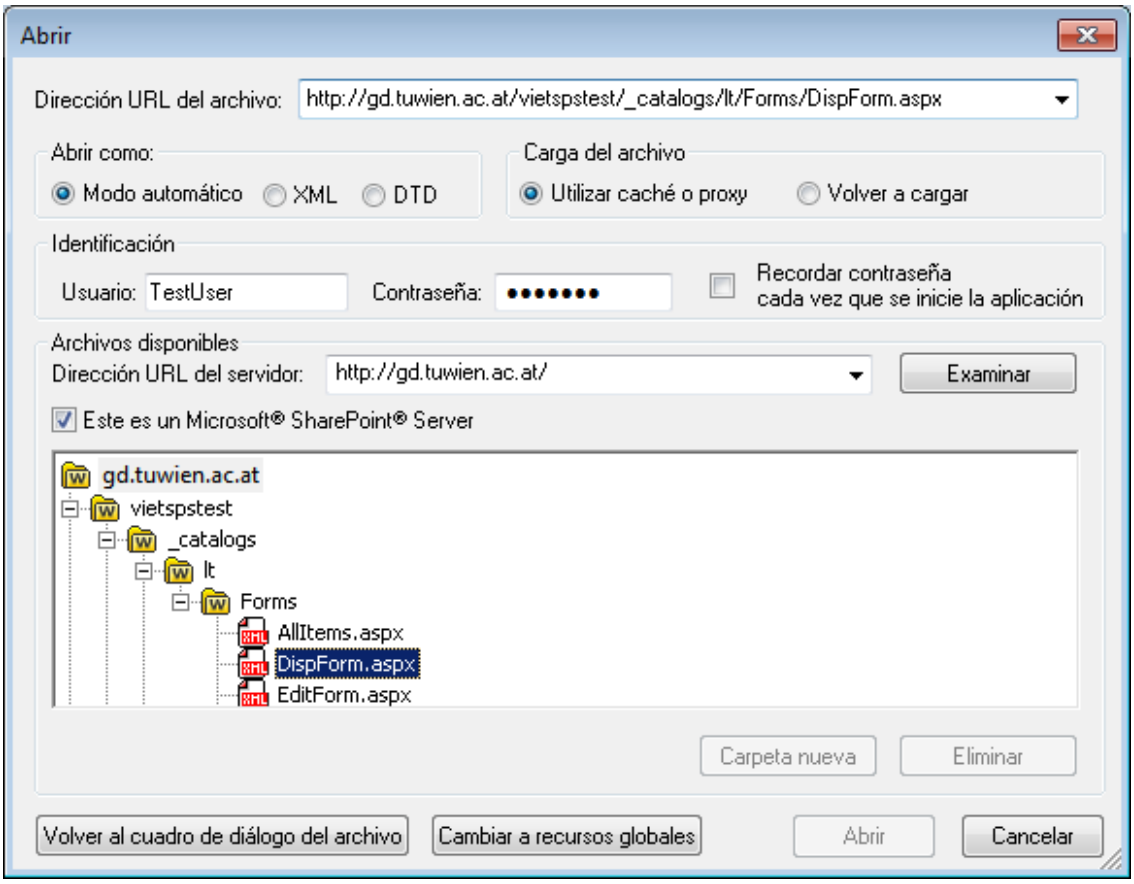
Siga estas instrucciones para seleccionar un archivo a través de una dirección URL (ya sea para abrirlo o para guardarlo):

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:

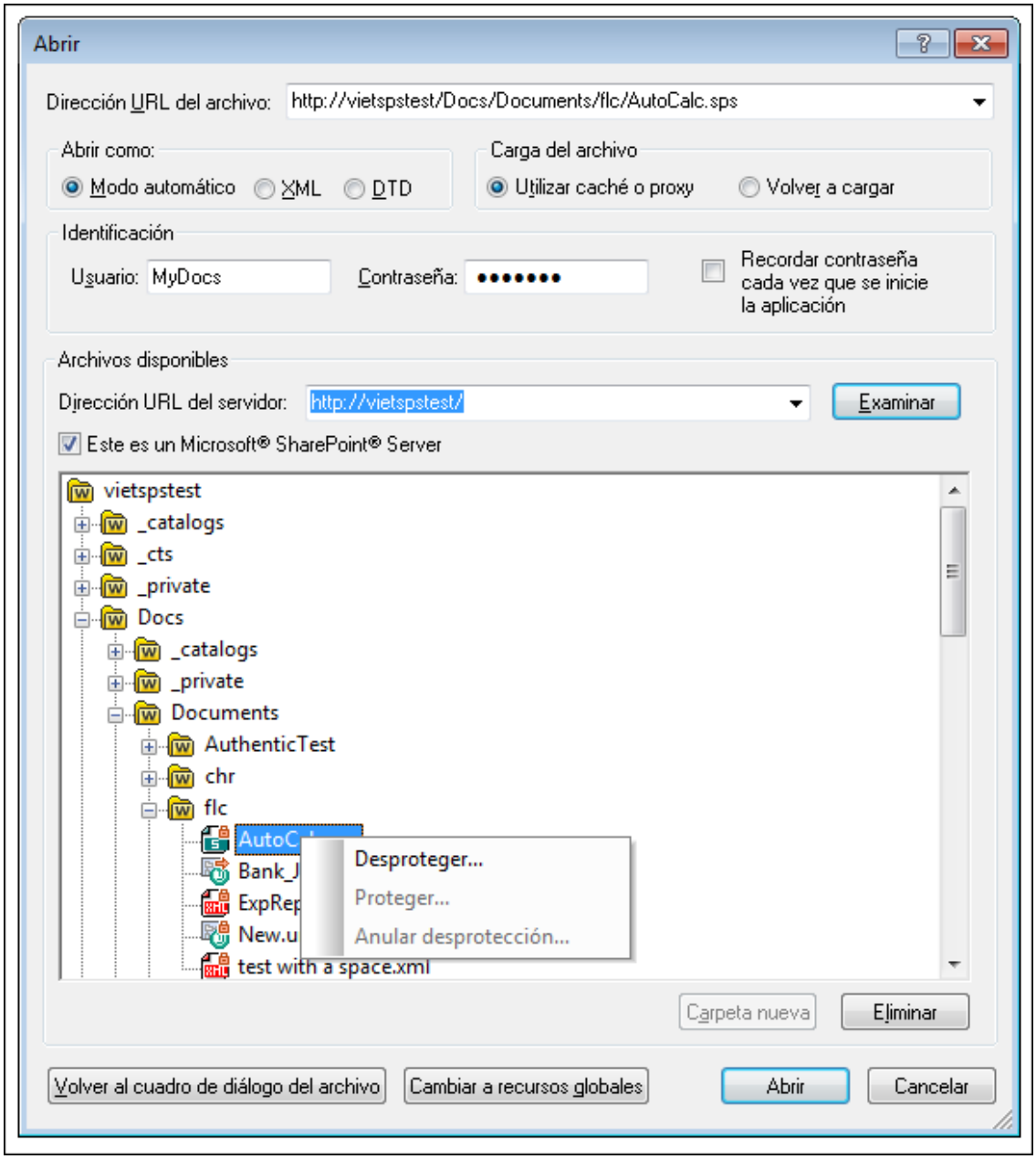
- La función **Examinar** solamente está disponible en servidores compatibles con WebDAV y servidores Microsoft SharePoint. Los protocolos compatibles son FTP, HTTP y HTTPS.
- Si desea un mayor control durante el proceso de carga del archivo, StyleVision ofrece la opción de cargar el archivo desde la 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 456. Para obtener información sobre esta característica consulte la sección Recursos globales 444 de este manual.

16.4.2 Abrir, Volver a cargar, Cerrar, Cerrar todos

El comando **Abrir** (**Ctrl+O**) abre un archivo SPS o PXF ya existente. Al hacer clic en este comando se abre el cuadro de diálogo Abrir 763 de Windows, donde puede seleccionar cualquier archivo con la extensión `.sps` o `.pxf`.

El comando **Volver a cargar** carga la última versión guardada en disco del archivo SPS activo. Los cambios realizados desde entonces se perderán. El archivo XML de trabajo también se carga, lo cual permite actualizar el archivo XML de trabajo si otro usuario realizó cambios en él.

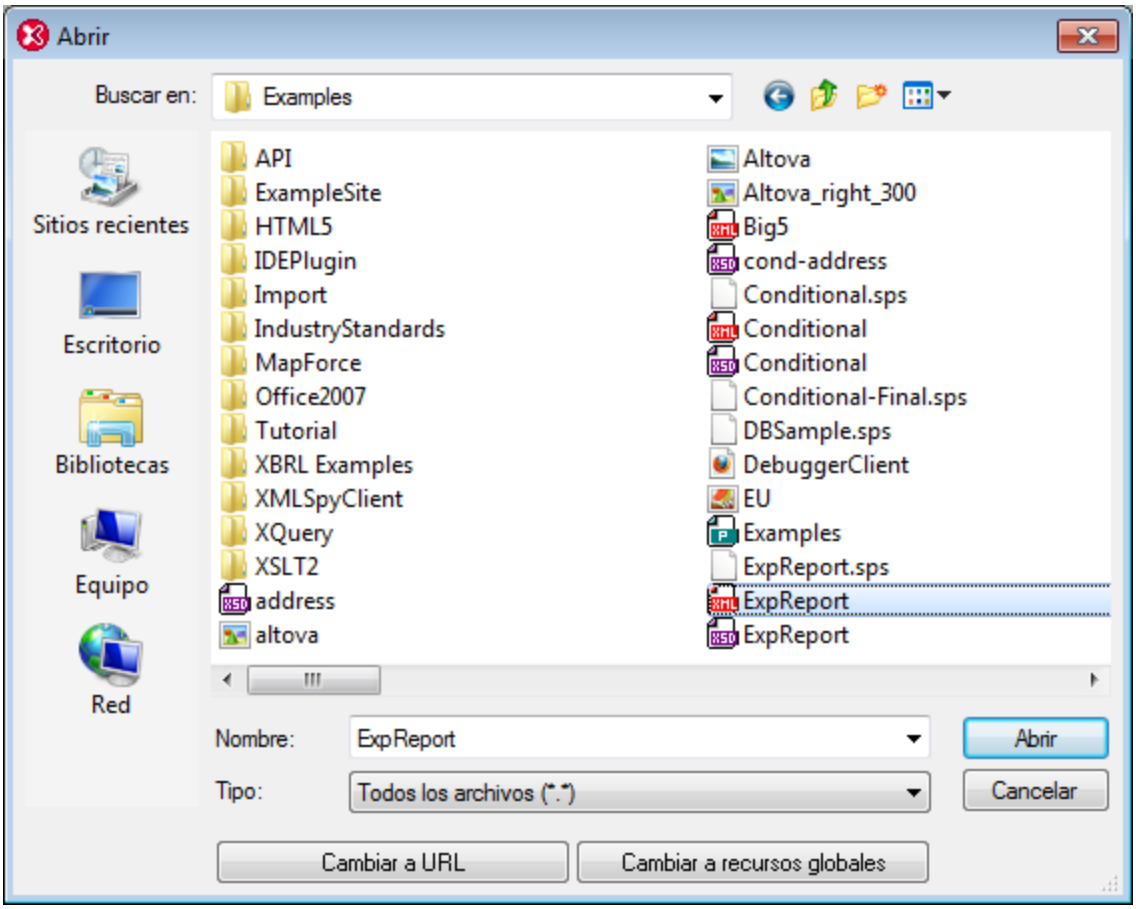
El comando **Cerrar** cierra el archivo SPS activo. Recuerde que aunque tenga varios archivos abiertos, solo uno se considera activo. También puede cerrar el documento activo haciendo clic en el botón **Cerrar** situado en la esquina superior derecha de la Ventana principal 36. Si realizó cambios en el documento y no los guardó, la aplicación le avisa de ello.

El comando **Cerrar todos** cierra todos los documentos SPS que están abiertos. Si realizó cambios en algún documento y no los guardó, aparece un aviso.

▼ Seleccionar y guardar archivos a través de direcciones URL y recursos globales

© 2016-2022 Altova GmbH

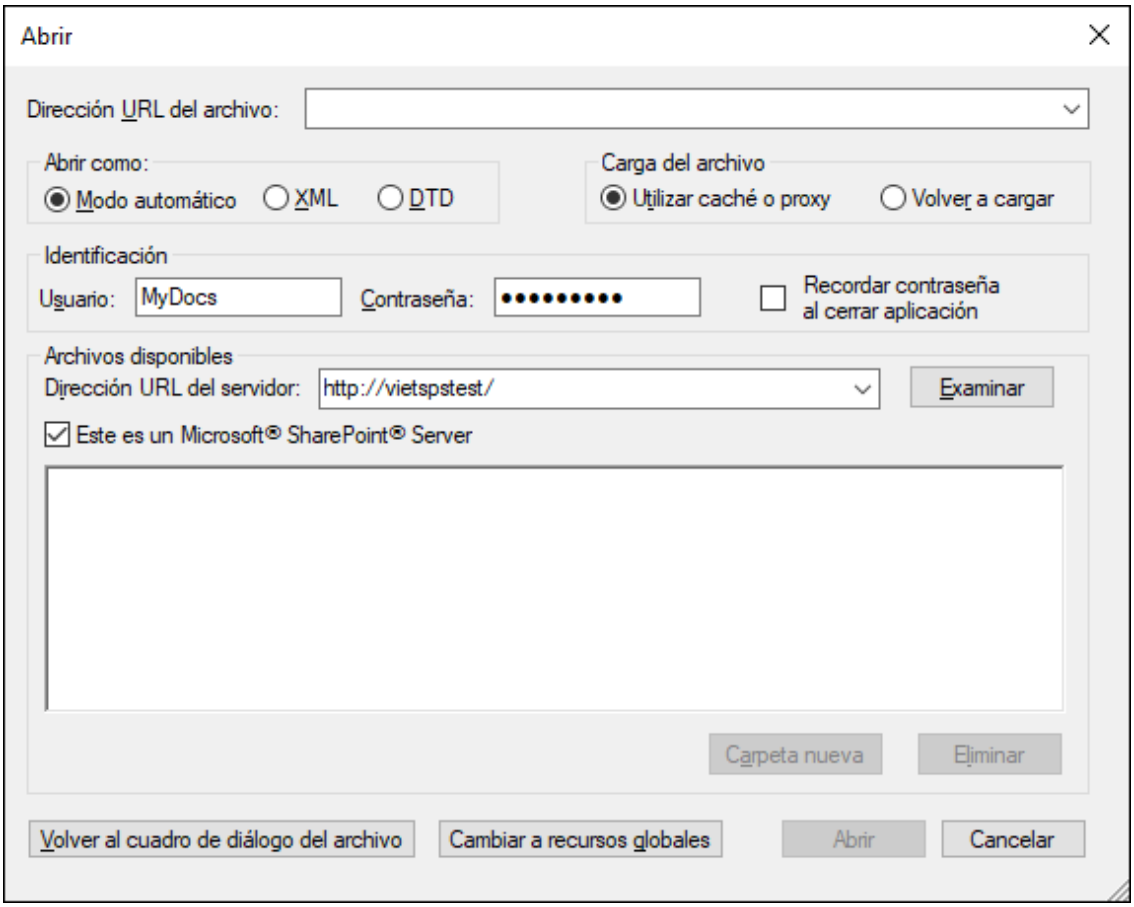
En varios cuadros de diálogo "Abrir archivo" y "Guardar archivo" de StyleVision podrá seleccionar o guardar el archivo a través de una URL o un recurso global (*imagen siguiente*). Haga clic en el botón **Cambiar a URL** o **Cambiar a recursos globales** según corresponda.



Seleccionar archivos a través de direcciones URL

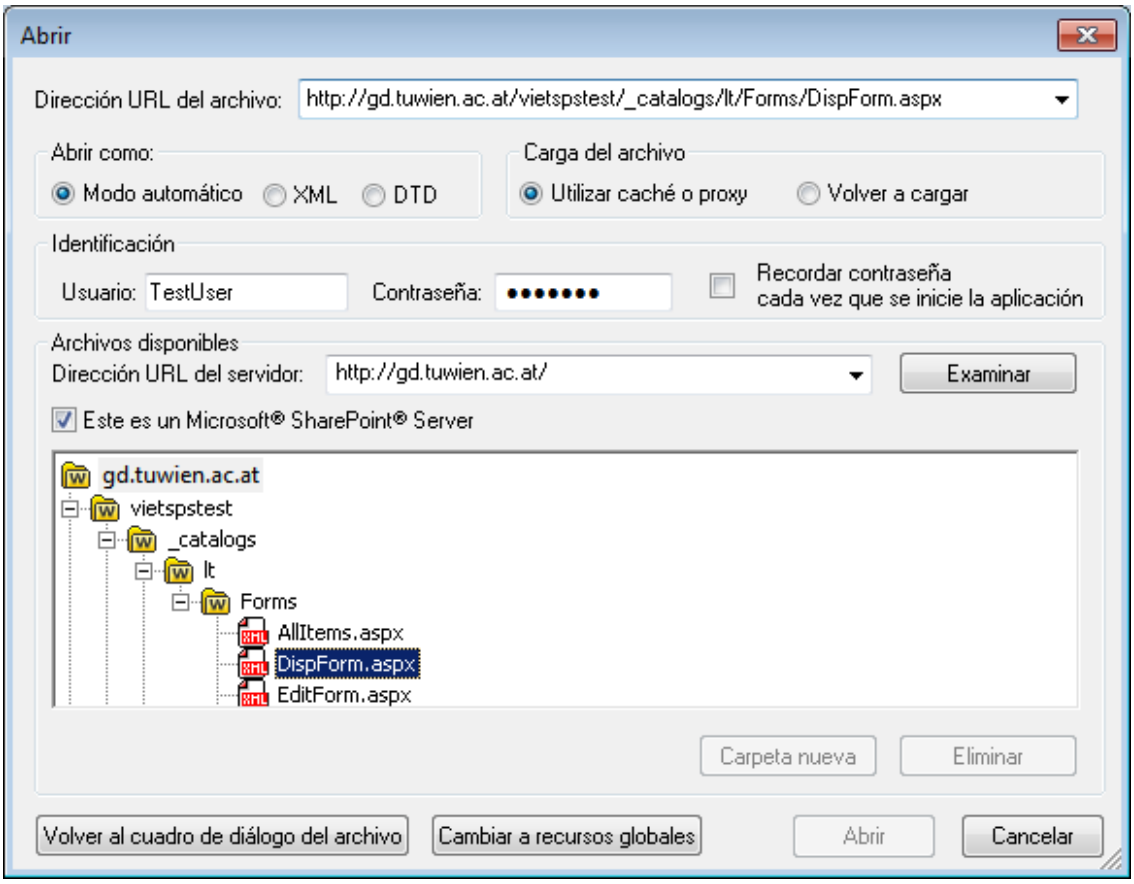
Siga estas instrucciones para seleccionar un archivo a través de una dirección URL (ya sea para abrirlo o para guardarlo):

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.

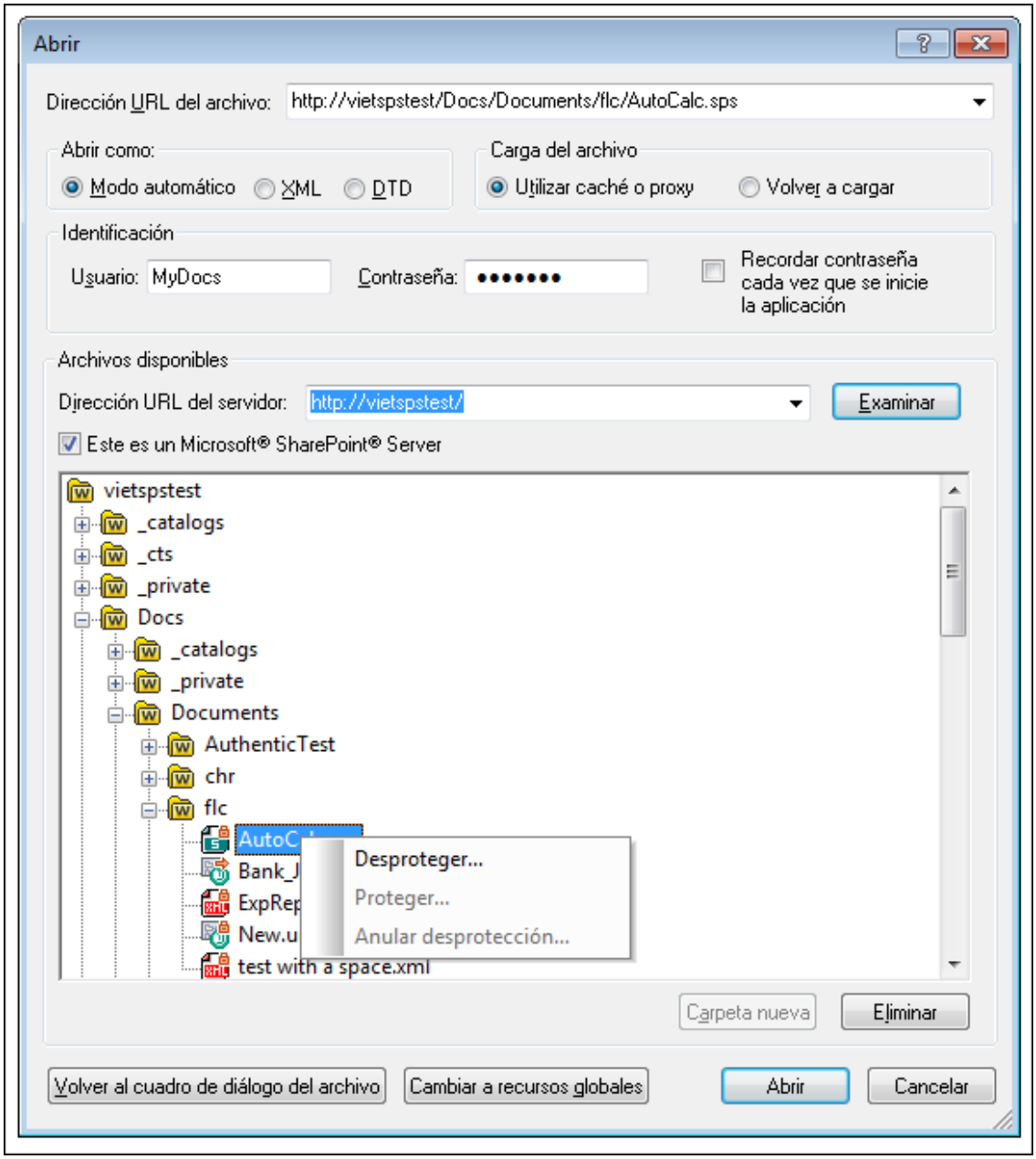
Debe tener en cuenta que:

- 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, StyleVision 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 456. Para obtener información sobre esta característica consulte la sección Recursos globales 444 de este manual.

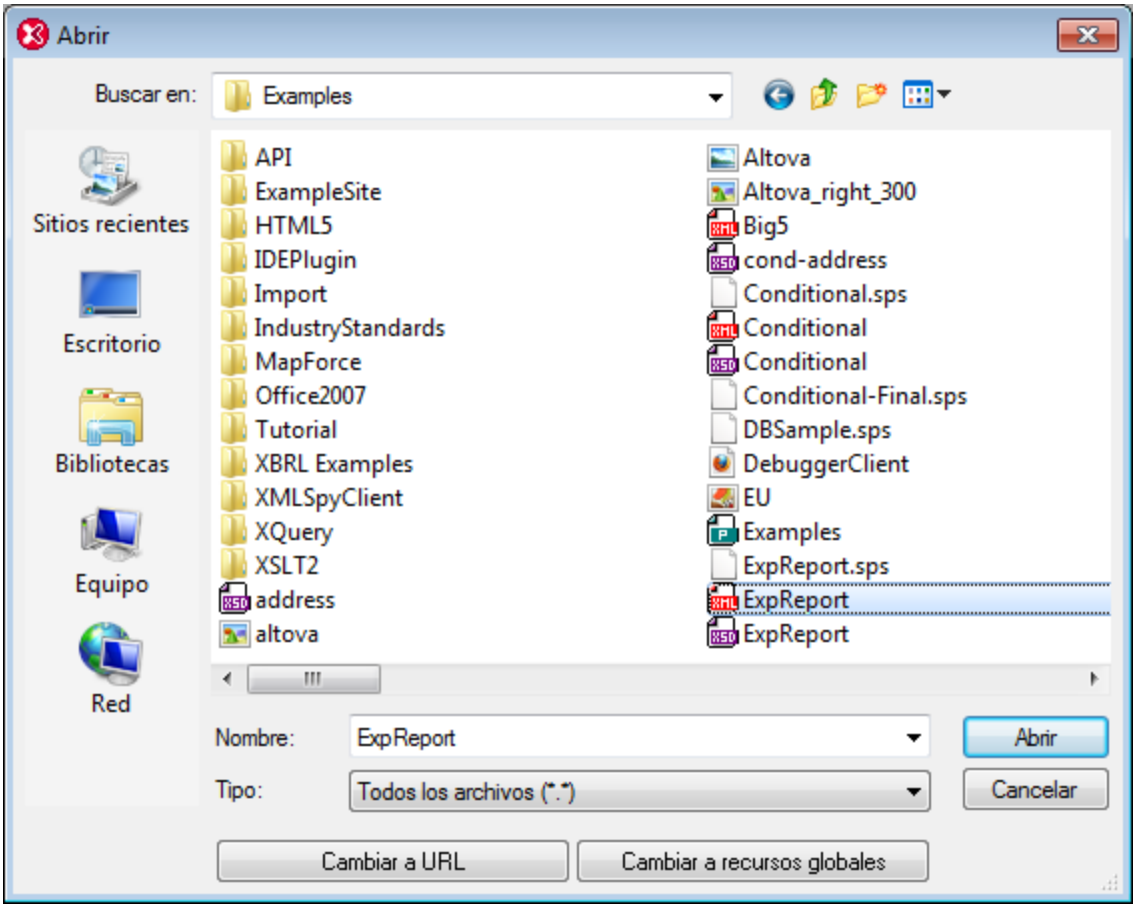
16.4.3 Guardar diseño, Guardar todos

El comando **Guardar diseño** (**Ctrl+S**) guarda el archivo activo como archivo SPS (con la extensión de archivo `.sps`).

El comando **Guardar todos** (**Ctrl+Mayús+S**) guarda todos los documentos SPS que están abiertos.

▼ Seleccionar y guardar archivos a través de direcciones URL y recursos globales

En varios cuadros de diálogo "Abrir archivo" y "Guardar archivo" de StyleVision 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.

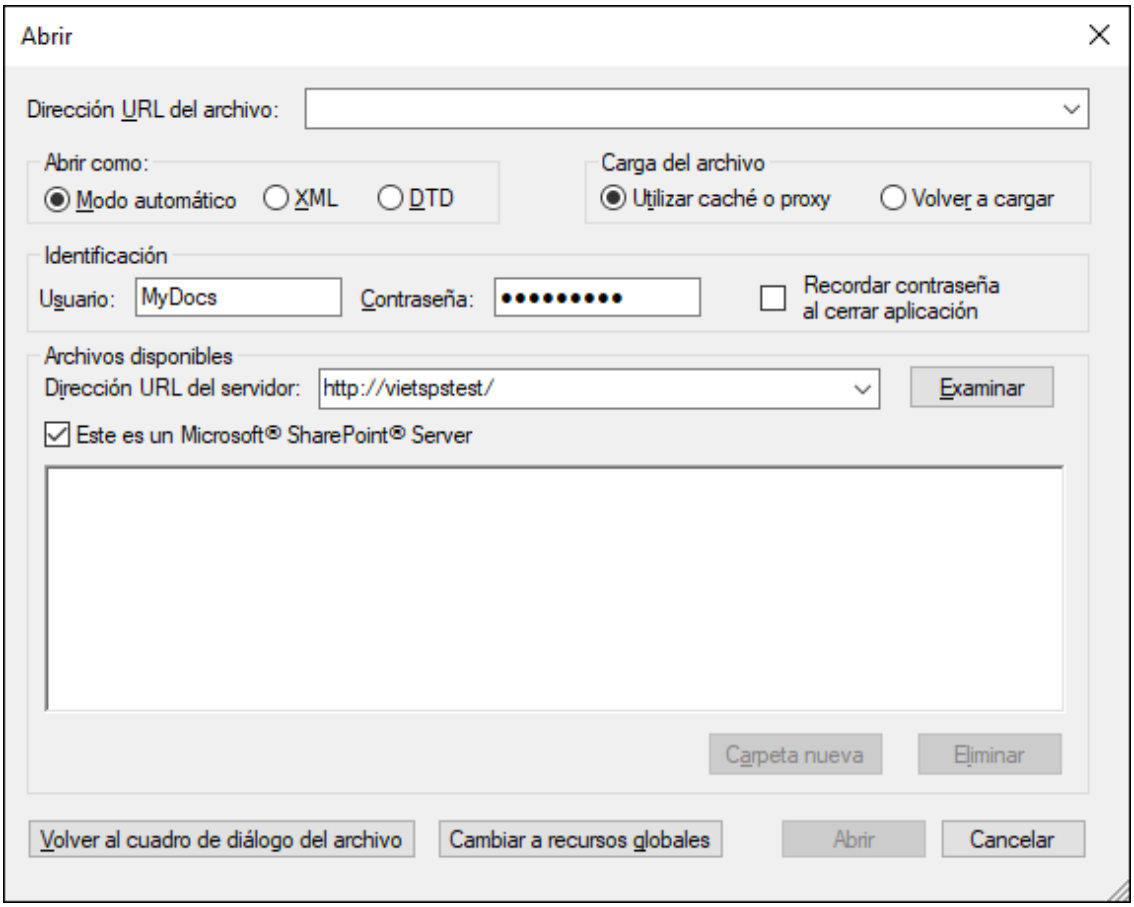


Seleccionar archivos a través de direcciones URL

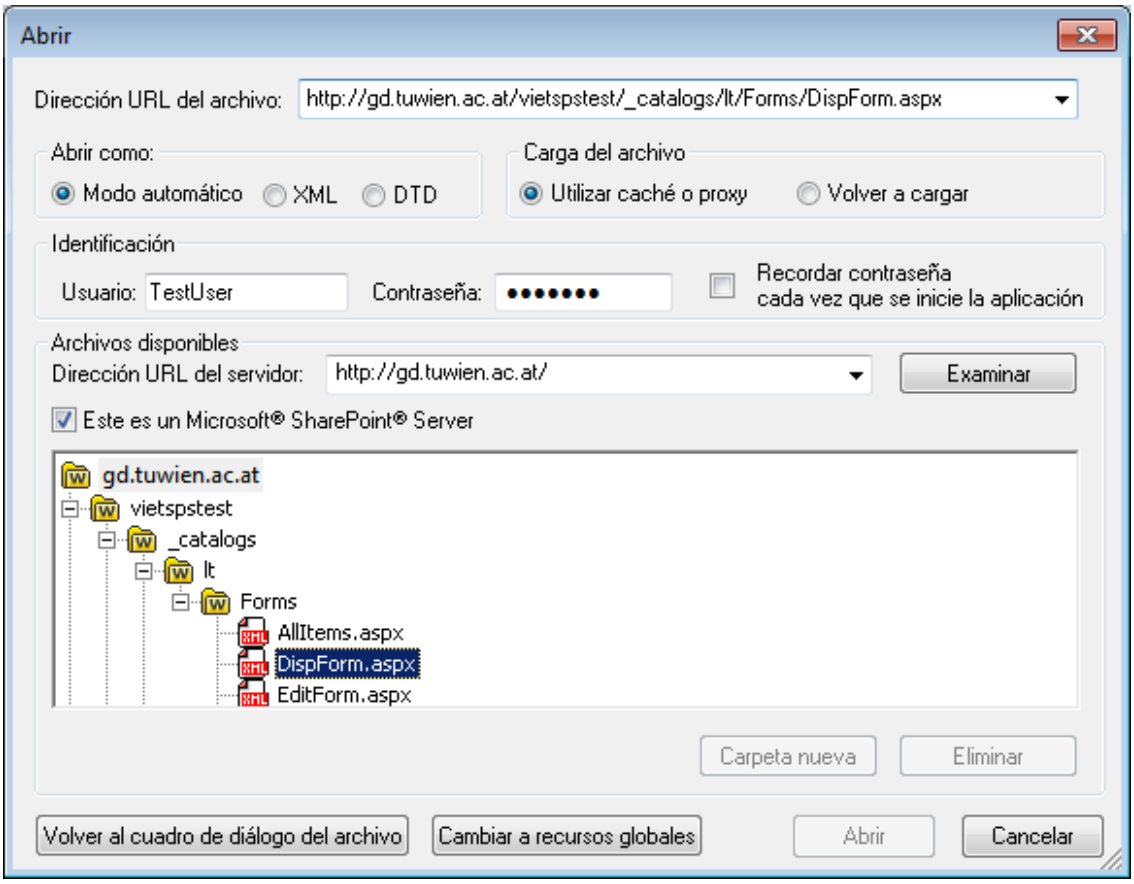
Siga estas instrucciones para seleccionar un archivo a través de una dirección URL (ya sea para abrirlo o para guardarlo):

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:

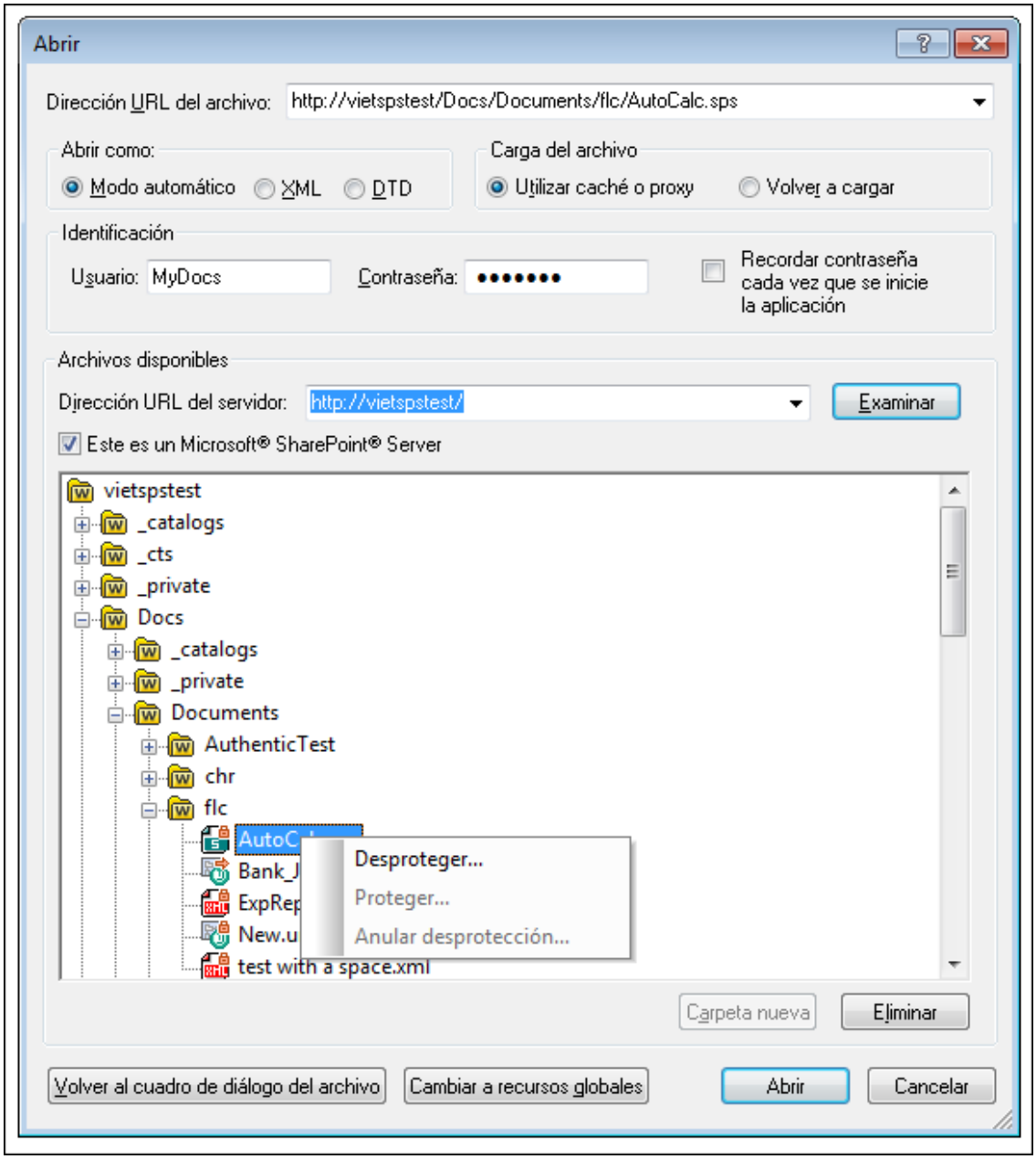
- La función **Examinar** solamente está disponible en servidores compatibles con WebDAV y servidores Microsoft SharePoint. Los protocolos compatibles son FTP, HTTP y HTTPS.
- Si desea un mayor control durante el proceso de carga del archivo, StyleVision ofrece la opción de cargar el archivo desde la 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**.

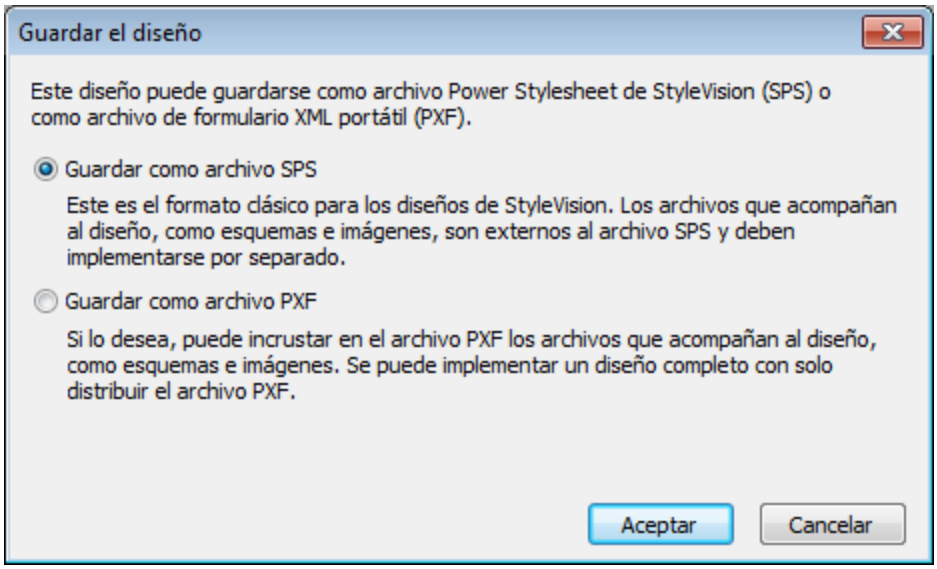
Abrir y guardar archivos a través de recursos globales

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 456. Para obtener información sobre esta característica consulte la sección Recursos globales 444 de este manual.

16.4.4 Guardar como

El comando **Guardar como** guarda el diseño (i) *como archivo SPS* o (ii) *como archivo PXF* (formulario XML portátil). Al hacer clic en el comando **Guardar como** aparece el cuadro de diálogo "Guardar el diseño" (*imagen siguiente*). Seleccione el formato que desea usar y haga clic en **Aceptar**.

© 2016-2022 Altova GmbH



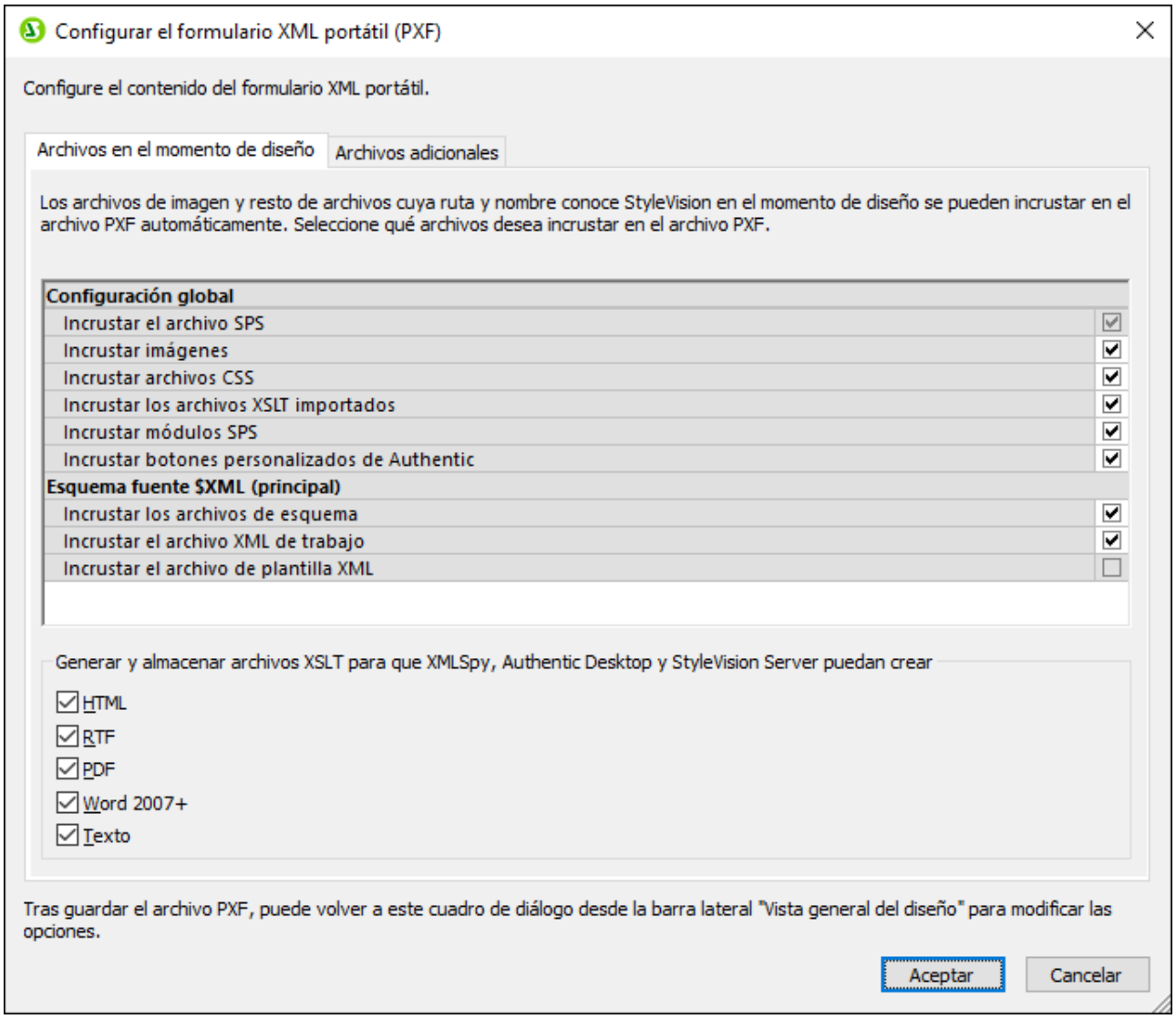
El **formato SPS** es el formato estándar de Altova para todos los diseños de StyleVision. El **formato PXF** es un formato desarrollado por Altova que permite incrustar en el diseño todos los archivos relacionados con él (archivos de esquema, archivos XML, imágenes, hojas de estilos XSLT, etc.). Este formato es ideal para transportar todos los archivos necesarios para abrir el diseño en la vista Authentic o para generar documentos de salida HTML, texto y/o RTF basados en el diseño.

Guardar como SPS

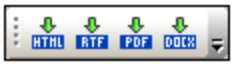
Si selecciona esta opción aparece el cuadro de diálogo "Guardar como" de Windows. Con el comando **Guardar como** también puede guardar con otro nombre los archivos guardados anteriormente.

Guardar como PXF

Si selecciona la opción *Guardar como archivo PXF* se abre el cuadro de diálogo "Guardar" de Windows, que funciona igual que con el comando **Guardar diseño** 773 solo que incluye un paso más: seleccionar los ficheros que se deben incluir en el archivo PXF. Tras indicar el nombre de archivo del PXF, aparece el cuadro de diálogo "Configurar el formulario XML portátil (PXF)" (*imagen siguiente*) donde puede seleccionar qué ficheros se incrustan en el PXF.



Bajo la entrada **`Configuración global`** de la pestaña *Archivos en el momento de diseño* puede seleccionar qué archivos relacionados con el diseño se incrustan y cuáles se omiten. También puede seleccionar los archivos XSLT generados a partir del diseño. En el panel inferior puede seleccionar para qué formatos desea generar e incrustar archivos XSLT. Si incrusta ficheros XSLT en el archivo PXF y abre el archivo PXF en la vista Authentic de un producto de Altova, se habilita el botón de la barra herramientas de Authentic que sirve para generar y ver documentos de salida (*imagen siguiente*).

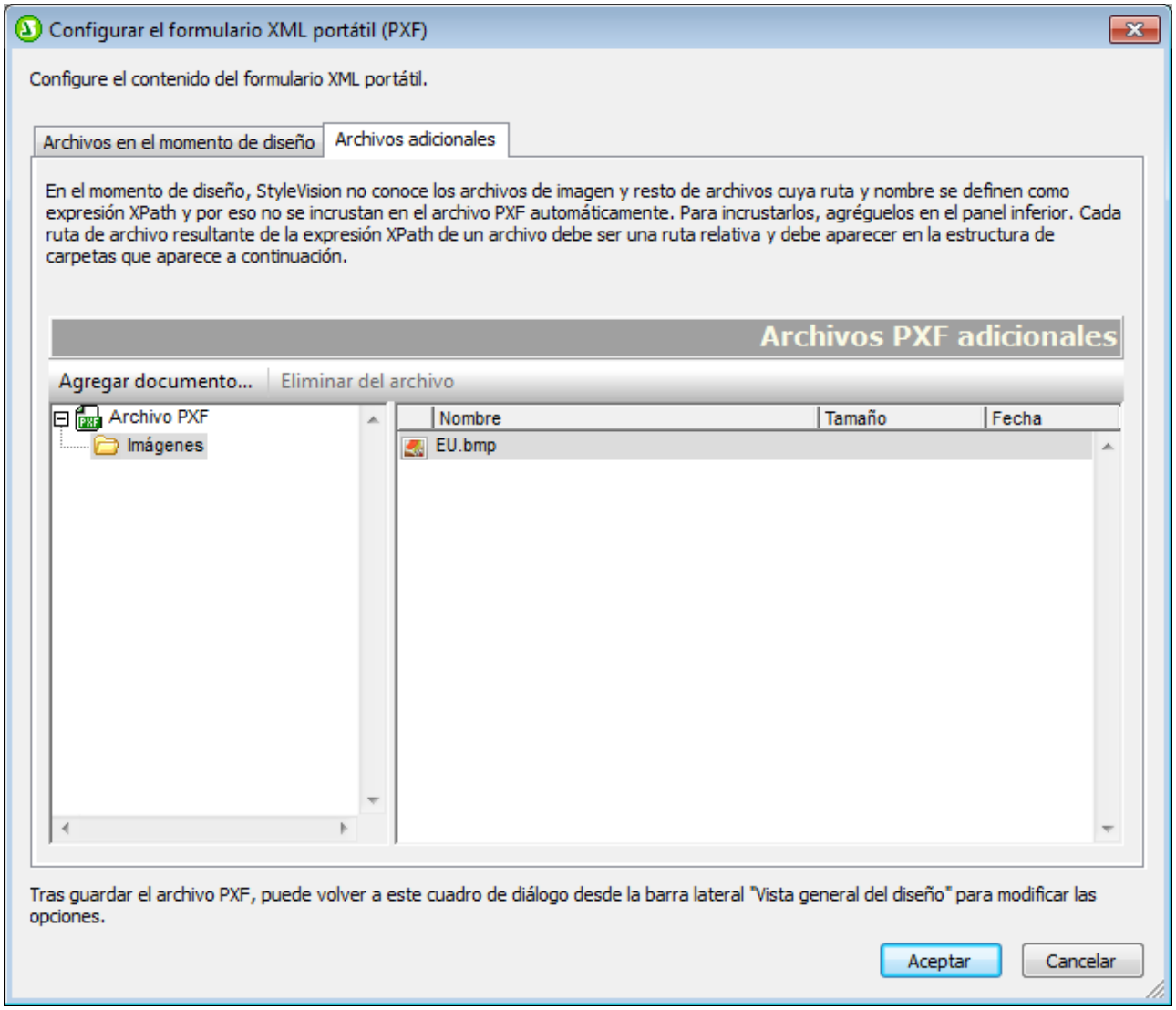


Nota: si en el PXF incrusta archivos XSLT para formatos de salida que solo son compatibles con una edición superior de StyleVision (Enterprise es superior a Professional y Professional es superior a Basic) y el PXF se abre en una edición inferior, cuando guarde el PXF los ficheros XSLT para formatos de salida incompatibles con la edición inferior no se guardarán. Aparece un aviso preguntando si quiere guardar el archivo PXF de todos modos.

En la pestaña *Archivos adicionales* (*imagen siguiente*) puede incluir más ficheros que no estén relacionados

 © 2016-2022 Altova GmbH

con el diseño pero que desee incrustar en el PXF de todos modos. Por ejemplo: imágenes a las que hace referencia el diseño a través de una URL generada con una expresión XPath. En el ejemplo siguiente el usuario seleccionó el archivo de imagen `EU.bmp` de la carpeta `Imágenes` para incrustarlo en el PXF.



Para incluir ficheros adicionales en el archivo PXF, haga clic en el botón **Agregar documento...** y busque el archivo. El cuadro de diálogo "Abrir" se abre automáticamente por la carpeta donde está el diseño SPS. Puede seleccionar ficheros de esta carpeta o de carpetas descendientes. Tras añadir el fichero al archivo XPF, el cuadro de diálogo muestra el fichero y la estructura de carpetas donde se encuentra. Por ejemplo, en la imagen anterior puede ver que el archivo `EU.bmp` está en una carpeta llamada `Imágenes`, que a su vez está en la carpeta del archivo SPS.

StyleVision devuelve un error si selecciona un archivo de una carpeta situada en un nivel superior a la carpeta donde está el diseño SPS.

En el diseño SPS las referencias a archivos adicionales deben hacerse con una ruta relativa y debe usar la estructura de carpetas que aparece en la pestaña *Archivos adicionales* (*imagen anterior*). Por ejemplo, para hacer referencia a la imagen `EU.bmp` de nuestro ejemplo debe usar la ruta relativa `Imágenes/EU.bmp`.

Nota: para poder guardar archivos PXF debe marcar la opción *Incrustar imágenes para RTF y Word 2007+* (**Archivo | Propiedades | Imágenes**).

16.4.5 Exportar como archivo de diseño de MobileTogether

Este comando genera un archivo de diseño de Altova MobileTogether a partir del diseño SPS activo. Los archivos de diseño de MobileTogether sirven para ejecutar soluciones en la aplicación MobileTogether para dispositivos móviles. Por ejemplo, una solución de MobileTogether puede abrirse en un dispositivo móvil (p. ej. un teléfono móvil inteligente) para ver y editar el contenido de una base de datos. Las soluciones de MobileTogether se diseñan en la aplicación Altova MobileTogether Designer.

Con este comando puede convertir un diseño SPS en un diseño de MobileTogether que se puede editar en Altova MobileTogether Designer. Consulte el sitio web de MobileTogether y la documentación de MobileTogether Designer para obtener más información. Las opciones de conversión se pueden configurar en la pestaña *Diseño de MobileTogether* del cuadro de diálogo "Opciones" (**Herramientas | Opciones** 894).

Nota: no todas las características de diseño de los archivos SPS tienen equivalentes en los diseños de MobileTogether. Por tanto, tras ejecutar este comando, abra el archivo generado en MobileTogether Designer, revíselo y realice los cambios que sean necesarios. Estas son las funcionalidades de diseño de los que se sabe que no se pueden exportar a los diseños de MobileTogether: (i) Fuentes más allá de la primera de la lista en el diseño de StyleVision(una funcionalidad de la versión Enterprise); (ii) plantillas globales 266; (ii) archivos SPS modulares 251.

16.4.6 Guardar datos XML de Authentic, Guardar como

En la vista Authentic puede editar el archivo XML de trabajo o la base de datos en la que está basado el SPS.

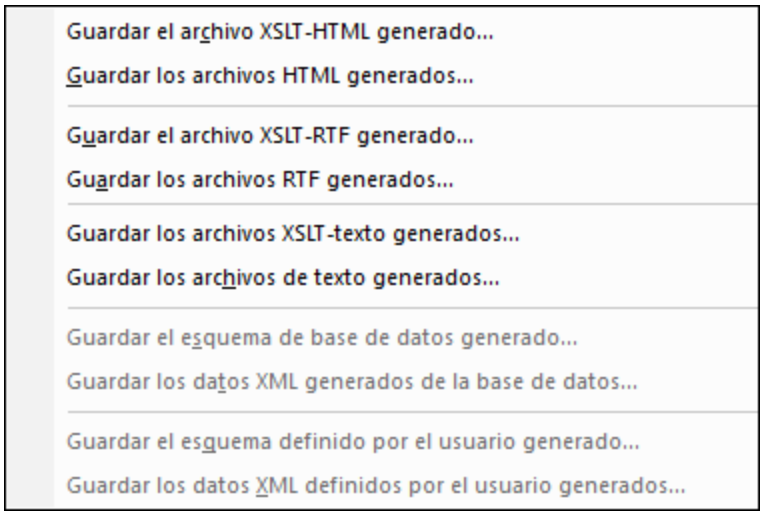
El comando **Guardar datos XML de Authentic** guarda todos los cambios en el archivo XML de trabajo o en la BD propiamente dichos. Es decir, si no quiere editar el archivo XML en StyleVision, puede editar el documento XML o la BD en la vista Authentic de XMLSpy o de Authentic Desktop.

El comando **Guardar datos XML de Authentic como** sirve para guardar el documento XML modificado en la vista Authentic con otro nombre.

16.4.7 Guardar archivos generados

El comando **Guardar archivos generados** abre un submenú (*imagen siguiente*) que incluye opciones para guardar los tipos de archivo que aparecen a continuación. Para entender qué son exactamente y cómo usarlos consulte el apartado Resumen de características | Archivos generados 121.

 © 2016-2022 Altova GmbH



Guardar el archivo XSLT-HTML generado

Genera un archivo XSLT para formatos de salida HTML a partir del diseño SPS. El archivo XSLT generado se puede usar para transformar documentos XML en documentos HTML.

Guardar los archivos HTML generados

Genera un archivo HTML o varios archivos HTML si el diseño está dividido para múltiples documentos de salida 284. Para esta operación son necesarios dos archivos de entrada:

- El archivo XML de trabajo asignado al archivo SPS activo. Si no se asignó ninguno, el comando se deshabilita. Si se trata de un archivo SPS basado en una BD, el archivo XML no editable generado automáticamente es el archivo XML de trabajo (consulte Esquema de BD y archivos XML de BD 620). Es decir, en este caso no hace falta asignar un archivo XML de trabajo.
- El archivo XSLT-HTML que se genera automáticamente a partir del archivo SPS activo.

Guardar el archivo XSLT-RTF generado

Genera un archivo XSLT para formatos de salida RTF a partir del diseño SPS. El archivo XSLT generado se puede usar para transformar documentos XML en documentos RTF.

Guardar los archivos RTF generados

Genera un archivo RTF o varios archivos RTF si el diseño está dividido para múltiples documentos de salida 284 . Para esta operación son necesarios dos archivos de entrada:

- El archivo XML de trabajo asignado al archivo SPS activo. Si no se asignó ninguno, el comando se deshabilita. Si se trata de un archivo SPS basado en una BD, el archivo XML no editable generado automáticamente es el archivo XML de trabajo (consulte Esquema de BD y archivos XML de BD 620). Es decir, en este caso no hace falta asignar un archivo XML de trabajo.
- El archivo XSLT-RTF que se genera automáticamente a partir del archivo SPS activo.

Guardar los archivos XSLT-texto generados

El comando **Guardar los archivos XSLT-texto generados** genera un archivo XSLT para salida texto a partir del SPS. Después puede usar este archivo XSLT para transformar un documento XML en un documento de texto.

Guardar los archivos de texto generados

El comando **Guardar los archivos de texto generados** genera un archivo de texto. Si en el diseño se indica que debe haber múltiples documentos de salida 284 se generan varios archivos de texto. Para ello son necesarios dos archivos de entrada:

- El archivo XML asignado al archivo SPS activo. Si no se asignó ninguno el comando se deshabilita. Para archivos SPS basados en bases de datos, el archivo XML no editable que se genera automáticamente es el archivo XML asignado (consulte Esquema XML y archivos XML de la BD 620); es decir, no tiene que asignar un archivo XML.
- Un archivo XSLT-para-texto, que se genera automáticamente a partir del archivo SPS activo cuando se ejecuta este comando.

Guardar el esquema de base de datos generado

Cuando se conecta a una BD para crear un archivo SPS basado en una BD, StyleVision genera y carga un esquema XML temporal basado en la estructura de la BD. Con el comando **Guardar el esquema de base de datos generado** puede guardar este esquema XML generado por StyleVision. Recuerde que para las BD XML StyleVision no genera un archivo de esquema, sino que usa un archivo de esquema de la BD o de alguna otra ubicación. Por tanto, este comando se deshabilita si está trabajando con una BD XML.

Guardar los datos XML generados de la BD

Genera y guarda un archivo XML que contiene datos de la BD en una estructura XML acorde con a la estructura del esquema XML generado a partir de la BD. Si definió filtros de BD en el diseño, los filtros se aplican durante la importación de datos. Recuerde que para las BD XML StyleVision no genera un archivo de esquema, sino que usa datos XML de las columnas XML de la BD XML. Por tanto, este comando se deshabilita si está trabajando con una BD XML.

Guardar el esquema definido por el usuario generado

Este comando se habilita si el diseño SPS utiliza un esquema definido por el usuario. El esquema creado en la ventana Estructura del esquema se guarda como esquema XML con la extensión de archivo `.xsd`.

Guardar los datos XML definidos por el usuario generados

Los datos del archivo HTML importado que corresponden al esquema definido por el usuario se guardan como archivo XML. Los datos correspondientes son los nodos del documento HTML (de la vista **Diseño**) que se crearon como nodos del esquema XML.

 © 2016-2022 Altova GmbH

16.4.8 Implementar en FlowForce

El comando **Implementar en FlowForce** implementa un archivo de transformación `.transformation` en un servidor Altova FlowForce Server. El archivo `.transformation` contiene todos los ficheros e información necesarios para realizar las transformaciones diseñadas en el SPS. Después de implementar el archivo `.transformation` en FlowForce Server, podrá crear trabajos que usen este archivo para generar transformaciones en función de los desencadenadores especificados en la definición del trabajo. Para más información consulte la documentación de Altova FlowForce Server.

Los archivos de transformación `.transformation` se generan a partir de un archivo PXF. De modo que el comando **Implementar en FlowForce** solo funciona si el archivo activo es un archivo PXF. Si el archivo activo es un SPS, el comando **Implementar en FlowForce** está habilitado pero al hacer clic en él aparece un aviso solicitando guardar el archivo SPS como PXF. Para crear un archivo PXF a partir de un archivo SPS, utilice el comando **Archivo | Guardar como** y seleccione el formato PXF.)

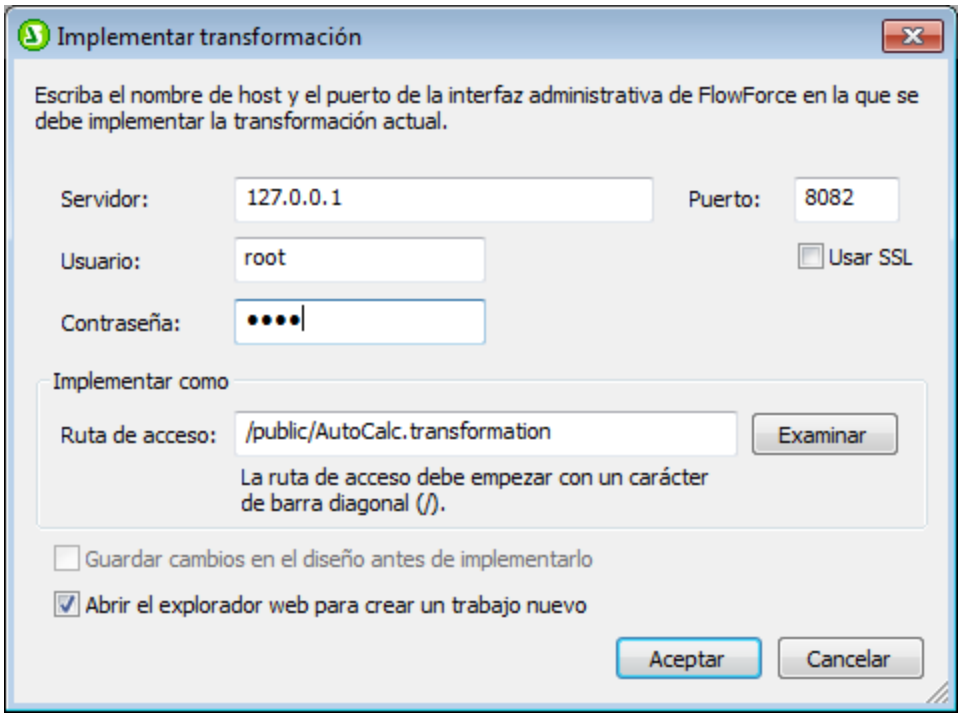
Tenga en cuenta que:

- al guardar el archivo PXF hay una opción para incluir archivos externos (p. ej. imágenes) en el archivo. Si un archivo externo no está incluido en el archivo PXF pero se necesita para la transformación, el archivo externo debe estar guardado en FlowForce Server. Los archivos externos se abren desde el directorio de trabajo (especificado en la definición del trabajo de FlowForce) y por eso deben estar en una posición relativa al directorio de trabajo, para que los vínculos procedentes del directorio de trabajo puedan acceder a ellos correctamente.
- cuando se ejecuta un trabajo de FlowForce que necesita ejecutar una transformación de StyleVision, el trabajo se pasa a StyleVision Server, que a su vez extrae el contenido del archivo PXF en el directorio de trabajo que se indicó en los parámetros del trabajo. Para evitar conflictos entre los nombres de archivo, el directorio de trabajo no debería contener archivos llamados igual que los archivos incluidos en el archivo PXF.

Antes de ejecutar el comando **Implementar en FlowForce**, compruebe que Altova FlowForce Server y Altova StyleVision Server tienen asignada una licencia y están en ejecución. Para más información consulte la documentación de Altova FlowForce Server.

Comando Implementar en FlowForce Server

Este comando abre el cuadro de diálogo "Implementar transformación" (*imagen siguiente*).



En este cuadro de diálogo puede especificar:

- La dirección y el número de *puerto* del *servidor* web de FlowForce (no del servidor FlowForce Server) así como los datos de acceso del servidor (el *usuario* y la *contraseña*).
- El nombre del archivo de transformación y la ubicación donde debe guardarse en el servidor FlowForce Server. La ruta de acceso debe empezar con una barra diagonal, que representa el directorio raíz de FlowForce Server.
- Si el diseño tiene cambios sin guardar, se habilita la casilla *Guardar cambios en el diseño antes de implementarlo*. Marque la casilla si quiere que se guarden los cambios. De lo contrario, deje la casilla sin marcar.
- Para implementar el diseño a través de una conexión con cifrado SSL, marque la casilla *Usar SSL.* Esto da a entender que FlowForce Server ya está configurado para aceptar conexiones SSL. Consulte la documentación de FlowForce Server para obtener más información.

Al hacer clic en **Aceptar** en el cuadro de diálogo "Implementar transformación", el archivo `.transformation` se implementa en la ubicación indicada del servidor FlowForce Server. Si marcó la casilla *Abrir el explorador web para crear un trabajo nuevo* (*imagen anterior*), se abre la interfaz del servidor web FlowForce Server en una ventana del explorador web, donde puede editar el trabajo creado durante el paso de implementación.

Varias versiones de StyleVision Server

Si el servidor donde implementa el archivo `.transformation` ejecuta varias versiones de StyleVision Server bajo la dirección de FlowForce Server (solo en servidores Windows), entonces aparece un cuadro de diálogo llamado "Seleccione StyleVision Server", donde debe especificar la versión de StyleVision Server con la que desea ejecutar esta asignación. Puede seleccionar a mano la versión que desea utilizar o dejar que el servidor seleccione automáticamente la versión más adecuada.

Este cuadro de diálogo aparece cuando el directorio de instalación de FlowForce Server contiene un archivo `.tool` por cada versión de StyleVision Server que se ejecuta bajo la dirección de FlowForce Server. El archivo

 © 2016-2022 Altova GmbH

`.tool` de StyleVision Server `.tool` se añade por defecto a este directorio cuando se instala StyleVision Server con el paquete de instalación de FlowForce Server. La ruta de acceso donde se almacenan los archivos `.tool` e FlowForce es: `C:\Archivos de programa\Altova\FlowForceServer2023\tools`. Si tiene más versiones de StyleVision Server y desea ejecutarlas bajo la dirección de FlowForce Server, deberá copiar a mano sus archivos `.tool` al directorio recién mencionado. El archivo `.tool` de StyleVision Server se encuentra en `C:\Archivos de programa\Altova\StyleVisionServer2023\etc`.

Para más información consulte la documentación de Altova FlowForce Server.

16.4.9 Diseño web

El comando **Diseño web** abre un submenú que incluye el comando **Generar aplicación web ASPX**. Este comando genera todos los archivos necesarios para ejecutar una aplicación ASPX en la carpeta de destino indicada. El explorador web leerá el archivo ASPX, que es el documento de salida. El código C# de este archivo iniciará un proceso en el que los datos de la BD o del archivo XML fuente se transforman de forma dinámica con ayuda del archivo XSLT incluido en el paquete ASPX. Este archivo ASPX (el documento de salida del proceso de transformación) se actualiza con los datos más recientes de la BD o del archivo XML fuente

Consulte el apartado Interfaz ASPX para aplicaciones web 509 para obtener más información.

16.4.10 Propiedades

El comando **Propiedades** abre el cuadro de diálogo "Propiedades", donde puede configurar las propiedades del archivo SPS activo.

Formato de salida

En la pestaña *Formato de salida* puede configurar:

- ***Codificación de salida:*** en este panel puede seleccionar la codificación de los documentos de salida del archivo SPS activo. Si cambia la codificación en este panel, cambia la codificación del archivo SPS activo. También puede especificar la codificación predeterminada 894 de todos los documentos SPS que se creen de ahora en adelante, pero esto se hace en la pestaña *Codificación* del cuadro de diálogo "Opciones".

- ***Modo de salida HTML:*** aquí puede seleccionar si se genera un documento entero o solamente los elementos secundarios del elemento HTML `body`. Los elementos secundarios se generan en paralelo en el fragmento de salida, es decir, en el mismo nivel y contendrán todos los descendientes recursivamente. En otras palabras, el documento de salida puede ser un fragmento de código HTML.

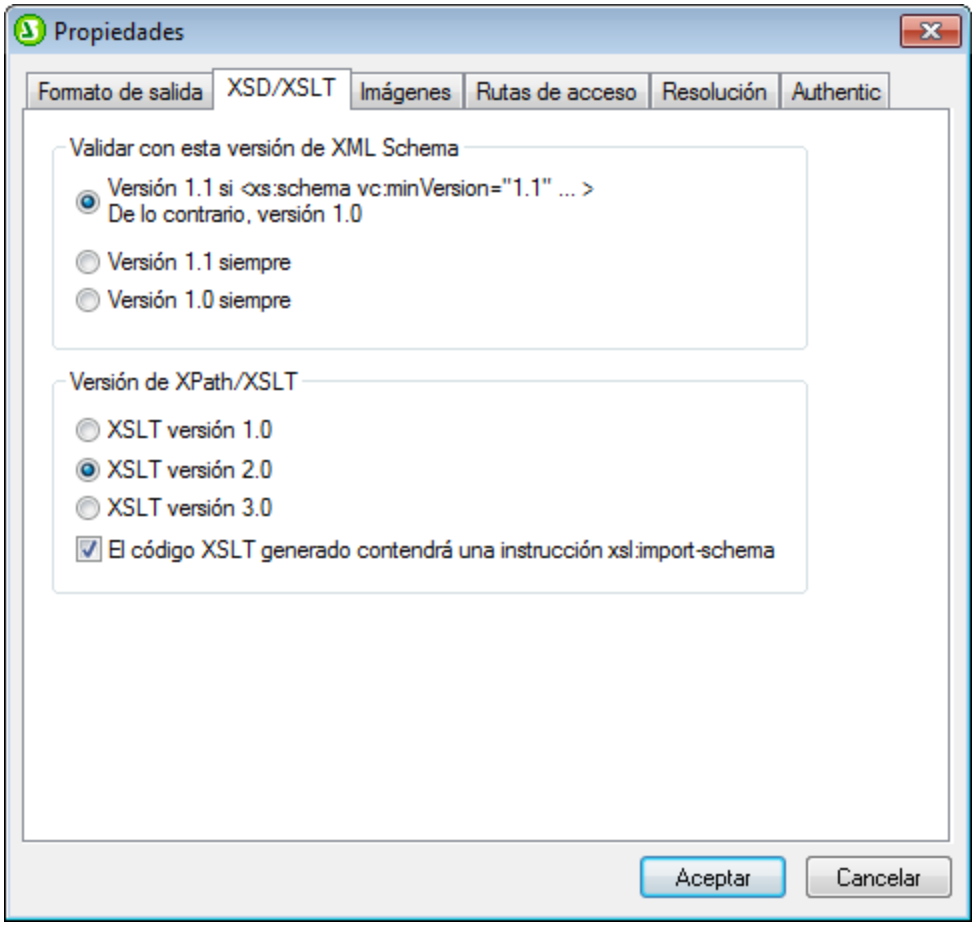
- ***Modo de salida HTML (DOCTYPE):*** en este panel puede seleccionar el tipo de documento HTML que se genera: HTML5, de transición HTML 4.01 o de transición XHTML 1.0. Esta opción de configuración puede cambiarse en cualquier momento.

- ***Versión de Internet Explorer compatible:*** en las versiones de Internet Explorer anteriores a Internet Explorer 6 la característica CSS no era totalmente compatible y a veces había algunos problemas. A partir de la versión IE6 estos problemas se han solucionado y la compatibilidad con CSS ha mejorado.

Por tanto, en este panel puede seleccionar con qué versión de IE debe ser compatible el diseño SPS: IE 5, IE 7, o IE 9. (Si selecciona la opción *IE 9* debe tener instalado IE9 o superior.) El nivel de compatibilidad seleccionado estará disponible inmediatamente en la vista Authentic y en la vista previa HTML. Los documentos SPS nuevos se crean con compatibilidad para IE7. Los documentos SPS creados en versiones antiguas de Altova StyleVision se pueden guardar otra vez con el modo de compatibilidad deseado.

XSD/XSLT

En esta pestaña puede indicar qué validador XSD se utiliza para la validación XML y qué versión XSLT se usa en el archivo SPS.



StyleVision tiene un validador XSD 1.0 y un validador XSD 1.1 y puede elegir una de estas tres opciones:

- Usar el validador XSD 1.1 si el atributo `/xs:schema/@vc:minVersion` del documento XSD tiene el valor `1.1`. De lo contrario, usar el validador XSD 1.0.
- Usar el validador XSD 1.1 siempre.
- Usar el validador XSD 1.0 siempre.

Seleccione la versión XSLT para el documento activo en esta pestaña. Marque la opción de la instrucción *xsl:import-schema* para incluir el elemento `xsl:import-schema` de las especificaciones XSLT 2.0 y XSLT 3.0 en el documento XSLT generado por StyleVision. Recomendamos marcar esta opción para que los tipos de

 © 2016-2022 Altova GmbH

datos puedan leerse en el esquema en caso de que en el documento XML no exista el atributo `xsi:schemaLocation`.

Para la salida en formato texto debe indicar:

- Los caracteres de fin de línea que quiere usar. Los predeterminados son **`\r\n`** (que indican los caracteres de retorno de carro y cambio de renglón en sistemas Windows).
- El carácter que indica el final de una celda de tabla. El carácter predeterminado es el punto y coma.

Imágenes

En la pestaña *Imágenes* puede configurar si las imágenes se enlazan o se incrustan en los documentos de salida RTF. Las imágenes se pueden incrustar si utiliza la versión XSLT 2.0. Esta opción se debe configurar para cada diseño SPS por separado. Siga estas instrucciones para incrustar las imágenes en los documentos de salida:

1. Abra el cuadro de diálogo "Propiedades" del documento SPS activo (**Archivo | Propiedades**).
2. Marque la casilla *Incrustar imágenes para RTF y Word 2007+* (marcada por defecto). Recuerde que las imágenes se pueden incrustar solamente si la versión XSLT del diseño SPS activo es XSLT 2.0.
3. Haga clic en **Aceptar** y guarde el archivo SPS. La opción queda guardada para el archivo SPS activo.

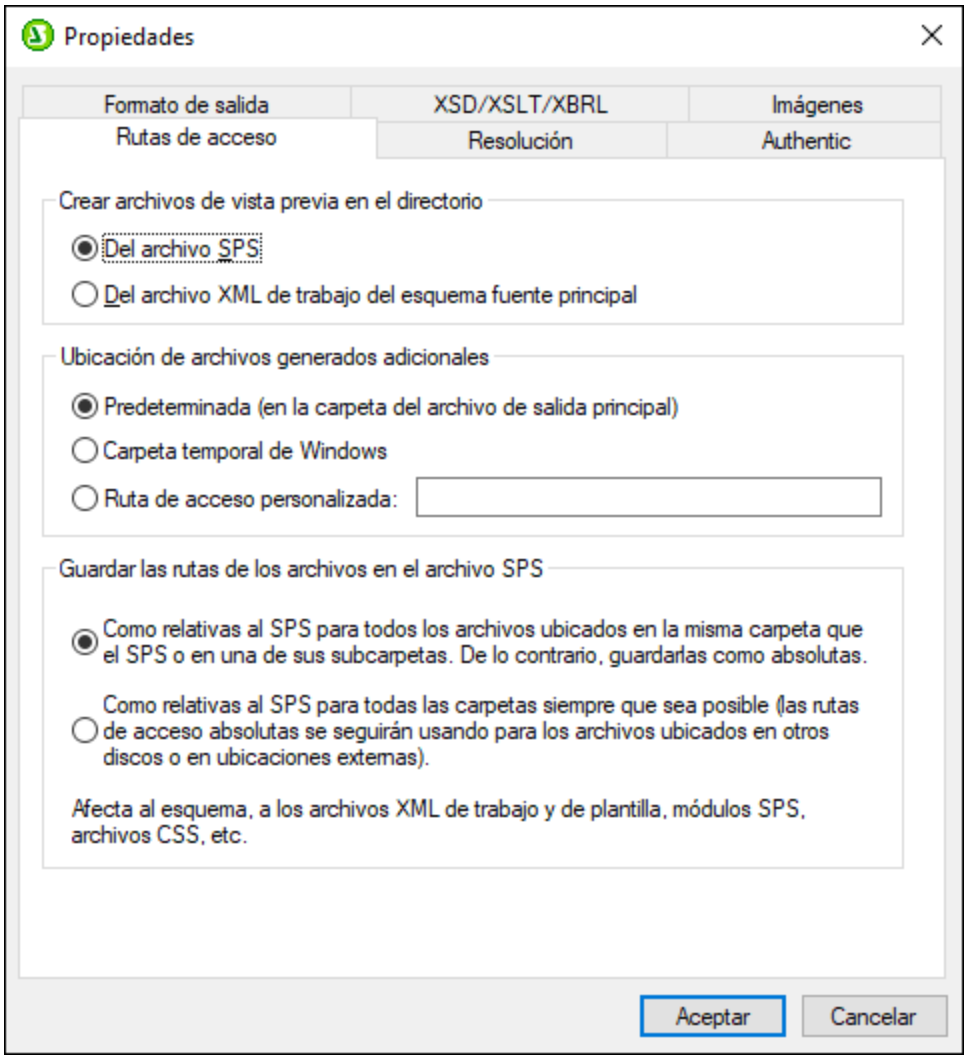
Si no marca la casilla *Incrustar imágenes*, las imágenes se enlazan usando la ruta de archivo de imagen especificada en las propiedades de la imagen (seleccione la imagen y después seleccione la propiedad `URL` en la ventana Propiedades 59). El apartado Imágenes: datos incrustados y URI 183 explica cómo se resuelven las rutas de acceso.

Para poder guardar archivos PXF 512 es obligatorio marcar esta opción.

Nota: en los documentos de salida RTF solamente se pueden incrustar archivos de imagen EMF, JPG y PNG.

Rutas de acceso

En la pestaña *Rutas de acceso* se especifican las rutas de acceso predeterminadas para los archivos que genera el archivo SPS y las rutas de acceso que se guardan en él (*en la imagen siguiente se ve el cuadro de diálogo de la edición Enterprise Edition*).



Estas son las propiedades que puede seleccionar en la pestaña *Rutas de acceso*:

- Ubicación donde se deben crear los archivos de vista previa: en el directorio del SPS o en el directorio del archivo XML de trabajo del esquema fuente principal.
- Ubicación donde se deben crear los archivos generados adicionales (archivos de imagen, de código de barras, gráficos, etc.).
- Si las rutas de acceso del SPS son relativas cuando el directorio de destino está en el árbol de directorios del archivo SPS o si son relativas incluso cuando el directorio de destino está fuera del árbol de directorios del archivo SPS.

Resolución

En la pestaña *Resolución* puede especificar el factor de resolución, que resuelve cómo se convierten las longitudes especificadas en píxeles en unidades de longitud para impresión (no para pantallas). La lista desplegable del cuadro combinado incluye estas opciones: **72ppp**, **96ppp** y **120ppp**. Cuanto mayor sea la medida ppp, menor será cada píxel y la longitud absoluta correspondiente. Los factores de conversión (para obtener el número de puntos en función del número de píxeles) para es 1.0, 0.75 y 0.6 para cada opción respectivamente. Consulte el apartado Resolución en píxeles 437 para obtener más información.

Authentic

En la pestaña *Authentic* puede definir estas propiedades para el diseño SPS:

- ***Modo predeterminado para operaciones de pegado en Authentic:*** con este grupo de opciones puede indicar si el texto seleccionado en la vista Authentic y copiado en el portapapeles se pega como XML o como texto. Si se pega como XML y la selección contiene marcado, entonces se pega con marcado (con etiquetas XML). De lo contrario, el modo de pegado predeterminado copia el contenido de texto de los nodos sin marcado. El modo de pegado predeterminado se puede invalidar en la vista Authentic haciendo doble clic en el punto de inserción y seleccionando el comando **Pegar como | XML** o **Pegar como | Texto** en el menú contextual. El modo de pegado predeterminado puede cambiarse en cualquier momento mientras se edita el documento SPS.

  También puede indicar si, cuando se arrastra y suelta texto en la vista Authentic, el usuario tiene la posibilidad de seleccionar cómo se pega el texto o si se usa el modo de pegado predeterminado. Para dar al usuario la opción de decidir el modo de pegado, seleccione el botón de opción *Preguntar en cada operación de arrastrar y colocar*. Para que se utilice el modo de pegado predeterminado sin preguntar antes al usuario, seleccione el botón de opción *Usar el modo predeterminado para operaciones de pegado.*

- ***Scripting con Authentic:*** estas son las opciones para los scripts que están guardados en el SPS: (i) puede habilitar o no los scripts; (ii) puede procesar o deshabilitar los eventos de los scripts y (iii) las macros de los scripts se pueden ejecutar en modo de depuración desde fuera de StyleVision (es decir, desde la vista Authentic de otros productos de Altova) o se pueden deshabilitar. Esta opción es muy práctica si quiere probar y depurar una macro fuera del entorno de StyleVision.

- ***Profundidad máxima de llamada a plantillas:*** en este cuadro puede indicar el número máximo de llamadas a plantilla que se pueden hacer de forma recursiva en los documentos de salida. Si el número de llamadas a plantillas supera el número especificado aquí, StyleVision emite un error. Esta opción se puede usar para evitar que la vista Authentic entre en un bucle infinito.

16.4.11 Vista previa de impresión, Imprimir

El comando **Vista previa de impresión** se habilita en la vista **Diseño** y en la vista Authentic (*la vista Authentic está disponible en las ediciones Enterprise y Professional solamente*). Al hacer clic en este comando se abre una ventana con una vista previa del diseño SPS (si está activa la vista **Diseño**) o de la vista Authentic del archivo XML de trabajo (si está activa la vista Authentic). La vista previa de impresión muestra el diseño con o sin etiquetas, dependiendo de la vista.

La ventana "Vista previa de impresión" ofrece una barra de herramientas con barios comandos (*imagen anterior*) y varios iconos de navegación situados al final de la página. Los comandos de la barra de herramientas, de izquierda a derecha, son estos:

- Imprimir la página con el botón **Imprimir**.

- Orientación vertical u horizontal para la página.
- Definir las propiedades de la página con el botón **Configurar página**.
- Activar/desactivar la visualización/impresión de encabezados y pies de página.
- Ajustar la vista para que el ancho de página o el alto de página se ajuste al ancho o al alto de la pantalla respectivamente.
- Establecer cuántas páginas deben aparecer en la pantalla.
- Cambiar el nivel de zoom de las páginas de la vista previa.

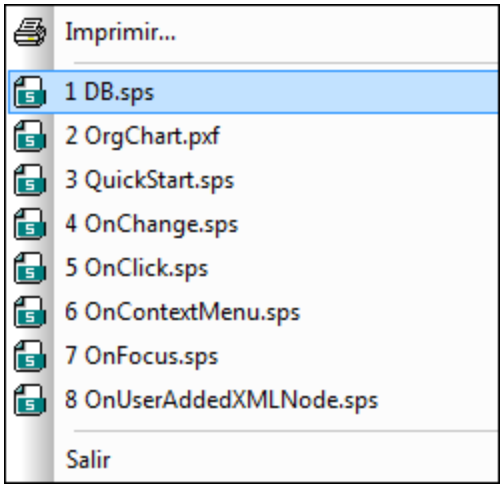
Para desplazarse por las páginas de la vista previa puede usar los botones de navegación situados en la parte inferior de la ventana o escribir el número de la página en el cuadro *Página*.

El comando **Imprimir** se habilita en la vista Authentic y en las vistas de resultados. Al hacer clic en este comando se imprime la vista seleccionada del archivo XML de trabajo según la configuración de página elegida para la vista. Recuerde que la configuración de página de la vista Authentic se puede modificar (desde la ventana "Vista previa de impresión").

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) Después haga clic en **Aceptar**.

16.4.12 Archivos usados recientemente, Salir

La lista de archivos recientes del menú **Archivo** muestra el nombre de archivo y la ruta de los últimos 9 archivos que se abrieron en StyleVision. Haga clic en una de las entradas de la lista para abrir el archivo en una pestaña nueva de la ventana principal.



También puede abrir los archivos con las teclas del teclado. P. ej.: pulse **Alt+F** para abrir el menú **Archivo** y después pulse el número del archivo que desea abrir. Por ejemplo, si pulsa la tecla **1** se abre el primer archivo de la lista. Si pulsa la tecla **2**, se abre el segundo archivo, y así sucesivamente.

Haga clic en el comando **Salir** para cerrar StyleVision. Si realizó cambios en alguno de los archivos abiertos pero no los guardó, aparece un aviso.

 © 2016-2022 Altova GmbH

16.5 Menú Edición

Los comandos del menú **Edición** sirven para editar documentos SPS y documentos en la vista Authentic. Además de los comandos de edición estándar, como **Cortar** (**Mayús+Supr** o **Ctrl+X**), **Copiar** (**Ctrl+C**), **Pegar** (**Ctrl+V**) y **Eliminar** (**Supr**), que no se describen en este apartado, el menú **Edición** ofrece estos comandos:

- **Deshacer, Rehacer, Seleccionar todo** 794: para deshacer o restaurar cambios realizados previamente y para seleccionar todo el contenido del diseño SPS.
- **Buscar, Buscar siguiente y Reemplazar** 794: para buscar texto en el diseño SPS, en la vista Authentic, y en la vista previa de las hojas de estilos XSLT y para reemplazar texto en la vista Authentic.
- **Parámetros de la hoja de estilos** 799: para editar parámetros declarados globalmente para el diseño SPS.
- **Contraer o expandir el marcado** 800: para contraer o expandir las etiquetas de marcado de los componentes de diseño.

Estos comandos también están disponibles en el menú contextual que aparece al hacer clic con el botón derecho en un componente o simplemente en la posición del cursor. Además, algunos comandos tienen teclas de acceso rápido e iconos en la barra de herramientas. Sin embargo, recuerde que los comandos que no correspondan a una vista determinada o a una posición determinada aparecen atenuados en el menú (es decir, se deshabilitan según corresponda).

16.5.1 Deshacer, Rehacer, Seleccionar todo

El comando **Deshacer** (**Ctrl+Z**) permite eliminar todos los cambios realizados y devolver el archivo a versiones anteriores. Todos los cambios se pueden deshacer uno por uno hasta llegar a la primera acción realizada desde que se abrió el documento.

El comando **Rehacer** (**Ctrl+Y**) permite rehacer las acciones que deshizo con el comando **Deshacer**. Esto significa que puede avanzar y retroceder en el historial de acciones con los comandos **Deshacer** y **Rehacer**.

El comando **Seleccionar todo** selecciona todo el contenido de la vista.

16.5.2 Buscar, Buscar siguiente, Reemplazar

El comando **Buscar** (**Ctrl+F**) sirve para buscar cadenas de texto en la vista **Diseño**, en el editor JavaScript, en la vista Authentic y en hojas de estilos XSLT.

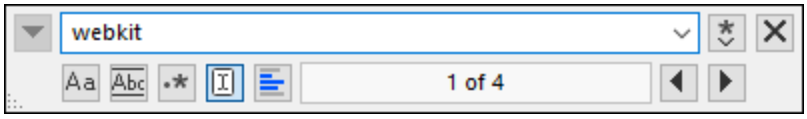
El cuadro de diálogo Buscar y reemplazar es distinto en función de la vista que esté activa.

- Si la vista activa es la de diseño, al hacer clic en los comandos **Buscar** o **Reemplazar** se activa la barra lateral Buscar y reemplazar 65. La funcionalidad de búsqueda y reemplazo de la vista de diseño se describe en el apartado Buscar y reemplazar 65.
- La funcionalidad de búsqueda y reemplazo en el editor de JavaScript, la vista Authentic, y las hojas de estilos XSLT se describen en este apartado.

© 2016-2022 Altova GmbH

Hojas de estilos XSLT y el editor JavaScript

Si hace clic en el comando **Buscar** desde el editor JavaScript o en la vista previa de una hoja de estilos XSLT para HTML y RTF, aparece este cuadro de diálogo:



En este cuadro de diálogo puede seleccionar varias opciones:

- *Coinc. mayús/min*: busca teniendo en cuenta las mayúsculas y minúsculas (por ejemplo, si marca esa casilla `Dirección` no se considera igual que `dirección`).
- *Solo palabras completas*: busca solamente las palabras que coincidan exactamente con el texto indicado. Por ejemplo, si busca `fit` la función de búsqueda no encontrará `fitness`, aunque la palabra contenga también `fit`.
- *Expresión regular*: si marca esta casilla el término de búsqueda se leerá como una expresión regular. Consulte el apartado Expresiones regulares 796 para aprender a usar expresiones regulares.
- *Buscar delimitador:* cuando se introduce un término de búsqueda, las coincidencias que se encuentran en el documento se resaltan y una de ellas se marca como la selección actual. Esta función indica si la primera coincidencia que se seleccione debe ser relativa a la posición del cursor. Si está activada, la primera coincidencia que se seleccione será la primera a partir del punto en que se encuentre el cursor del ratón. Si está desactivada, la primera coincidencia que se seleccione será la primera que se encuentre a partir del inicio del documento.
- *Buscar en selección:* si esta función está activada, la búsqueda de términos se lleva a cabo dentro del texto que esté seleccionado. Si no está activada, el término de búsqueda se busca en todo el documento. Para cambiar el texto seleccionado en el que quiere buscar el término primero debe deseleccionar primero la selección actual desactivando esta función.
- *Reemplazar*: en la pestaña Editor de JavaScript haga clic en la flecha hacia abajo que hay en la parte superior izquierda del cuadro de diálogo, con la que se abre el campo *Reemplazar*. Aquí puede introducir la cadena de texto con la que quiere sustituir la cadena encontrada.

Vista HTML y vista Authentic

Si hace clic en el comando **Buscar** desde la vista **Diseño** o **HTML** o desde la vista Authentic, se activa la barra lateral Buscar y reemplazar 65.

Debe tener en cuenta que:

- Si quiere que los resultados coincidan exactamente con el término de búsqueda, marque la casilla *Solo palabras completas*. P. ej., si busca `soft` no podrá encontrar `software`.
- Si quiere poder encontrar fragmentos de palabras no marque la casilla *Solo palabras completas*. De esta forma, si busca `soft` podrá encontrar `software`.
- Si no quiere que el resultado coincida con el término de búsqueda en el uso de mayúsculas y minúsculas, deje la casilla *Coinc. mayús/minús* sin marcar. P. ej. si busca `programa` encontrará tanto `programa` como `Programa`.
- En la vista Authentic la búsqueda se realiza en los datos dinámicos (en los datos XML) y no en el texto de entrada estático (XSLT).
- La funcionalidad **Reemplazar** en la vista Authentic permite reemplazar una cadena de texto con otra. Las opciones entre las que puede escoger son: (i) buscar solo palabras completas (no palabras que

contengan la cadena de búsqueda) y (ii) buscar palabras que coincidan como cadena de texto y en el uso de mayúsculas y minúsculas con la cadena de búsqueda.

Buscar siguiente

El comando **Buscar siguiente** (**F3**) busca la siguiente repetición del texto utilizado en la operación de búsqueda inmediatamente anterior. Para más información consulte la descripción del comando **Buscar** 794.

Cómo usar expresiones regulares

Puede usar expresiones regulares (regex) para buscar cadenas de texto. Para ello primero debe activar la función Expresión regular (*véase más arriba*). Esto indica que el texto que se introduce en el campo de búsqueda se evalúa como una expresión regular. Después introduzca la expresión regular en el campo de búsqueda. Si necesita ayuda con la expresión regular puede usar el botón **Generador de expresiones regulares** que encontrará a la derecha del campo de búsqueda. Haga clic en un elemento del Generador para introducir los metacaracteres regex correspondientes en el campo de búsqueda. Más abajo se incluye una breve descripción de cada uno de esos metacaracteres.

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 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).

© 2016-2022 Altova GmbH

$	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.

   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

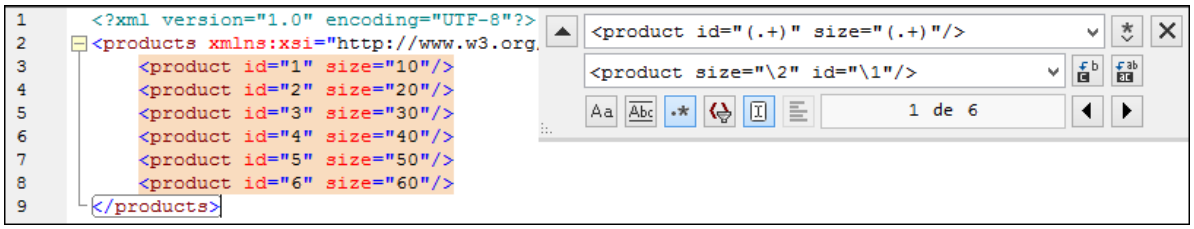
El fragmento de código XML que aparece a continuación incluye una lista de producto. Cada producto tiene dos atributos llamados `id` y `size`. Imaginemos que nuestro objetivo es cambiar el orden de los atributos `id` y `size` de cada elemento `product` (es decir, en todos los elementos `product` el atributo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



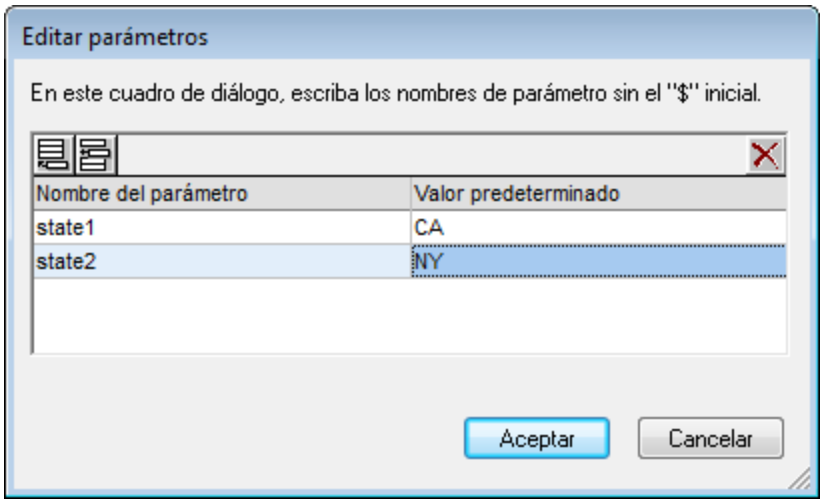
5. Haga clic en **Reemplazar todos** y observe el resultado. Todos los elementos `product` del documento XML se reemplazaron de modo que el atributo `size` aparezca delante del atributo `id`.

Temas relacionados

- Barras de herramientas 748

16.5.3 Parámetros de la hoja de estilos

El comando **Parámetros de la hoja de estilos** sirve para declarar y editar parámetros con sus valores predeterminados. Este comando está habilitado en la vista **Diseño** y en la vista Authentic. Al hacer clic en este comando se abre el cuadro de diálogo "Editar parámetros".



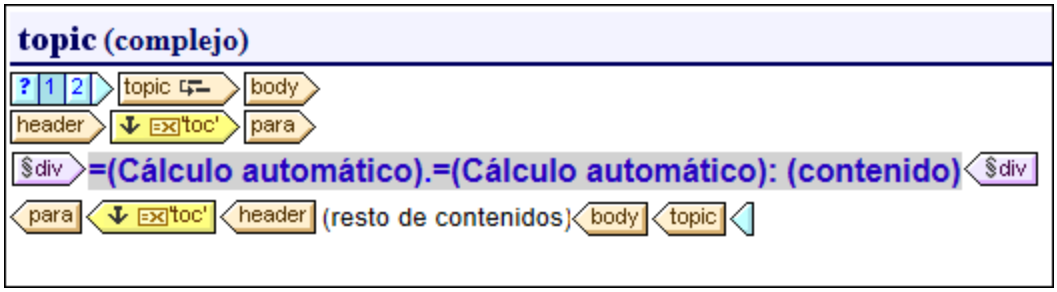
Debe tener en cuenta que:

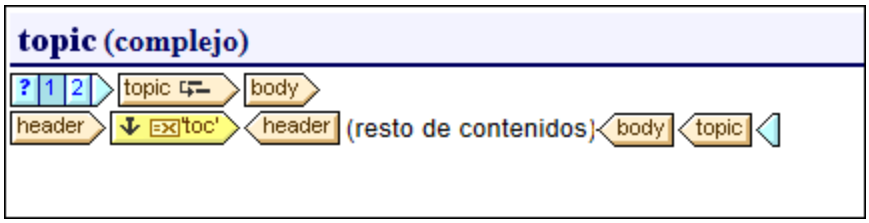
- Desde este cuadro de diálogo puede insertar, anexar, editar y eliminar parámetros para toda la hoja de estilos y para los filtros de BD.
- Los nombres de parámetro deben empezar con una letra y pueden incluir los caracteres `A - Z`, `a - z`, `0 - 9` y el signo de subrayado (_).
- Este cuadro de diálogo contiene todos los parámetros definidos por el usuario en el documento SPS.
- También puede declarar parámetros en la ventana Vista general del diseño 45.

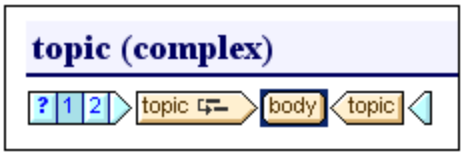
16.5.4 Contraer o expandir el marcado

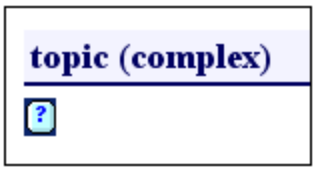
El comando **Contraer o expandir el marcado** es un comando conmutador que contrae/expande la etiqueta de marcado seleccionada. Este comando puede utilizarse con cualquier tipo de etiqueta, ya sea un nodo, formato predefinido, mecanismo SPS, etc. Para contraer/expandir una etiqueta, haga doble clic en ella. Esta operación funciona tanto con la etiqueta de inicio como con la etiqueta final.

Las imágenes que aparecen a continuación muestran ejemplos de diseño con etiquetas contraídas. Haga doble clic en una etiqueta contraída para expandirla.



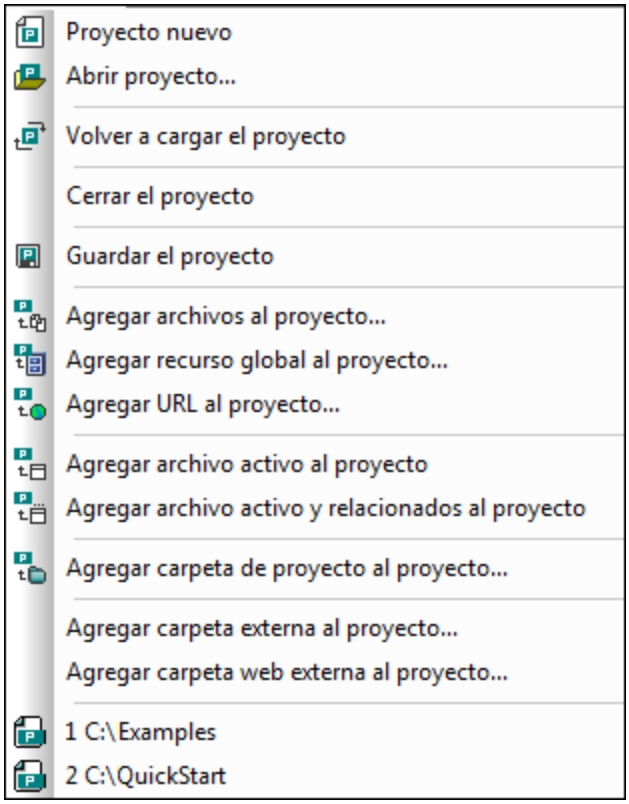

topic (complejo)
?
1
2
topic
body
header
'toc'
header
(resto de contenidos)
body
topic


topic (complex)
?
1
2
topic
body
topic


topic (complex)
?

16.6 Menú Proyecto

El menú **Proyecto** (*imagen siguiente*) sirve para crear y modificar proyectos y darles una estructura. Gracias a este menú podrá crear un proyecto, definir cuáles son sus archivos y ordenar estos archivos por carpetas según el tipo de archivo. Desde la ventana Proyecto puede acceder a los diferentes archivos.



El menú **Proyecto** ofrece todos estos comandos (*imagen anterior*):

- **Proyecto nuevo** 803 para crear un proyecto nuevo
- **Abrir proyecto** 803 para abrir un proyecto ya existente
- **Volver a cargar el proyecto** 803 para actualizar un proyecto en la ventana Proyecto
- **Cerrar el proyecto** 803 para cerrar un proyecto en la ventana Proyecto
- **Guardar el proyecto** 803 para guardar y dar nombre a un proyecto
- **Agregar archivos al proyecto** 804 para añadir archivos a un proyecto en la ventana Proyecto
- **Agregar recurso global al proyecto** 804 para añadir recursos globales de Altova a un proyecto en la ventana Proyecto
- **Agregar URL al proyecto** 804 para añadir un archivo a un proyecto a través de una dirección URL en la ventana Proyecto
- **Agregar archivo activo al proyecto** 806 para añadir el archivo activo a un proyecto
- **Agregar archivo activo y relacionados al proyecto** 806 para añadir el archivo activo y sus archivos relacionados a un proyecto
- **Agregar carpeta de proyecto al proyecto** 806 para añadir a carpetas a un proyecto en la ventana Proyecto
- **Agregar carpeta externa al proyecto** 806 para añadir carpetas locales a un proyecto en la ventana Proyecto

© 2016-2022 Altova GmbH

- **Agregar carpeta web externa al proyecto** 806 para añadir carpetas a un proyecto a través de una URL en la ventana Proyecto

Proyectos recientes

Al final del menú **Proyecto** aparecen los últimos 9 proyectos más recientes, para que pueda abrirlos rápidamente.

Arrastrar y colocar

En la ventana Proyecto puede arrastrar carpetas dentro de otras carpetas o a otra posición dentro de la misma carpeta. Los archivos se pueden arrastrar a otras carpetas pero no a otra posición dentro de la misma carpeta (porque los archivos están ordenados alfabéticamente en cada carpeta). Además puede arrastrar archivos y carpetas desde el explorador de Windows hasta la ventana Proyecto.

16.6.1 Proyecto nuevo, Abrir proyecto, Volver a cargar el proyecto

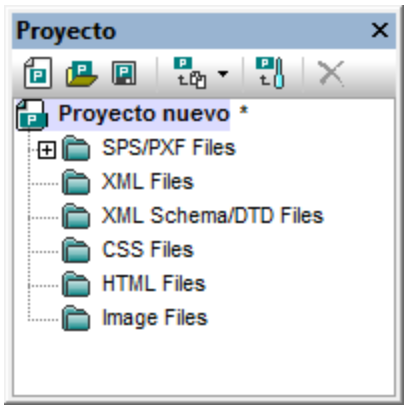
El comando **Proyecto nuevo** crea un proyecto nuevo. El nuevo proyecto reemplaza el proyecto anterior (si lo había) en la ventana Proyecto. Si el proyecto en el que estaba trabajando tiene cambios sin guardar, aparece un aviso preguntando si quiere guardar los cambios. Recuerde que el comando **Proyecto nuevo** crea un proyecto nuevo pero no lo guarda. Para guardarlo haga clic en el comando **Guardar el proyecto** 803.

El comando **Abrir proyecto** abre un proyecto ya existente, que aparece en la ventana Proyecto. El proyecto que estaba abierto en la ventana Proyecto se reemplaza con este otro proyecto. Si el proyecto que está abierto tiene cambios sin guardar, aparece un aviso preguntando si desea guardar los cambios.

El comando **Volver a cargar el proyecto** vuelve a cargar el proyecto actual. Este comando es ideal si está trabajando en un entorno con varios usuarios y los demás usuarios hacen cambios en el proyecto.

16.6.2 Cerrar el proyecto, Guardar el proyecto

El comando **Cerrar el proyecto** cierra el proyecto activo. Si el proyecto tiene cambios sin guardar, aparece un aviso para que guarde el proyecto antes de cerrarlo. En la ventana Proyecto los cambios sin guardar se indican por medio de un asterisco situado junto al nombre del proyecto (*imagen siguiente*).



El comando **Guardar el proyecto** guarda el proyecto actual. Recuerde que al proyecto se le pone un nombre cuando se guarda por primera vez. Una vez guardado solamente podrá cambiar el nombre del proyecto desde fuera de StyleVision, p. ej. desde el explorador de archivos de Windows.

16.6.3 Agregar archivos / recurso global / URL al proyecto

Agregar archivos al proyecto

Este comando añade archivos al proyecto actual. Al hacer clic en este comando se abre el cuadro de diálogo "Abrir", donde puede seleccionar los archivos que desea añadir al proyecto. Los archivos seleccionados se añaden a las subcarpetas del proyecto, dependiendo de qué extensiones de archivo se definieran para cada subcarpeta. Si definió la misma extensión de archivo para más de una carpeta, el archivo que tenga esa extensión se añade a la primera subcarpeta que tenga esa extensión.

Para añadir un archivo a una carpeta o subcarpeta de la carpeta de proyecto, haga clic con el botón derecho en la carpeta, seleccione el comando **Agregar archivos** y busque los archivos que desea agregar.

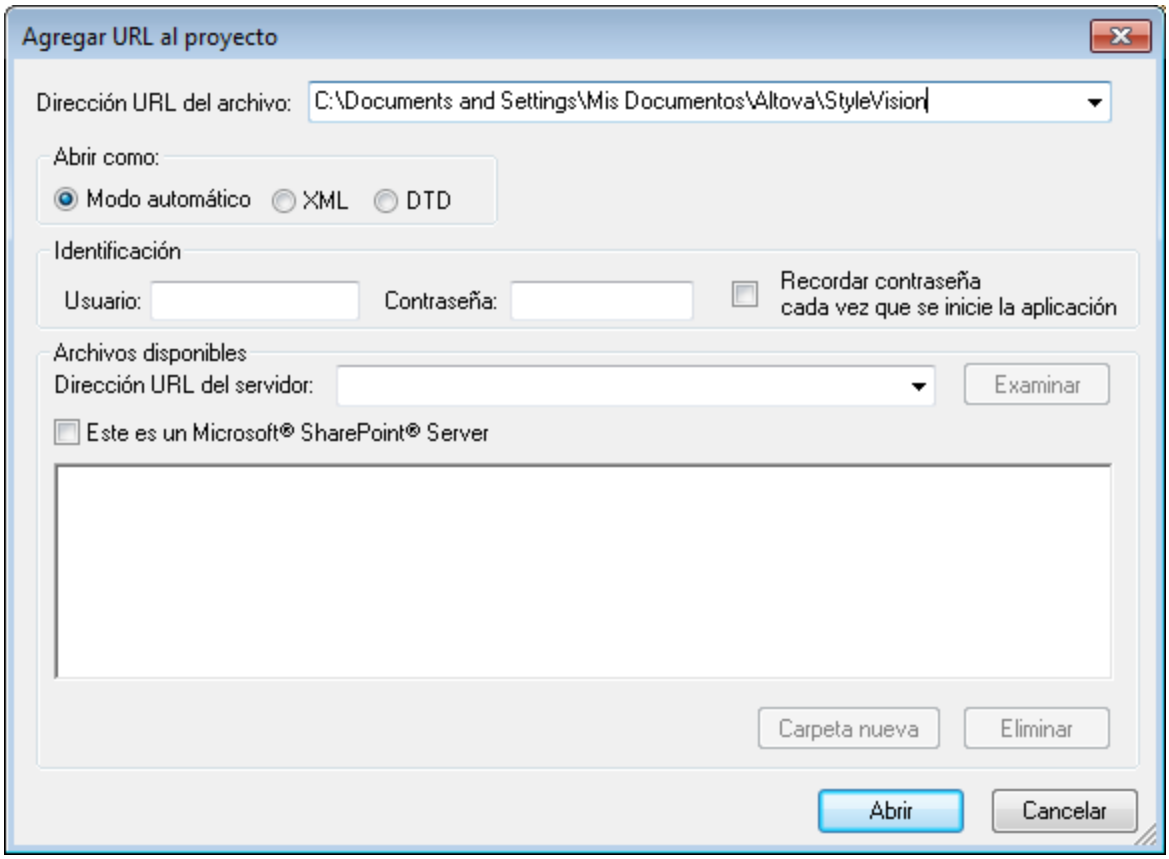
Agregar recurso global al proyecto

Este comando abre un cuadro de diálogo que enumera los recursos globales que están activos en el archivo XML de recursos globales. En el cuadro de diálogo puede seleccionar uno de estos recursos para agregarlo al proyecto activo. Seleccione el recurso global y haga clic en **Aceptar**.

Agregar URL al proyecto

Este comando añade al proyecto actual una URL, que a su vez añade el objeto de destino de la URL al proyecto. Al hacer clic en este comando se abre el cuadro de diálogo "Agregar URL al proyecto" (*imagen siguiente*).

 © 2016-2022 Altova GmbH



En este cuadro de diálogo puede insertar la URL de un archivo (con o sin el protocolo `file:\\`) o de un servidor. Sin inserta la URL de un servidor, escriba el nombre de usuario y la contraseña y después la URL del servidor. Haga clic en **Examinar** para conectarse al servidor y después en la lista de *Archivos disponibles*, seleccione el archivo que desea añadir a la carpeta de proyecto.

No olvide que también puede añadir direcciones URL a carpetas y subcarpetas de la carpeta de proyecto principal. Para ello haga clic con el botón derecho en la carpeta de proyecto y seleccione el comando **Agregar URL**. Esto abre el cuadro de diálogo "Agregar URL". Siga las instrucciones que acabamos de dar para usar este cuadro de diálogo.

Arrastrar y colocar

Puede arrastrar un archivo de una carpeta a otra pero no a una posición nueva dentro de la misma carpeta. Esto se debe a que los archivos están ordenados alfabéticamente dentro de cada carpeta. Además puede arrastrar archivos desde el explorador de archivos de Windows hasta la ventana Proyecto.

Eliminar archivos, recursos y direcciones URL

Si quiere eliminar un archivo, un recurso global o una dirección URL, haga clic con el botón derecho en el objeto en la ventana Proyecto y seleccione **Eliminar** en el menú contextual que aparece.

16.6.4 Agregar archivo activo y relacionados al proyecto

Agregar archivo activo al proyecto

Este comando añade el archivo SPS activo al proyecto actual. El archivo se añade a la primera carpeta que tenga definida la extensión de archivo `.sps`. Si además del archivo SPS activo quiere añadir el esquema, el archivo XML de trabajo, la plantilla XML y los archivos CSS y de imagen relacionados con el SPS, use el comando **Agregar archivo activo y relacionados al proyecto** (ver más abajo). Para agregar el archivo activo a una carpeta o subcarpeta de la carpeta de proyecto principal, haga clic con el botón derecho en la carpeta de destino y seleccione el comando **Agregar archivo activo**.

Agregar archivo activo y relacionados al proyecto

Este comando añade al proyecto el archivo SPS activo y el archivo de esquema, el archivo XML de trabajo, la plantilla XML y los archivos CSS y de imagen relacionados con él. Cada archivo se añade a la primera carpeta que tenga definida la pertinente extensión de archivo. Si quiere agregar el archivo activo y sus archivos relacionados a una carpeta o subcarpeta de la carpeta de proyecto principal, haga clic con el botón derecho en la carpeta de destino y seleccione el comando **Agregar archivo activo y relacionados**.

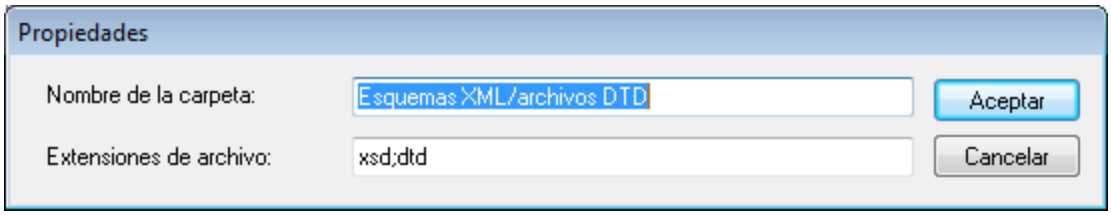
Eliminar archivos

Si quiere eliminar un archivo, haga clic en el archivo con el botón derecho (en la ventana Proyecto) y seleccione Eliminar.

16.6.5 Agregar carpeta y carpetas externas al proyecto

Agregar carpeta de proyecto al proyecto

Este comando añade una carpeta nueva al proyecto actual. Al hacer clic en el comando se abre el cuadro de diálogo "Propiedades" (*imagen siguiente*), donde puede escribir el nombre de la carpeta y las extensiones de archivo asociadas a la carpeta (si quiere indicar más de una extensión, sepárelas con punto y coma). Cuando se añade al proyecto un archivo con esa extensión, el archivo se añade automáticamente a esa carpeta. La carpeta nueva se anexa a la lista de carpetas del proyecto en la ventana Proyecto.



Para crear una subcarpeta en una carpeta del proyecto, haga clic con el botón derecho en la carpeta en la ventana Proyecto y seleccione **Agregar carpeta de proyecto** en el menú contextual. En el diálogo "Propiedades" que aparece, escriba el nombre de la carpeta y las extensiones de archivo que desea asignarle.

Agregar carpeta externa al proyecto

Este comando añade al proyecto actual una carpeta externa, local o de red, con todos sus contenidos. La carpeta externa agregada al proyecto se puede expandir y contraer. Para agregar una carpeta externa a una

 © 2016-2022 Altova GmbH

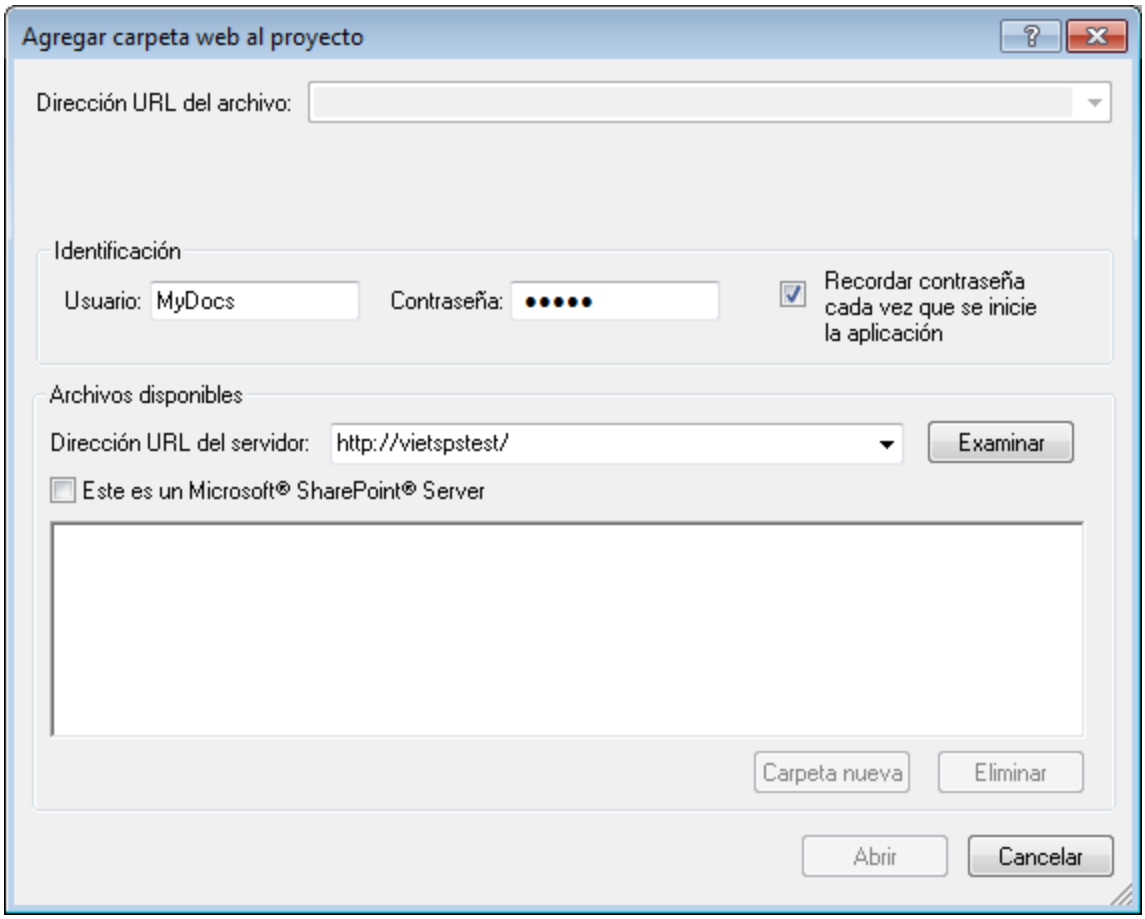
carpeta de proyecto determinada, haga clic con el botón derecho en la carpeta en la ventana Proyecto y seleccione el comando **Agregar carpeta externa** en el menú contextual.

Agregar carpeta web externa al proyecto

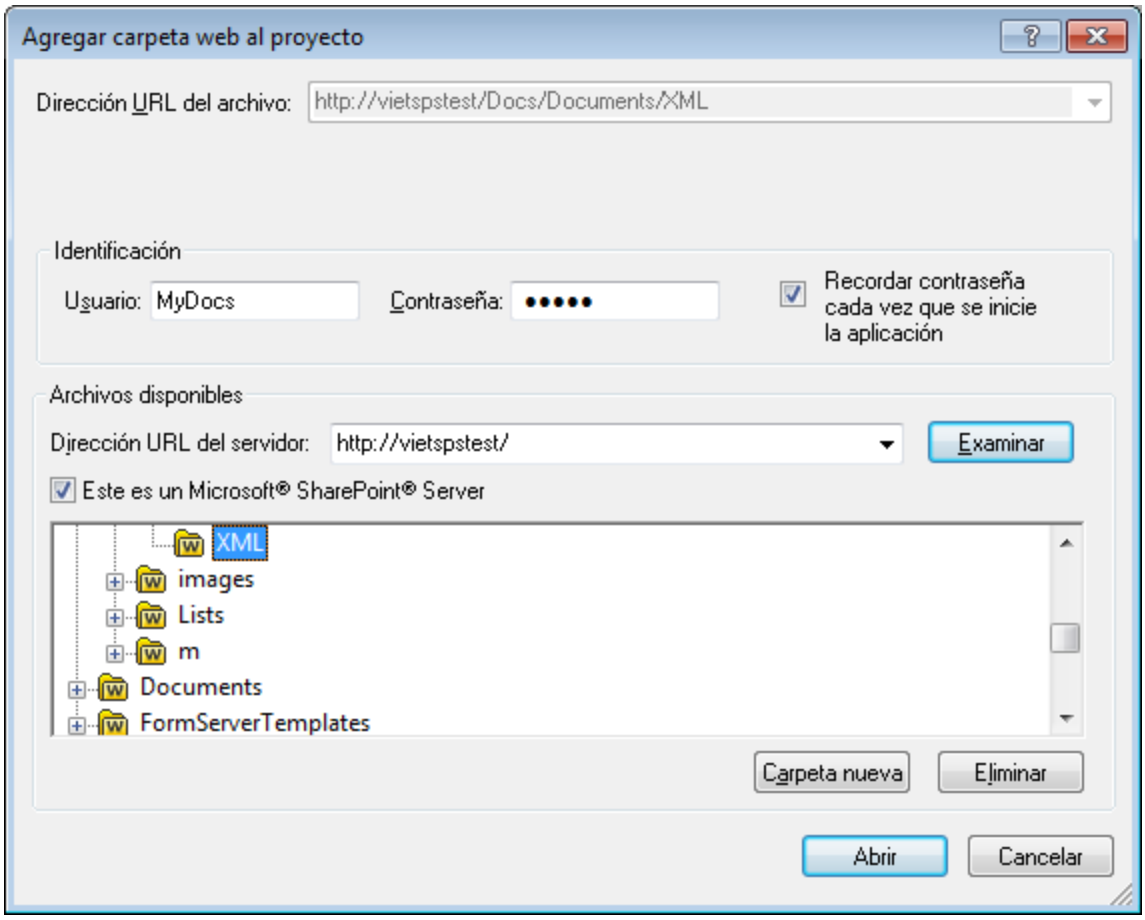
Este comando añade al proyecto actual una carpeta externa a través de una URL. La carpeta externa agregada al proyecto se puede expandir y contraer. Para agregar una carpeta externa a una carpeta de proyecto determinada, haga clic con el botón derecho en la carpeta en la ventana Proyecto y seleccione el comando **Agregar carpeta web externa** en el menú contextual.

Al hacer clic en este comando se abre el cuadro de diálogo "Agregar carpeta web" (*imagen siguiente*).

1. Escriba la dirección URL del servidor en el campo *Dirección URL del servidor* y el nombre de usuario y la contraseña en el grupo de opciones *Identificación*.



2. Haga clic en el botón **Examinar** para examinar el servidor y ver las carpetas disponibles.



3. Haga clic en la carpeta que desea añadir a la ventana Proyecto (en este momento se habilita el botón **Abrir**). El nombre de la carpeta elegida y al URL del servidor aparecen ahora en el campo *Dirección URL del archivo*.
4. Haga clic en **Aceptar** para agregar la carpeta al proyecto.
5. Haga clic en el icono **+** para ver el contenido de la carpeta.
6. Para asignar extensiones de archivo a la carpeta web, haga clic con el botón derecho en la carpeta y seleccione **Propiedades** en el menú contextual. Ahora inserte las extensiones de archivo en el cuadro de diálogo "Propiedades".

Arrastrar y colocar

En la ventana Proyecto puede arrastrar una carpeta dentro de otra carpeta o a otra posición dentro de la misma carpeta. Además puede arrastrar carpetas desde el explorador de archivos de Windows hasta la ventana Proyecto.

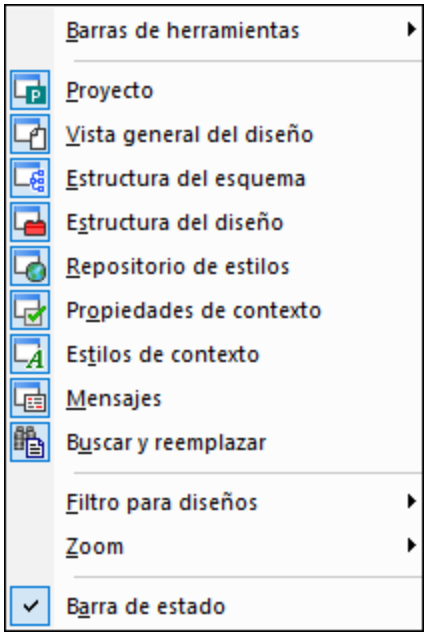
Eliminar carpetas

Si quiere eliminar una carpeta o varias carpetas, selecciónelas en la ventana Proyecto, haga clic con el botón derecho y seleccione **Eliminar** en el menú contextual.

© 2016-2022 Altova GmbH

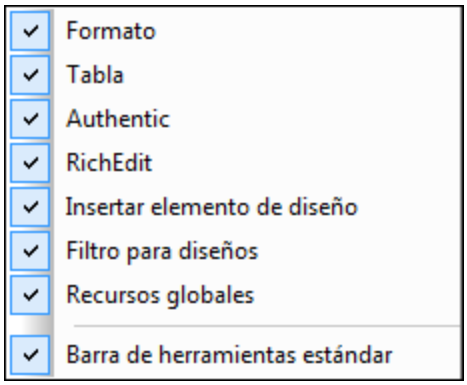
16.7 Menú Vista

Los comandos del menú **Vista** (*imagen siguiente*) sirven para cambiar el aspecto de la interfaz gráfica del usuario y mostrar/ocultar sus componentes. Con este menú podrá elegir qué barras de herramientas y qué ventanas laterales se muestran/ocultan en la interfaz gráfica. Además puede seleccionar filtros para el diseño y activar/desactivar la barra de estado.



16.7.1 Barras de herramientas y barra de estado

Al pasar el cursor por el comando **Barras de herramientas** aparece un submenú (*imagen siguiente*) donde puede activar/desactivar cada una de las barras de herramientas de StyleVision.



Si marca una barra de herramientas, la barra de herramientas se muestra en la interfaz gráfica. Por ejemplo, en el ejemplo de la imagen anterior seleccionamos todas las barras de herramientas y, por tanto, todas se

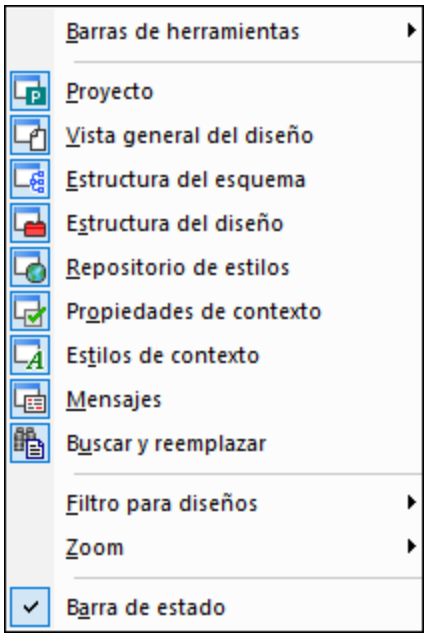
mostrarán en la interfaz gráfica. Para más información sobre las barras de herramientas consulte el apartado Referencia del usuario | Barras de herramientas 748.

Barra de estado

La barra de estado está situada en la parte inferior de la ventana de la aplicación y puede ocultarse/mostrarse haciendo clic en el comando conmutador **Barra de estado**.

16.7.2 Barras laterales de diseño

Los comandos del menú **Vista** son conmutadores que sirven para mostrar/ocultar las ventanas laterales de la interfaz gráfica del usuario (*imagen siguiente*).



Si una ventana lateral está marcada (activada), el icono que aparece junto al nombre de la ventana en el menú **Vista** tiene un contorno azul. Haga clic en una ventana lateral del menú **Vista** para activarla/desactivarla (mostrarla/ocultarla). Las opciones elegidas en este menú son específicas de cada vista. Es decir, las opciones elegidas para una vista determinada (vista **Diseño**, vista Authentic, vista de resultados, etc.) se conserva en esa vista concreta hasta que se elijan otras opciones.

16.7.3 Filtro para diseños, Zoom

Filtro para diseños

El comando **Filtro para diseños** del menú **Vista** abre un submenú con comandos que sirven para filtrar qué plantillas aparecen en la vista **Diseño**. Estos comandos son muy prácticos si el diseño con el que trabaja es muy extenso o contiene varias plantillas. Gracias al filtro para diseños podrá indicar qué tipo de plantillas aparecen en la vista **Diseño** y cuáles se ocultan. Estas son las opciones del filtro para diseños:

 © 2016-2022 Altova GmbH

Icono	Comando	Descripción
	Mostrar solo una plantilla	Solo muestra la plantilla seleccionada. Ponga el cursor en una plantilla y haga clic en este icono para ver solo la plantilla seleccionada.
	Mostrar todos los tipos de plantilla	Muestra todas las plantillas del SPS (plantilla principal, global, con nombre y de diseño).
	Mostrar plantillas importadas	Haga clic para mostrar/ocultar las plantillas importadas.
	Mostrar/ocultar la plantilla principal	Haga clic para mostrar/ocultar la plantilla principal.
	Mostrar/ocultar las plantillas globales	Haga clic para mostrar/ocultar las plantillas globales.
	Mostrar/ocultar los fragmentos de diseño	Haga clic para mostrar/ocultar los fragmentos de diseño.

Recuerde que estos comandos también están disponibles en forma de iconos en la barra de herramientas Filtro para diseños 758.

Zoom

El comando **Zoom** del menú **Vista** abre un submenú donde puede elegir un nivel de zoom para la vista. También puede cambiar el nivel de zoom en el cuadro combinado *Zoom* de la barra de herramientas Estándar. Otra manera de cambiar el nivel de zoom es pulsar la tecla **Ctrl** y mover la rueda de desplazamiento del ratón.

16.8 Menú Insertar

El menú **Insertar** incluye comandos con los que podrá insertar diferentes componentes de diseño en el SPS. Algunos de ellos también están disponibles como iconos en barras de herramientas 748. Además, los comandos del menú **Insertar** están disponibles en los **menús contextuales** que aparecen al hacer clic con el botón derecho dentro del SPS en la vista **Diseño**. Los comandos que no correspondan a la posición del cursor se deshabilitan.

Nota: los comandos del menú **Insertar** solamente están disponibles en la vista **Diseño** porque sirven para construir el diseño SPS.

16.8.1 Contenido

El comando **Insertar contenido** inserta un marcador de posición `(contenido)` en la posición del cursor. El marcador se puede insertar dentro de dos tipos de nodos (**elementos** y **atributos**) e indica que se procesarán todos los secundarios del nodo actual.

- Si el nodo actual es un nodo de **elemento**, se procesan sus todos sus nodos de elemento y texto secundarios. Para procesar los nodos de elemento secundarios se usarán plantillas globales, si existen. De lo contrario, se usará la regla de plantilla integrada para elementos. Para procesar los nodos de texto secundarios se usará la regla de plantilla integrada para nodos de texto, cuyo efecto es dar salida el texto. En la práctica, la regla de plantilla integrada para elementos da salida al texto de todos los nodos de texto descendientes. No olvide que, cuando se usa el marcador de posición `(contenido)`, no se da salida a los valores de los atributos a no ser que se defina una plantilla global para el elemento primario del atributo o para uno de sus antecesores y a no ser que se de salida al atributo de forma explícita, con ayuda del marcador de posición `(contenido)` o de cualquier otro componente de presentación de contenidos.

- Si el nodo actual es un nodo de **atributo**, se procesa su nodo de texto secundario con la regla de plantilla integrada. Esta plantilla copia el texto del nodo de texto en el documento de salida. Es decir, da salida al valor del atributo .

También puede insertar el marcador de posición `(contenido)` dentro de un nodo haciendo clic con el botón derecho entre las etiquetas del nodo y seleccionando **Insertar | Contenido** en el menú contextual. Otra opción es hacer clic en el icono **Insertar contenido** de la barra de herramientas Insertar elementos de diseño 755 y después en la posición del diseño donde desea insertar el elemento.

Dar estilo al contenido

Para aplicar formato al marcador de posición `(contenido)` selecciónelo y use un formato predefinido y las propiedades de la ventana Estilos. Este formato se puede apreciar en el diseño y en el documento de salida se aplicará al contenido del nodo.

Reemplazar contenido

Si suelta un nodo nuevo del esquema dentro de un nodo que contiene un marcador de posición `(contenido)`, el marcador `(contenido)` actual se sustituye con el nodo nuevo.

© 2016-2022 Altova GmbH

Eliminar contenido

Para eliminar el marcador de posición `(contenido)` selecciónelo y pulse la tecla **Suprimir** del teclado.

Nota: puede crear una regla de plantilla vacía si elimina el marcador de posición `(contenido)` de un nodo. Las reglas de plantilla vacías son ideales si no quiere aplicar ninguna plantilla a un nodo determinado (es decir, que el nodo no genere resultados).

16.8.2 Resto de contenidos

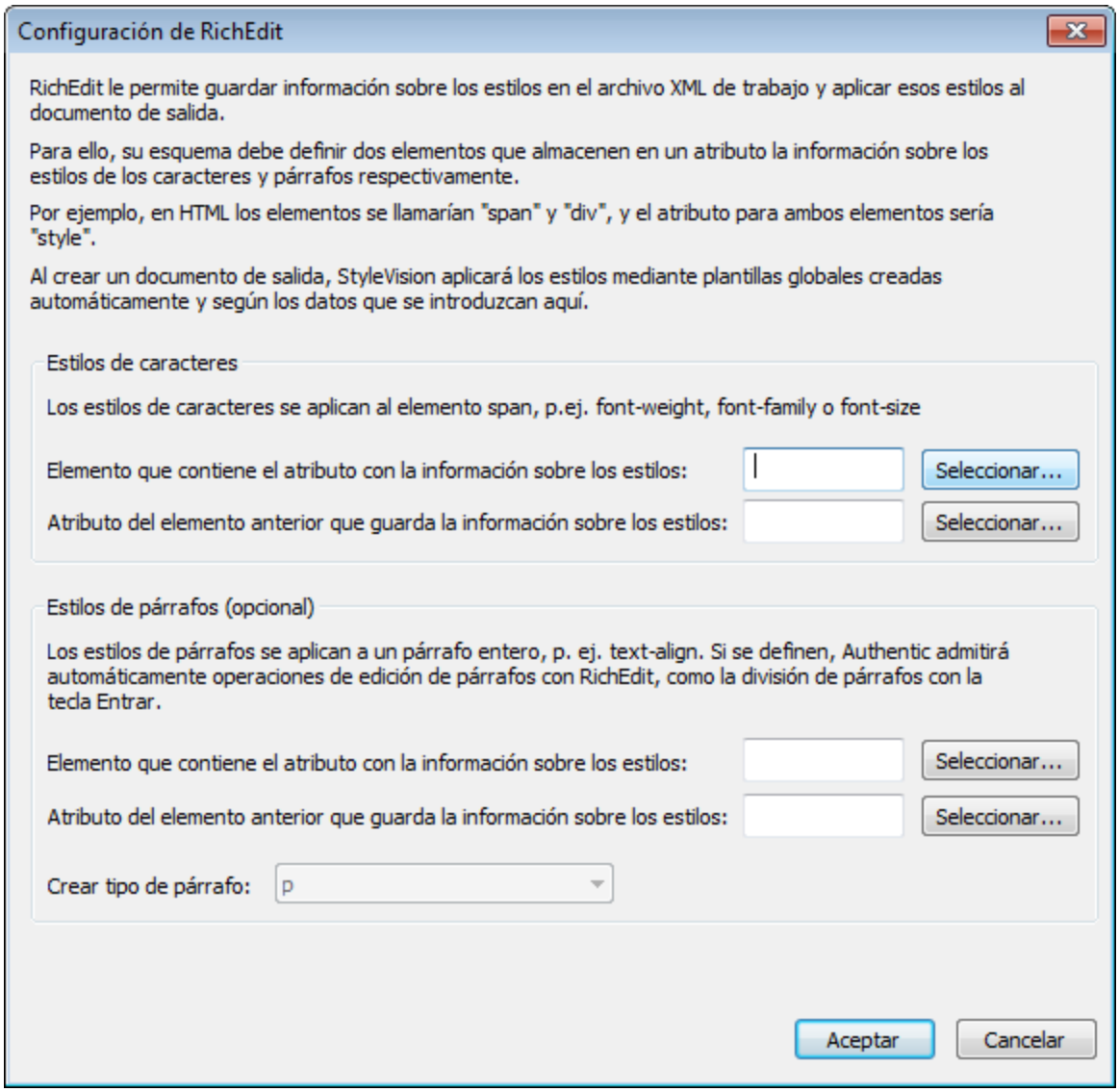
El comando **Insertar resto de contenidos** inserta el marcador de posición `(resto de contenidos)` para el nodo. Este marcador de posición representa el contenido de los nodos secundarios no utilizados del nodo actual y equivale a aplicar la regla `xsl:apply-templates` de XSLT a los nodos de elemento y de texto no utilizados del elemento actual. Recuerde que las plantillas no se aplican a los atributos secundarios. El marcador de posición `(resto de contenidos)` también se puede insertar poniendo el cursor entre las etiquetas de un elemento, haciendo clic con el botón derecho y seleccionando **Insertar resto de contenidos** en el menú contextual.

Recomendamos usar el marcador de posición `(resto de contenidos)` cuando quiera procesar un elemento secundario de una manera especial y aplicar plantillas a los elementos del mismo nivel. Es importante aplicar plantillas a los elementos del mismo nivel para garantizar que estos se procesan. Así podrá llegar hasta los elementos más inferiores de la jerarquía del documento.

Para eliminar el marcador de posición `(resto de contenidos)` selecciónelo y pulse la tecla **Suprimir** del teclado.

16.8.3 RichEdit

El comando **Insertar RichEdit** inserta un componente RichEdit en la posición del cursor. Cuando se introduce un componente RichEdit por primera vez, aparece el cuadro de diálogo "Configuración de RichEdit" (*imagen siguiente*). Esta configuración de RichEdit se aplica a todos los componentes RichEdit del documento de diseño. Por eso el cuadro de diálogo solo aparece una vez.



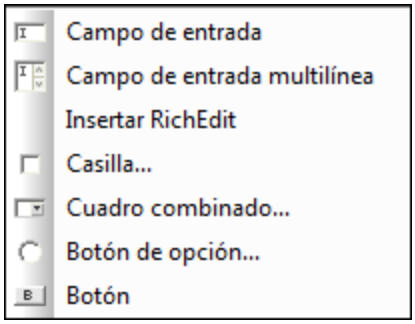
En este cuadro de diálogo debe indicar qué elemento de estilo y cuál de sus atributos debe contener las propiedades de estilo RichEdit. También puede seleccionar el elemento y atributo en la estructura del esquema. Para ello haga clic en el botón **Seleccionar...** correspondiente. Cuando termine de configurar los componentes RichEdit haga clic en **Aceptar**. A continuación se crea el componente RichEdit (*imagen siguiente*) y en el diseño se crea una plantilla global RichEdit no editable llamada igual que el elemento de estilo (p. ej. en la imagen siguiente se llama `Style`).

 © 2016-2022 Altova GmbH

Para más información sobre la característica RichEdit consulte la sección Flexibilidad de estilos en Authentic [408] del presente manual.

16.8.4 Controles de formulario

Al pasar el puntero del ratón sobre el comando **Insertar controles de formulario** aparece un submenú (*imagen siguiente*) con varios comandos. Estos comandos sirven para insertar controles de formulario (también llamados *dispositivos de entrada de datos* [190]).



La sección Controles de formulario [190] del manual explica cómo crear estos controles de formulario uno a uno. Después de crear el control de formulario puede editar sus propiedades. Para ello seleccione el control de formulario y edite la propiedad pertinente en la ventana Propiedades [59].

También puede insertarlos haciendo clic con el botón derecho en el punto de inserción y seleccionando **Insertar | Controles de formulario** en el menú contextual. Otra opción es hacer clic en el icono correspondiente de la barra de herramientas Insertar elementos de diseño [755] y después en la posición del diseño donde desea crear el control de formulario.

Nota: en los campos de entrada (es decir, en los cuadros de texto) no se pueden insertar secciones CDATA. Las secciones CDATA solamente se pueden insertar dentro de elementos que se representan en la vista Authentic como componentes de contenido de texto.

16.8.5 Control de base de datos

Al pasar el puntero del ratón sobre el comando **Insertar control de base de datos** se abre un submenú con varios comandos. Estos comandos sirven para insertar controles en la vista Authentic que ayudarán al usuario a navegar por los registros de la BD en la vista Authentic y a consultar la BD. Estos controles se pueden insertar en el diseño y aparecen en el documento de la vista Authentic en la posición correspondiente.

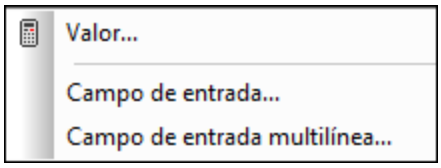
Los comandos para insertar controles de base de datos son:

- **Navegación**
- **Navegación + Ir a**
- **Botón de consulta**

Para más información sobre estos controles y cómo utilizarlos puede consultar la sección Características de diseño para BD 628.

16.8.6 Cálculo automático

Los **cálculos automáticos** utilizan expresiones XPath para calcular un valor. En el documento de salida el valor aparece en la posición donde se insertó el cálculo automático. En el diseño SPS puede insertar cálculos automáticos como valor de texto, como campo de entrada o como campo de entrada multilínea. Para ello ponga el cursor en la posición donde se debe insertar el cálculo automático. Después seleccione el comando de menú **Insertar | Insertar cálculo automático** y elija en qué forma se inserta el cálculo automático: **Valor**, **Campo de entrada** o **Campo de entrada multilínea** (*imagen siguiente*). En lugar de usar los comandos del menú **Insertar** puede hacer clic con el botón derecho y usar el menú contextual que aparece. Otra opción es hacer clic en el icono **Cálculo automático** de la barra de herramientas Insertar elementos de diseño 755 y después en el diseño.

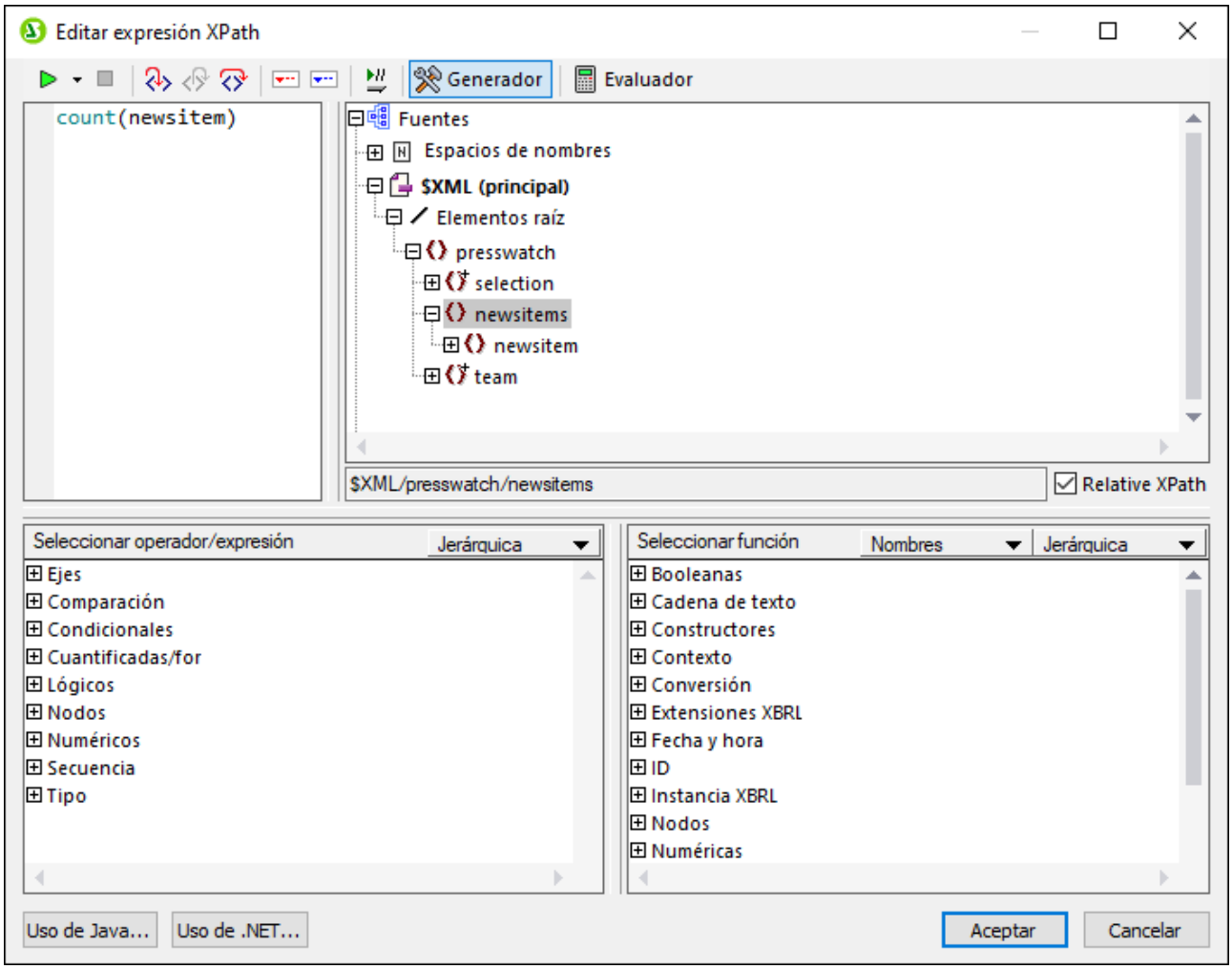


El valor del cálculo automático se representará según corresponda en la vista Authentic y en el documento de salida.

Expresión XPath para el cálculo automático

Tras elegir el formato de presentación para el cálculo automático (**Valor**, **Campo de entrada** o **Campo de entrada multilínea**) aparece el cuadro de diálogo Editar expresión XPath 729 (*imagen siguiente*).

 © 2016-2022 Altova GmbH



El nodo de contexto para la expresión aparece resaltado en el panel inferior izquierdo, que muestra la estructura del esquema. Puede escribir la expresión XPath directamente en el cuadro de texto *Expresión XPath* o puede hacer doble clic en los elementos de los paneles inferiores para insertarlos en el cuadro de texto. Los nodos de la estructura del esquema del panel izquierdo se insertan en relación al nodo de contexto (si marca la casilla *XPath relativa*) o como expresión absoluta empezando por el nodo de documento (si no marca la casilla *XPath relativa*).

Cuando termine de construir la expresión XPath, haga clic en **Aceptar** para insertar el cálculo automático en el diseño.

Actualizar un nodo XML con un cálculo automático

Los nodos del documento XML se pueden actualizar con el resultado de un cálculo automático. Esto se explica en la sección Actualizar nodos con cálculos automáticos 296.

16.8.7 Selector de fecha

El comando **Insertar selector de fecha** inserta un selector de fecha en la posición actual del cursor. Este comando solamente se habilita si el cursor está dentro de un nodo `xs:date` o `xs:dateTime` y si el elemento se usó para crear `(contenido)` o un campo de entrada.

16.8.8 Párrafo, párrafo especial

El comando **Párrafo** inserta un elemento de párrafo HTML `<p>` alrededor del componente seleccionado. Se considera que el componente está seleccionado si todo el nodo está seleccionado (tras hacer clic en una de sus etiquetas) o si hay texto estático seleccionado. Si pone el cursor dentro de texto estático, el elemento de párrafo se inserta en esa posición (su etiqueta inicial y final). También puede insertar un párrafo haciendo clic en el icono **Insertar párrafo** de la barra de herramientas Insertar elementos de diseño 755.

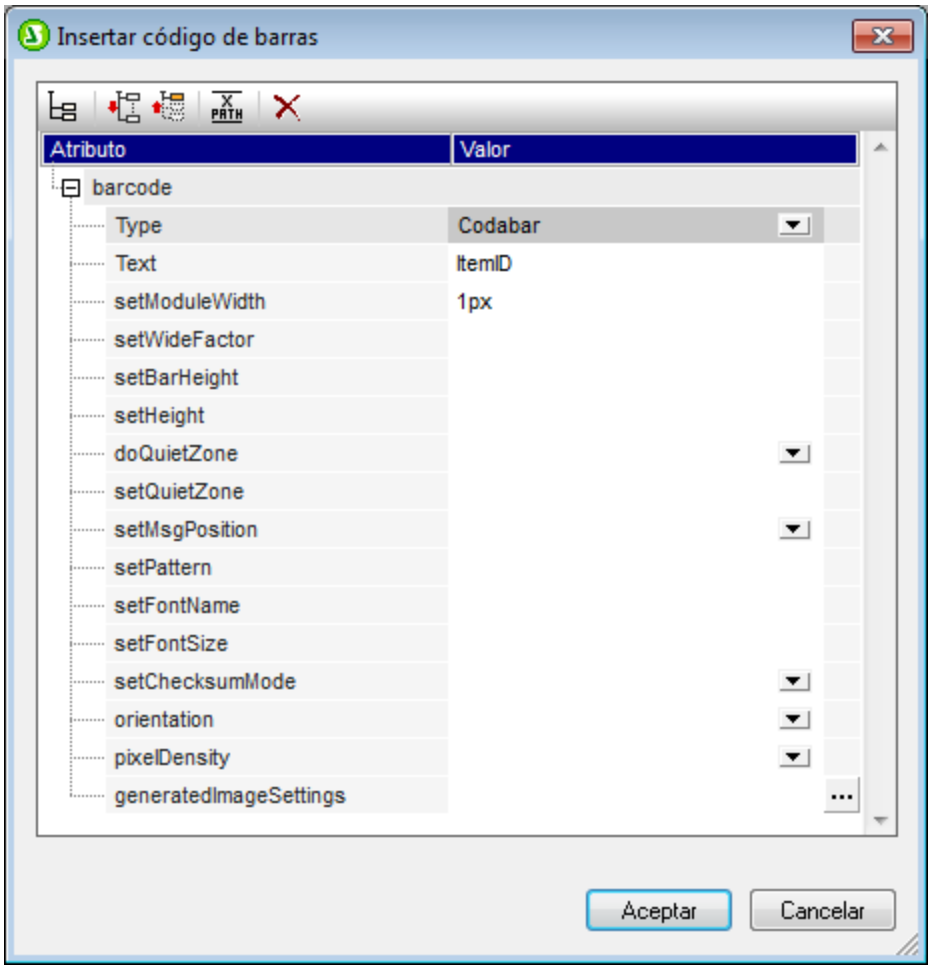
El comando **Párrafo especial** sirve para asignar un formato predefinido al nodo seleccionado. Los formatos predefinidos disponibles también se pueden seleccionar en un cuadro combinado de la barra de herramientas.

Cada tipo de párrafo tiene características de formato propias. Recuerde que con el tipo de formato `pre` puede representar los retornos de carro como tales en el documento de salida, en lugar de normalizarlos como espacios en blanco.

16.8.9 Código de barras

El comando **Insertar código de barras** abre el cuadro de diálogo "Insertar código de barras" (*imagen siguiente*)

 © 2016-2022 Altova GmbH



Las propiedades `Type` y `Text` de los códigos de barras son obligatorias. El resto son opcionales y tienen valores predeterminados. Cuando termine de definir el valor de las propiedades obligatorias y de otras propiedades, haga clic en **Aceptar**. El código de barras (p. ej. el de la imagen siguiente) se insertará en el diseño. Para más información sobre las propiedades de los códigos de barras consulte el apartado Códigos de barras 200.

Nota importante: para que los códigos de barras funcionen es necesario tener instalado Java Runtime Environment versión 1.4 o superior de 32 o 64 bits, dependiendo de la versión de StyleVision que utilice.

16.8.10 Imagen

El comando **Insertar imagen** abre el cuadro de diálogo "Insertar imagen" (*imagen siguiente*), donde puede elegir el tipo de imagen que desea insertar. Este comando también está disponible como icono en la barra de herramientas Insertar elementos de diseño 755.

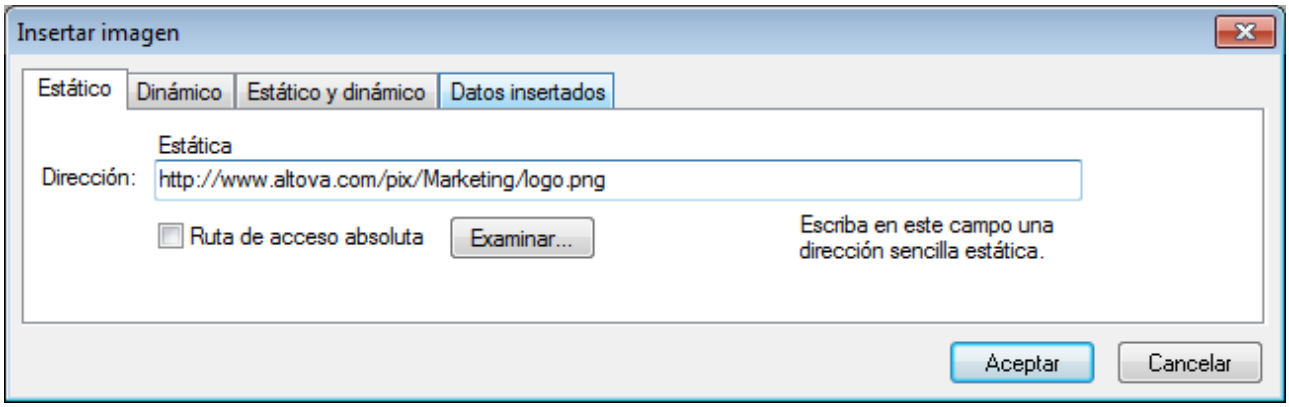
El cuadro de diálogo "Insertar imagen" tiene cuatro pestañas y cada una de ellas ofrece un método distinto de especificar la ubicación de la imagen:

- *Estático:* para introducir el URI de la imagen directamente.
- *Dinámico:* para obtener el URI de la imagen del documento XML o generarlo con una expresión XPath.
- *Estático y dinámico:* para combinar los dos métodos, estático y dinámico.
- *Datos insertados*: para seleccionar una imagen que está almacenada en un archivo XML como texto codificado de base 16 o base 64.

A continuación describimos estas pestañas más detalladamente.

Estático

El URI de la imagen se escribe directamente en el campo *Dirección* (*imagen siguiente*). En el ejemplo siguiente el URI de la imagen es `http://www.altova.com/pix/Marketing/logo.png`.

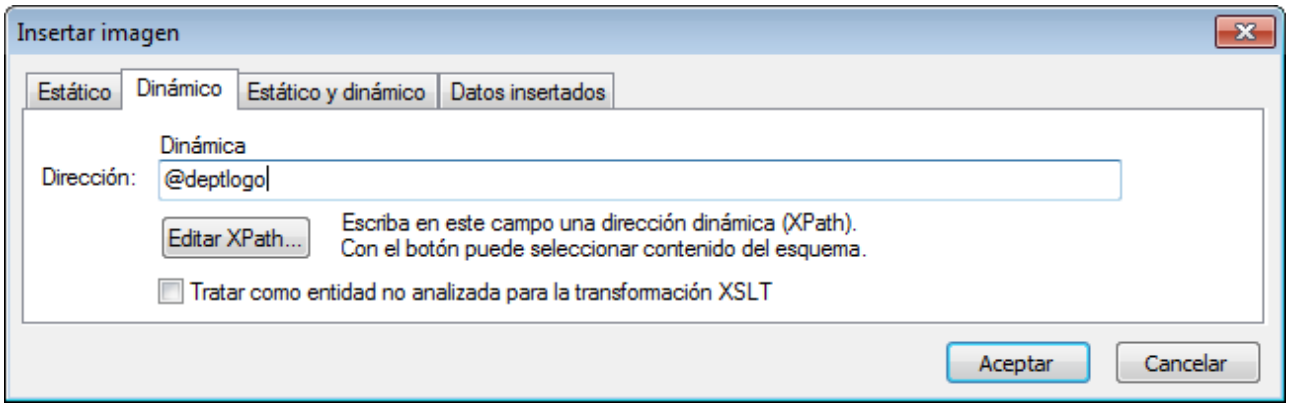


En esta pestaña puede especificar si el URI es absoluto (marque la casilla *Ruta de acceso absoluta*) o relativo (deje la casilla *Ruta de acceso absoluta* sin marcar). Si introduce un URI relativo, se resolverá en relación a la ubicación del archivo SPS. Para introducir un URI (absoluto o relativo) de forma automática, haga clic en **Examinar...** y busque el archivo de imagen.

Dinámico

En este caso se utiliza una expresión XPath que devuelve el URI de la imagen. En el ejemplo siguiente usamos la expresión XPath `@deptlogo`. Para ello suponemos que el URI de la imagen está almacenado en el atributo `deptlogo` del nodo de contexto. El nodo de contexto es el nodo del diseño donde estamos insertando la imagen.

 © 2016-2022 Altova GmbH

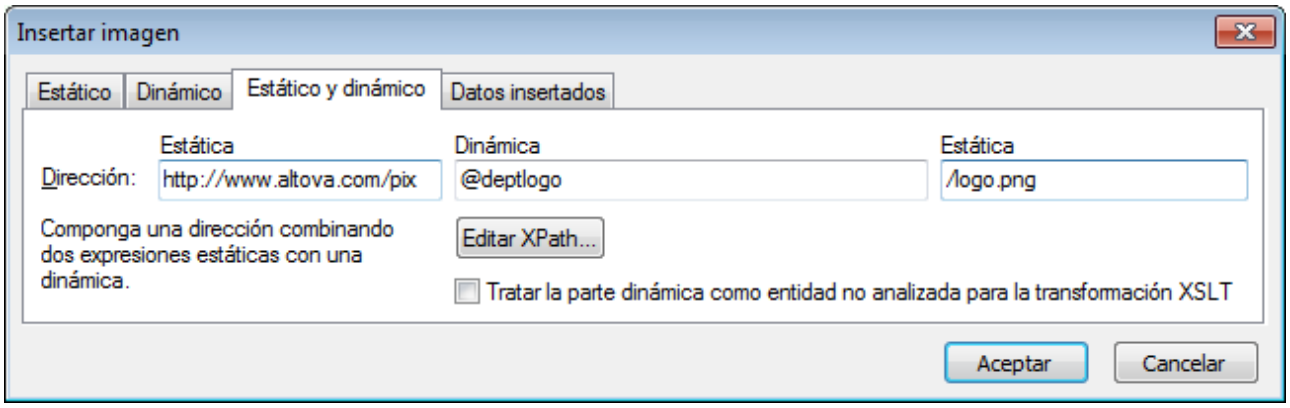


Haga clic en el botón **Editar XPath...** para abrir el Generador de expresiones XPath 744. En este cuadro de diálogo el nodo de contexto aparece resaltado en el árbol del esquema.

Si el diseño SPS está basado en un documento DTD y utiliza entidades sin analizar, puede usar una entidad sin analizar que remita al URI de la imagen. Para ello debe marcar la casilla *Tratar como entidad no analizada* y después escribir la expresión XPath que selecciona el nodo que contiene la entidad sin analizar. Para más información consulte el apartado URI de entidades sin analizar 469.

Estático y dinámico

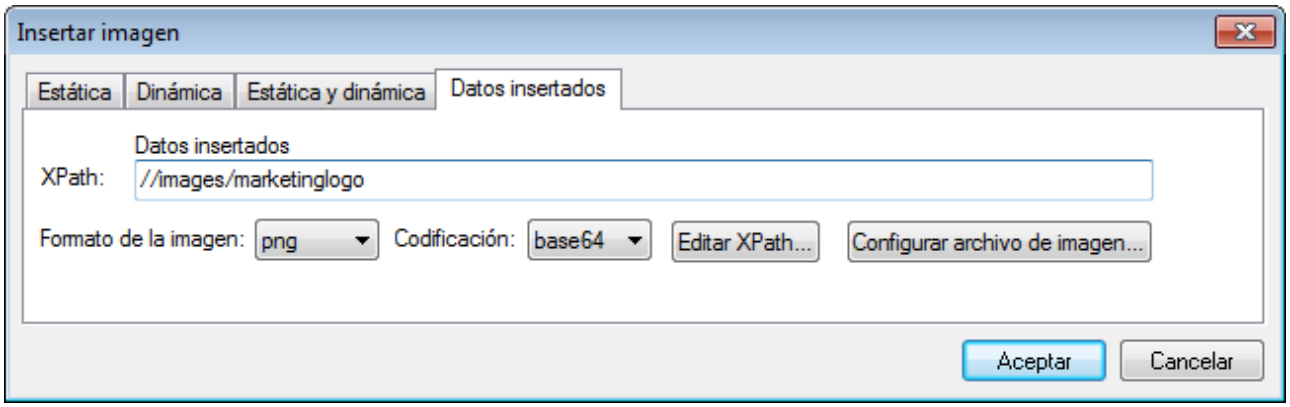
En esta pestaña puede usar el mecanismo estático y dinámico para generar el URI de la imagen.



Si el atributo `deptname` del nodo de contexto tiene el valor `Marketing`, el URI de la imagen del ejemplo anterior sería `http://www.altova.com/pix/Marketing/logo.png`. Recuerde que puede usar el Generador de expresiones XPath 744 para la dirección dinámica del URI.

Datos insertados

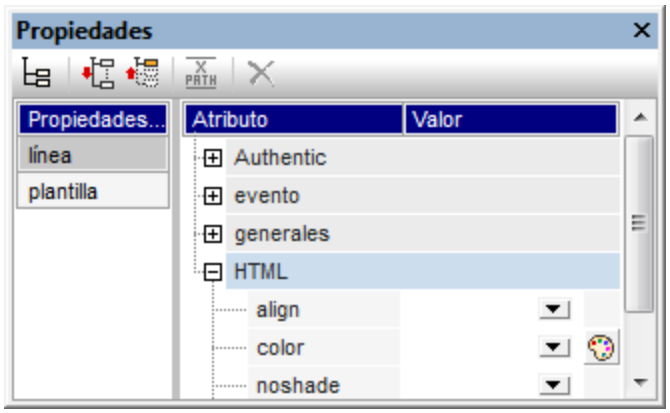
En los archivos XML se pueden almacenar imágenes como texto codificado en base 16 o base 64. La expresión XPath del cuadro de diálogo "Insertar imagen" (*imagen siguiente*) selecciona el nodo que contiene el texto codificado. El cuadro combinado *Codificación* sirve para especificar la codificación utilizada en el archivo XML de origen, para que StyleVision interprete correctamente el texto codificado. Además, el cuadro combinado *Formato de la imagen* indica en qué formato se debe generar el archivo de imagen. A partir de los datos de texto codificado se genera un archivo de imagen y este archivo se usa en el documento de salida.



En el cuadro de diálogo "Configuración del archivo de imagen generado" (que se abre haciendo clic en el botón **Configurar archivo de imagen...**) puede dar un nombre al archivo de imagen que se generará. Si lo prefiere, puede saltarse este paso y StyleVision generará un nombre automáticamente.

16.8.11 Línea horizontal

El comando **Insertar línea horizontal** inserta una línea horizontal en la posición del cursor. Este comando se deshabilita si está seleccionado un componente de diseño del SPS. Para definir las propiedades de la línea horizontal, seleccione la línea en la vista **Diseño** y en la ventana Propiedades seleccione el componente **`línea`**. Después especifique los valores de las propiedades del grupo `HTML` (*imagen siguiente*).

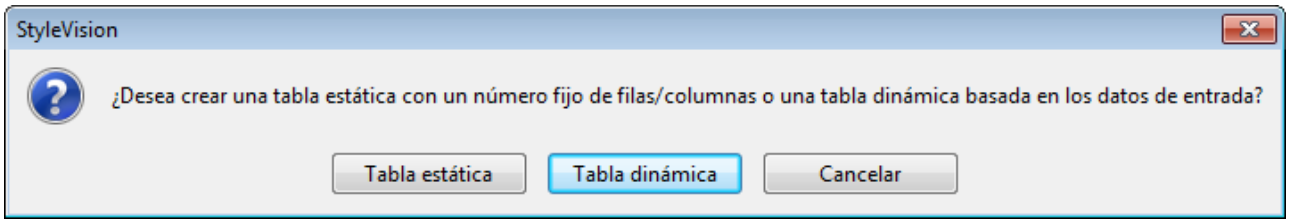


Estas son las propiedades HTML de las líneas horizontales: `color`, `size` (grosor), `width` (ancho en el diseño), `alignment` (alineación) y `noshade` (sin sombra).

16.8.12 Tabla

Al hacer clic en el comando **Insertar tabla** aparece un cuadro de diálogo donde debe confirmar el tipo de tabla que desea crear (*imagen siguiente*).

 © 2016-2022 Altova GmbH



En las secciones Tablas SPS estáticas 157 y Tablas SPS dinámicas 158 del manual encontrará información sobre cómo trabajar con cada tipo de tabla.

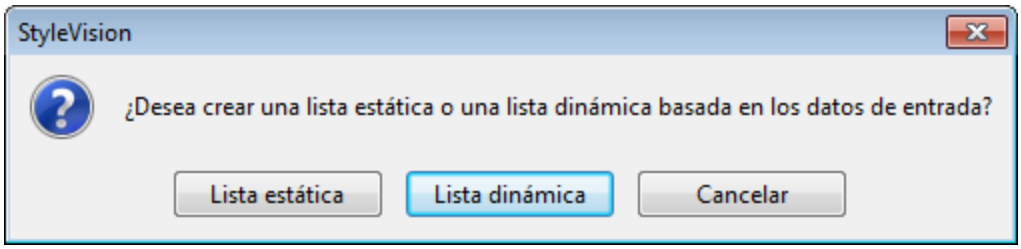
Recuerde que también puede crear tablas usando el icono **Insertar tabla** de la barra de herramientas Insertar elementos de diseño.

16.8.13 Numeración y viñetas

El comando **Insertar numeración y viñetas** sirve para crear una lista estática o dinámica en el diseño SPS. Los elementos de una lista estática se introducen directamente en la vista **Diseño**, mientras que los de las listas dinámicas corresponden a los valores de los nodos del mismo nivel del documento XML.

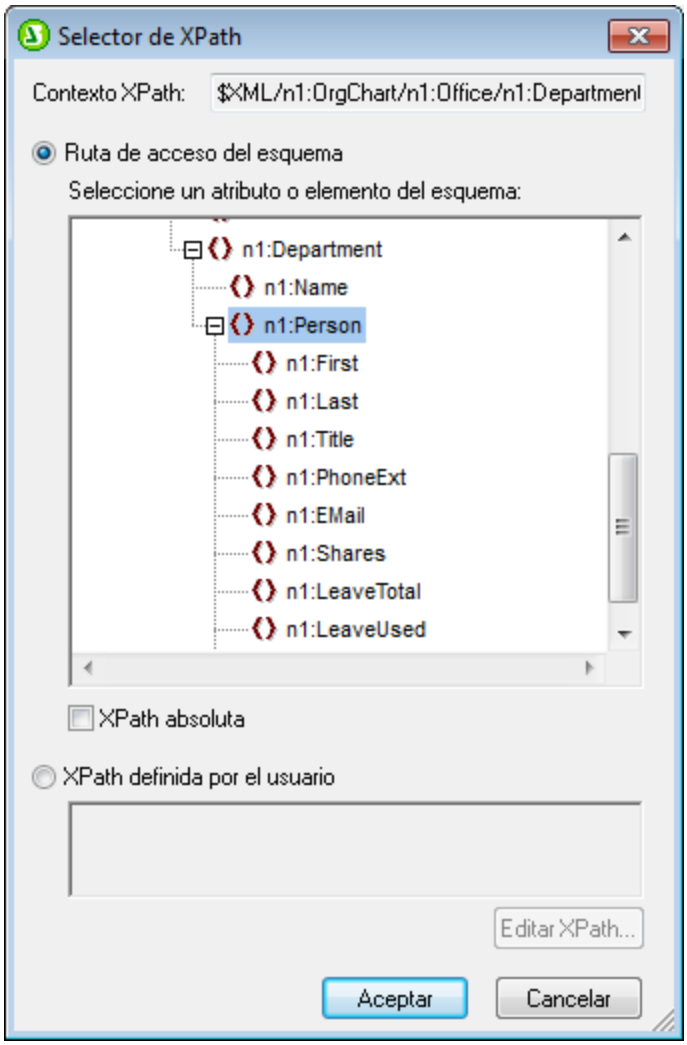
Siga estas instrucciones para crear una lista:

1. Ponga el cursor en la posición donde desea insertar la lista y haga clic en el comando de menú **Insertar | Insertar numeración y viñetas**. Aparece un aviso donde debe indicar qué tipo de lista desea crear, estática o dinámica (*imagen siguiente*).



   Si selecciona **Lista estática**, pase al paso 3 de estas instrucciones. Si selecciona **Lista dinámica**, aparece el cuadro de diálogo "Selector de XPath" (*imagen siguiente*).

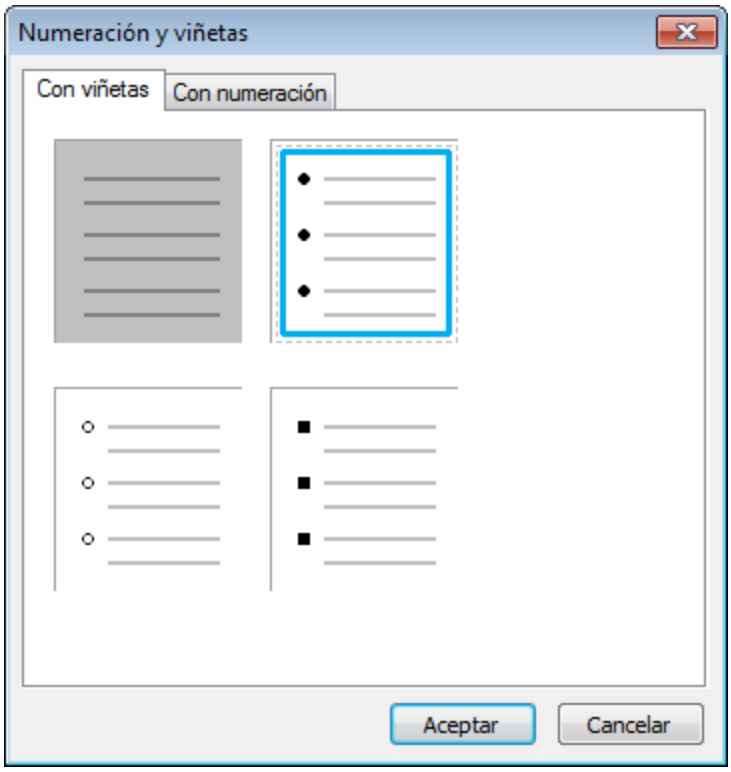
2. En el cuadro de diálogo "Selector de XPath" observe que el cuadro de texto *Contexto XPath* muestra el contexto del punto de inserción y que no se puede cambiar en el cuadro de diálogo. Seleccione el nodo que desea usar para crear la lista dinámica. En la imagen siguiente, por ejemplo, el nodo de contexto es `n1:Department` y la lista dinámica se creará a partir del nodo `n1:Person`. Esto significa que el contenido de cada nodo `n1:Person` dará lugar a un elemento en la lista.



Si selecciona el botón de opción *XPath definida por el usuario*, podrá introducir una expresión XPath para seleccionar el nodo con el que desea crear la tabla dinámica. Al hacer clic en **Aceptar** aparece el cuadro de diálogo "Numeración y viñetas".

3. En el cuadro de diálogo "Numeración y viñetas" seleccione el tipo de lista que desea crear. Puede elegir entre una lista con viñetas (con viñetas, círculos o cuadrados como marcadores de elemento de lista) o una lista numerada. Al hacer clic en **Aceptar** se crea crea una lista con el marcador de elementos de lista seleccionado.

© 2016-2022 Altova GmbH



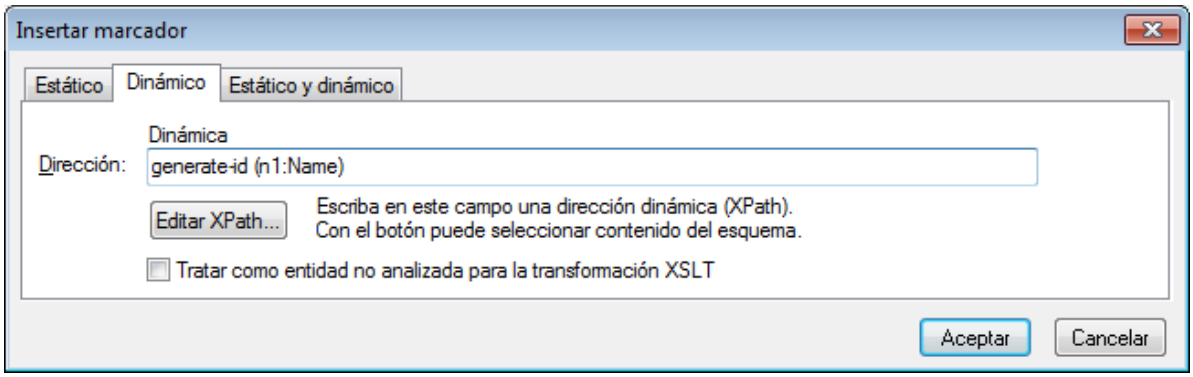
Nota: para crear una lista estática también puede hacer clic en el icono **Lista con viñetas** o **Lista numerada** de la barra de herramientas Insertar elementos de diseño 755 y después en la posición del diseño donde desea crear la lista. Para crear una lista dinámica también puede arrastrar un nodo desde la ventana Estructura del esquema hasta la vista **Diseño**.

16.8.14 Marcador

El comando **Insertar marcador** sirve para insertar un marcador (o anclaje) en la posición elegida del diseño SPS. Después puede hacer referencia al marcador por medio de un hipervínculo 826.

Siga estos pasos para insertar un marcador:

1. Ponga el cursor en la posición del diseño donde desea crear el marcador.
2. Seleccione el comando de menú **Insertar | Insertar marcador** o haga clic con el botón derecho y seleccione **Insertar marcador**. Aparece el cuadro de diálogo Insertar marcador 364 (*imagen siguiente*).



3. En este cuadro de diálogo seleccione la pestaña que más le convenga, dependiendo de si quiere crear un marcador estático (pestaña *Estático*), un marcador dinámico que se obtiene en el documento XML (pestaña *Dinámico*) o un marcador compuesto con una parte estática y otra dinámica (pestaña *Estático y dinámico*). En la imagen anterior, por ejemplo, se creó un marcador dinámico, cuyo nombre es el ID único de cada nodo secundario llamado `Name` del nodo de contexto.
4. Haga clic en **Aceptar** para crear el marcador.

Nota: los marcadores se crean en la posición indicada del diseño. Si la posición elegida está dentro de un elemento que se repite, entonces se crea un marcador dentro de cada instancia del elemento. Si se le da un nombre estático, cada marcador tendrá el mismo nombre. Por tanto, en el caso de marcadores situados en elementos que se repiten, es mejor darles un nombre dinámico (p. ej. el nombre de un elemento secundario del nodo de contexto). Si el nodo seleccionado para el nombre dinámico tiene el mismo contenido en varias instancias, puede utilizar la función `generate-id()` para generar un nombre único para el marcador. Para hacer referencia a este tipo de marcadores, se puede generar un Id. igual al valor `href` de un hipervínculo 369. En este caso asegúrese de usar el identificador de fragmentos `#` delante de la función `generate-id()`. La expresión XPath sería: `concat('#', generate-id(nodoXXX))`.

Después de crear el marcador puede editar su nombre. Para ello haga clic con el botón derecho en el marcador y seleccione **Editar nombre del marcador** en el menú contextual. Otra opción es editar el nombre en la ventana Propiedades: expanda el grupo de propiedades marcador y haga doble clic en el campo de valor de la propiedad `nombre del marcador`.

Eliminar un marcador

Para eliminar un marcador selecciónelo en el diseño y pulse la tecla **Suprimir**.

16.8.15 Hipervínculo

El comando **Insertar hipervínculo** sirve para insertar un enlace entre cualquier parte del documento de salida (HTML o RTF) y un anclaje del documento de salida o de un documento externo o fragmento del documento. Recuerde que los enlaces se crean solamente en el documento de salida y que no están disponibles en la vista Authentic.

Siga estas instrucciones para insertar un hipervínculo:

 © 2016-2022 Altova GmbH

1. El hipervínculo se puede crear alrededor de un componente de diseño ya existente o en cualquier posición del diseño (y después se inserta el texto del hipervínculo). Seleccione el componente de diseño o el fragmento de texto que desea convertir en hipervínculo o ponga el cursor en la posición del diseño donde desea insertar el hipervínculo.
2. Haga clic en el icono **Insertar hipervínculo** de la barra de herramientas Insertar elementos de diseño 755 o seleccione el comando de menú **Insertar | Insertar hipervínculo.** También puede hacer clic con el botón derecho y elegir **Insertar hipervínculo** (si no seleccionó ningún componente de diseño) o **Incluir en hipervínculo** (si seleccionó un componente de diseño).
3. Aparece el cuadro de diálogo Insertar hipervínculo 366, donde puede especificar a qué documento o fragmento del documento se enlaza. Si quiere crear un enlace con un fragmento de documento (es decir, con un marcador del documento), recuerde que debe incluir el símbolo `#`. El URI del hipervínculo puede indicarse de tres formas distintas:

   - Con una dirección estática: se inserta directamente y puede seleccionar un archivo XML con el botón **Examinar** o un fragmento del documento actual con el botón **Marcador**. Por ejemplo: `http://www.altova.com` (URI estático de página web), `U:\documentation\index.html` (introducido con ayuda del botón **Examinar**) o `#top_of_page` (introducido con ayuda del botón **Marcador**).
   - Con una dirección dinámica: se obtiene del nodo del documento XML especificado en el cuadro de diálogo. Por ejemplo: `//otherdocs/doc1`. Si el nombre del marcador se generó usando la función `generate-id()`, el atributo `href` del hipervínculo debe generarse usando la misma función `generate-id()`. Para más información consulte el apartado Definir hipervínculos 369.
   - Con una dirección estática y dinámica: debe indicar el texto estático y el nodo del documento XML en el cuadro de diálogo. Por ejemplo: `www.altova.com -- department/name -- #intropara`.

4. Cuando termine haga clic en **Aceptar** para crear el hipervínculo.

Nota: cuando especifique el nodo para un hipervínculo dinámico, puede introducir una expresión XPath absoluta (marcando la casilla *XPath absoluta*). Si no marca esta casilla, la expresión XPath será relativa al nodo de contexto, que es el nodo donde se está insertando el hipervínculo.

Usar entidades sin analizar

Si el diseño SPS está basado en un documento DTD, en la parte dinámica de la dirección del hipervínculo podrá usar el URI declarado de una entidad sin analizar de la DTD. Para más información consulte el apartado Usar los URI de entidades sin analizar 469.

Editar un hipervínculo

Después de crear el hipervínculo puede editar su atributo `href`. Para ello haga clic con el botón derecho en el hipervínculo y seleccione **Editar URL** en el menú contextual. Otra opción es editar el atributo en la ventana Propiedades. Primero seleccione el componente **`hipervínculo`** y extienda el grupo de propiedades *hipervínculo*. Después haga doble clic en el campo de valor de la propiedad `URL`.

Eliminar un hipervínculo

Para eliminar un hipervínculo debe seleccionarlo y pulsar la tecla **Suprimir**.

16.8.16 Nota al pie

Puede añadir notas al pie en un documento añadiendo el componente `Nota al pie` (**Insertar | Insertar nota al pie**) en la posición donde quiere que aparezca el número de nota al pie. Las notas al pie se pueden usar en los formatos para medios impresos(RTF).

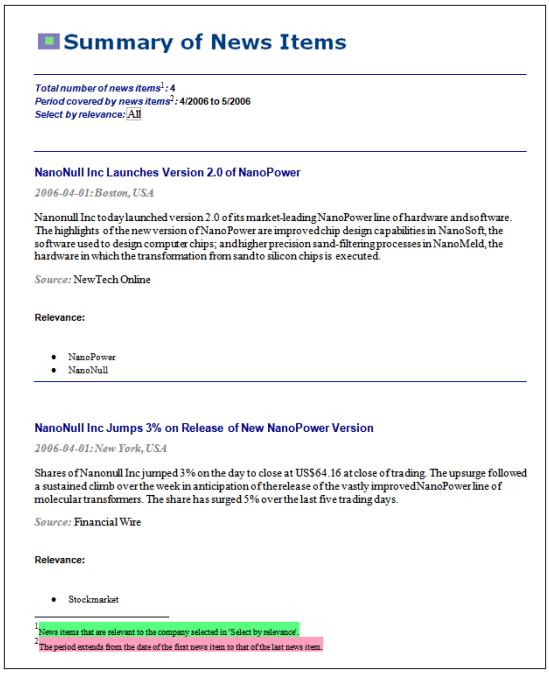
Cuando trabaje con notas al pie debe tener en cuenta que:

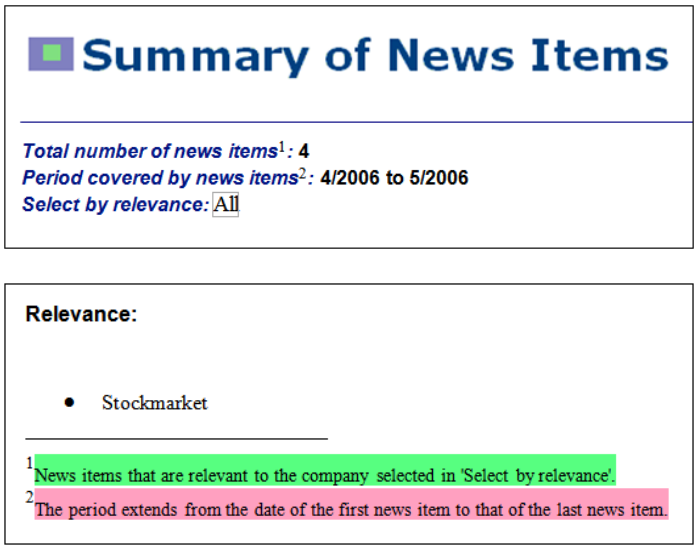
- El texto de la nota al pie debe colocarse entre las etiquetas del componente `Nota al pie` y puede aplicar formato al texto de la nota al pie.
- En el documento de salida el número de la nota al pie aparece en la posición donde se añadió la nota al pie. El texto, sin embargo, aparece al final de la página junto con el número de nota al pie que le corresponde.
- En el documento de salida el texto de la nota al pie tendrá el formato del texto que se insertó dentro del componente `Nota al pie` en el diseño.
- En el documento de salida las notas al pie se numeran automáticamente.
- Si se generan varios documentos de salida, la numeración de las notas al pie empieza desde cero en cada documento de salida.

En el ejemplo de la siguiente imagen se insertaron tres componentes `Nota al pie`. El texto de la nota al pie se insertó entre las etiquetas del componente y se le aplicó formato.

En las siguientes imagen puede ver el resultado. A la izquierda puede ver la página entera y a la derecha puede ver en detalle el número de las notas al pie y el texto de la nota al pie.

© 2016-2022 Altova GmbH





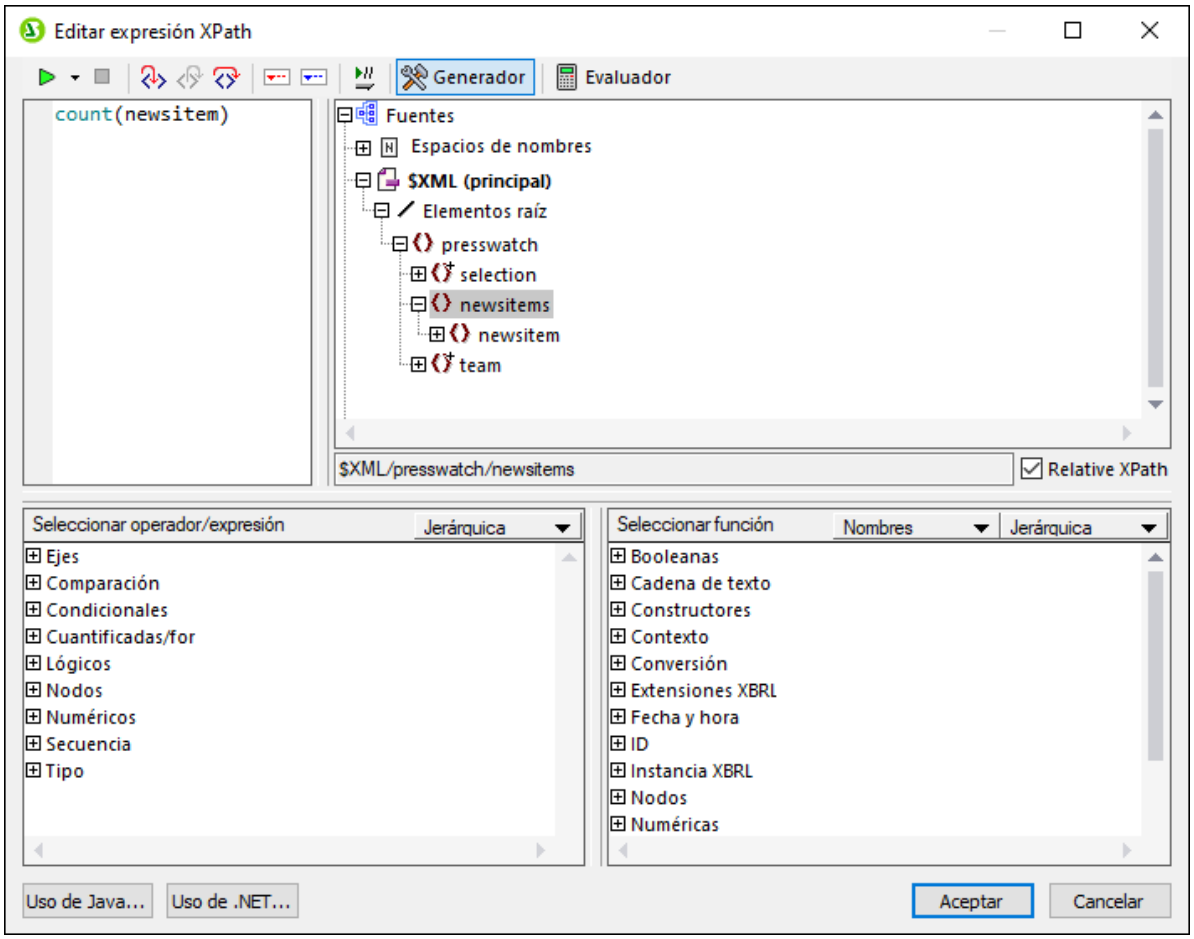
Nota: no se puede aplicar formato a los números de las notas al pie.

16.8.17 Condición y condición basada en el resultado

El comando **Condición** sirve para insertar una condición en la posición del cursor o alrededor del componente seleccionado. Una condición está compuesta por una o varias ramas y cada rama contiene un conjunto concreto de reglas de procesamiento. De este modo podrá especificar reglas de procesamiento distintas para cada rama. Por ejemplo, una rama puede comprobar si el contenido de un nodo es la cadena `Stop` y, si lo es, marcar el contenido del nodo en color rojo. Otra rama puede comprobar si el contenido del nodo es la cadena `Go` y, si lo es, marcar el contenido del nodo en color verde. La tercera rama puede comprobar si el contenido del nodo no es ni `Stop` ni `Go` y, si así es, marcar el contenido del nodo en color negro.

Siga estos pasos para insertar una condición en el diseño:

1. Ponga el cursor en la posición del diseño elegida para la condición o seleccione el componente que desea convertir en condición.
2. Seleccione el comando de menú **Insertar | Insertar condición** o haga clic con el botón derecho y elija **Insertar condición** en el menú contextual.
3. Se abre el cuadro de diálogo Editar expresión XPath 729 (*imagen siguiente*), donde puede escribir la expresión XPath para la condición.



El nodo de contexto para la expresión aparece resaltado en el panel inferior izquierdo, que muestra la estructura del esquema. Puede escribir la expresión XPath directamente en el cuadro de texto *Expresión XPath* o puede hacer doble clic en los elementos de los paneles inferiores para insertarlos en el cuadro de texto. Los nodos de la estructura del esquema del panel izquierdo se insertan en relación al nodo de contexto (si marca la casilla *XPath relativa*) o como expresión absoluta empezando por el nodo de documento (si no marca la casilla *XPath relativa*).

4. Haga clic en **Aceptar** para terminar e insertar la condición, que se crea con una sola rama que comprueba la expresión XPath introducida en el paso anterior.

Insertar condición basada en el resultado

Este comando inserta una condición basada en el resultado en la posición del cursor o alrededor del componente seleccionado. Cada condición de la rama equivale a un formato de salida diferente (vista Authentic, RTF, texto o HTML). Para ver qué tipo de formato de salida representa una rama, pase el puntero del ratón por encima de la etiqueta de la rama o mire qué expresión XPath tiene la rama seleccionada (en la ventana Propiedades, en la entrada rama condición, haga clic en el botón 307 **Edición** 307). Si la condición basada en el resultado se creó en el punto de inserción del cursor, todas las ramas estarán vacías y deberá introducir contenido para cada rama. Si la condición basada en el resultado se creó alrededor de un componente, cada rama contendrá ese componente. Para más información consulte el apartado sobre condiciones basadas en el resultado 307. Estas condiciones se pueden editar, mover y eliminar como si fuera una condición normal y corriente.

Editar las expresiones XPath de las ramas condicionales

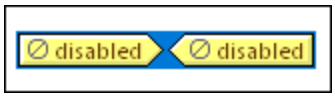
Si quiere editar la expresión XPath de una rama seleccione la rama en la vista **Diseño**. Después, en la ventana Propiedades, seleccione el componente **`rama condicional`** y expanda el grupo de propiedades *cuando*. Haga doble clic en el campo de valor de la propiedad `XPath`. Esto abre el cuadro de diálogo "Editar expresión XPath" (*imagen anterior*), donde puede editar la expresión. Cuando termine haga clic en **Aceptar**.

Agregar ramas, cambiar el orden de las ramas y eliminar ramas

Para agregar ramas nuevas, cambiar el orden de las ramas y eliminar ramas puede hacer clic con el botón derecho en la rama pertinente y seleccionar el comando correspondiente del menú contextual.

16.8.18 Deshabilitado

El comando Deshabilitado inserta un componente de tipo deshabilitado donde se encuentre el cursor (*imagen siguiente*). (Para colocar contenido entre dos componentes de este tipo use el comando **Incluir en | Deshabilitado** 842.)



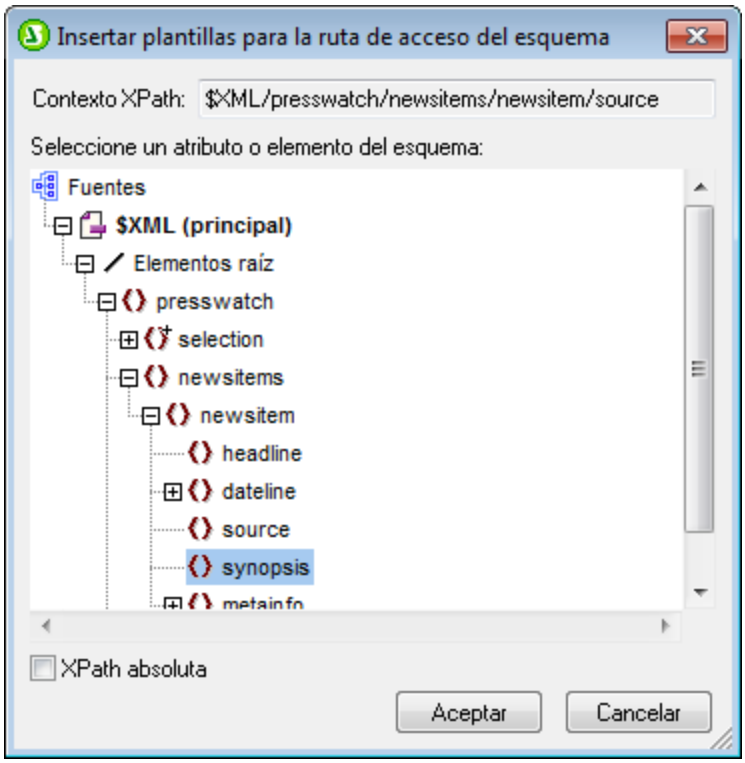
El contenido que se ecuentra dentro del componente de tipo deshabilitado (*disabled*) se ignora en el resultado. Esto quiere decir que puede usar los componentes de tipo deshabilitado para omitir en el resultado el contenido que quiere ignorar.

Si quiere restaurar ese contenido solo tiene que eliminar las etiquetas Disabled entre las que se encuentra ese contenido. Para ello, seleccione el componente de tipo deshabilitado, haga clic con el botón derecho y seleccione **Quitar solo la etiqueta**.

16.8.19 Plantilla

El comando **Insertar plantilla** inserta en la posición del cursor una plantilla vacía para el nodo seleccionado de la estructura del esquema. Siga estas instrucciones para insertar una plantilla.

1. Ponga el cursor en la posición del diseño donde desea insertar la plantilla.
2. Haga clic en el comando de menú **Insertar | Insertar plantilla**. Aparece el cuadro de diálogo "Insertar plantillas..." (*imagen siguiente*).



3. El campo *Contexto XPath* muestra el nodo de contexto del punto de inserción y será el nodo de contexto para la plantilla. En la estructura del esquema seleccione el nodo para el que desea usar la plantilla. Por ejemplo, en la imagen anterior elegimos crear una plantilla para el nodo `synopsis`.
4. Cuando termine haga clic en **Aceptar**.

Como resultado se crea una plantilla vacía para el nodo seleccionado (*imagen siguiente*).

16.8.20 Plantilla definida por el usuario

El comando **Insertar plantilla definida por el usuario** inserta en la posición del cursor una plantilla vacía que selecciona el nodo especificado por el usuario mediante una expresión XPath. Siga estas instrucciones para insertar una plantilla definida por el usuario:

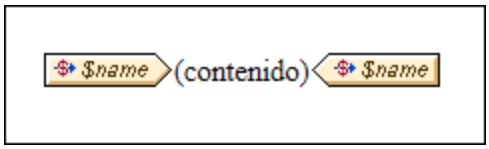
1. Ponga el cursor en la posición del diseño donde desea insertar la plantilla.
2. Haga clic en el comando de menú **Insertar | Insertar plantilla definida por el usuario**. Aparece el cuadro de diálogo Editar expresión XPath 729.
3. Escriba la expresión XPath que selecciona el nodo elegido. En este momento debe tener en cuenta un par de cosas: (i) la expresión XPath se evaluará en el contexto del nodo donde se está creando la plantilla y (ii) la expresión XPath puede seleccionar cualquier nodo del documento y de cualquier otro documento XML.
4. Cuando termine de construir la expresión XPath haga clic en **Aceptar**.

© 2016-2022 Altova GmbH

En el diseño se inserta una plantilla definida por el usuario vacía para el nodo de destino seleccionado. Consulte el apartado Archivos SPS: contenido | plantillas definidas por el usuario 148 para obtener más información.

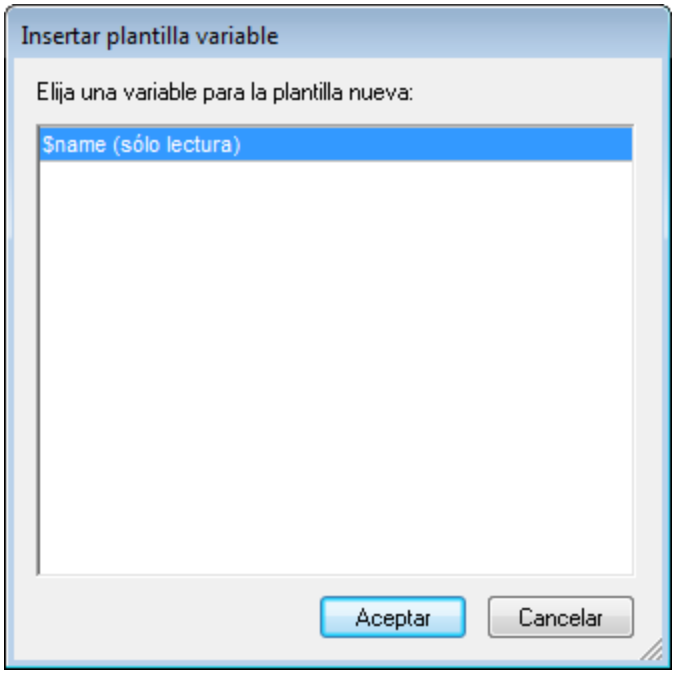
16.8.21 Plantilla variable

Una plantilla variable es una plantilla que apunta a una variable y que por defecto genera su contenido. Las plantillas variables se insertan con los comandos **Insertar | Insertar plantilla variable** o en **Incluir en | Plantilla variable**. Ambos comandos insertan una plantilla para una variable definida en el SPS. La plantilla variable (*imagen siguiente*) contiene por defecto el marcador de posición `(contenido)` y este marcador sirve para generar el contenido de la variable. Si quiere, puede insertar contenido adicional (estático o dinámico) dentro de la plantilla variable o modificarla como si fuera una plantilla normal y corriente. En su representación gráfica en la vista **Diseño** las etiquetas de la plantilla variable incluyen el símbolo del dólar (*imagen siguiente*).



Para insertar una plantilla variable:

1. En la vista **Diseño** ponga el cursor en la posición donde desea insertar la plantilla variable.
2. Haga clic en el comando de menú **Insertar | Insertar plantilla variable**. Aparece el cuadro de diálogo "Insertar plantilla variable" (*imagen siguiente*).



3. El cuadro de diálogo enumera todas las variables y los parámetros declarados por el usuario 327 y definidos en el archivo SPS. Seleccione la variable para la que desea crear la plantilla.
4. Para terminar haga clic en **Aceptar**.

16.8.22 Fragmento de diseño

Al pasar el puntero del ratón por encima del comando **Insertar fragmento de diseño** aparece un submenú que enumera todos los fragmentos de diseño que hay disponibles en el diseño. Haga clic en un fragmento de diseño del submenú para insertar el fragmento en el punto de inserción.

16.8.23 Contenedor de diseño, cuadro de diseño, línea

El comando **Insertar contenedor de diseño** sirve para insertar un contenedor de diseño en el SPS. Después de crear el contenedor de diseño puede insertar en él cuadros de diseño y líneas. Es decir, los comandos **Insertar cuadro de diseño** e **Insertar línea** se habilitan después de haber creado un contenedor de diseño.

Para insertar contenedores de diseño, cuadros de diseño y líneas también puede usar los iconos de la barra de herramientas Insertar elementos de diseño 755: seleccione el icono del componente que desea insertar y después haga clic en la posición del diseño donde desea insertar el componente.

16.8.24 Tabla de contenido

Al pasar el puntero del ratón por el comando **Insertar tabla de contenido** aparece un submenú con varios comandos para crear una plantilla de tabla de contenido (TDC), marcadores TDC y la estructura del documento necesaria para la TDC.

Los comandos del submenú **Insertar tabla de contenido** aparecen a continuación. Haga clic en los enlaces para consultar el apartado dedicado a cada uno de estos componentes de la tabla de contenido.

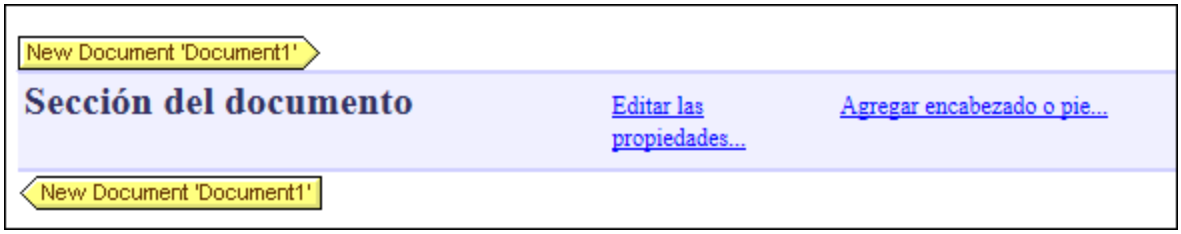
- Insertar tabla de contenido 347
- Marcador TDC 344
- Marcador TDC (Asistente) 340
- Referencia TDC 350
- Referencia TDC | Entrada de texto / Línea de guía / Referencia de página 351
- Numeración jerárquica 351
- Numeración secuencial 351
- Nivel de TDC 341
- Referencia de nivel de TDC 349
- La plantilla funciona como nivel 341

Nota: estos comandos también están disponibles en el menú contextual que aparece al hacer clic con el botón derecho en el diseño. Sin embargo, no todos los comandos están siempre disponibles, depende de la posición del diseño donde haga clic con el botón derecho.

© 2016-2022 Altova GmbH

16.8.25 Documento nuevo

El comando **Insertar documento nuevo** inserta una plantilla de documento nuevo (*imagen siguiente*) en la posición del cursor.



La plantilla de documento nuevo contiene una sección inicial del documento vacía, donde puede insertar contenido. Si quiere, puede anexar más secciones de documento a la sección inicial del documento con ayuda del comando **Insertar | Insertar página, columna o sección de documento**.

La plantilla de documento nuevo crea un documento nuevo de salida. Es decir, los resultados estarán compuestos por varios documentos de salida.

16.8.26 Página, columna, sección de documento

El comando **Insertar página, columna o sección de documento** está pensado para formatos de salida para medios impresos. Con este comando podrá insertar saltos de página (formato de salida HTML y RTF) y números de página (formato de salida RTF). Estos componentes solo se pueden insertar en la posición del cursor (es decir, no se pueden insertar alrededor de una selección).

Página nueva

Haga clic en el comando **Insertar | Insertar página, columna o sección de documento | Página nueva** para insertar un salto de página en el punto de inserción. El salto de página se ilustra con una línea discontinua que cruza toda la vista **Diseño**. En el resultado HTML, el salto de página no afecta a la vista Explorador pero sí afecta a la copia impresa de la vista del documento. En el resultado RTF el salto de página se inserta en la posición especificada.

Número de página

Haga clic en el comando **Insertar | Insertar página, columna o sección de documento | Número de página** para insertar el número de página actual en los documentos de salida RTF. En el documento de salida el número de página aparece en forma de bloque (es decir, en una línea distinta) o incrustado (es decir, dentro del texto) dependiendo de la posición donde insertó el número de página en el diseño SPS. Por ejemplo, si inserta el número de página dentro de un párrafo, en el documento de salida el número de página aparece incrustado dentro del párrafo. Si por el contrario inserta el número de página entre dos elementos, en el documento de salida aparecerá en una línea distinta.

Total de páginas

Haga clic en el comando **Insertar | Insertar página, columna o sección de documento | Total de páginas** para insertar el número total de páginas en los documentos de salida PDF. El total de páginas puede

insertarse en cualquier parte del documento, incluidos los encabezados y pies de página. Este comando es muy práctico a la hora de numerar las páginas del documento de salida. Por ejemplo, puede crear este encabezado: `Página (número de página) / (total de páginas)`. El resultado en el documento de salida sería `Página: 1/25`, por ejemplo.

Columna nueva

El número de columnas que debe tener una página de una sección determinada se especifica en las propiedades de página 429 de la sección. En el documento de salida el texto se repartirá en columnas en una página de varias columnas. Sin embargo, el texto se puede pasar a una columna nueva insertando un salto de columna (o columna nueva) en el diseño. Para insertar una columna nueva en el documento, ponga el cursor donde desea crear la sección nueva y haga clic en el comando **Insertar | Insertar página, columna o sección de documento | Columna nueva**.

Sección de documento nueva

Los documentos están compuestos por una sección inicial y varias secciones adicionales. Cada sección 420 tiene sus propias propiedades de página. Para insertar una sección nueva, ponga el cursor en la posición donde desea insertarla y haga clic en el comando **Insertar | Insertar página, columna o sección de documento | Sección de documento nueva**.

Eliminar saltos de página, números de página y el total de páginas
Para eliminar saltos de página, números de página y el total de páginas, seleccione el marcador de posición y pulse la tecla **Suprimir**.

16.8.27 Elemento definido por el usuario

Al pasar el puntero del ratón por el comando **Insertar elemento definido por el usuario** aparece un submenú que incluye comandos para insertar un **Elemento definido por el usuario** 151 o un **Bloque de texto XML definido por el usuario** 152. Para aprender a usar estos dos componentes consulte el apartado Archivos SPS: contenido | Elementos definidos por el usuario, Bloques de texto XML 151.

© 2016-2022 Altova GmbH

16.9 Menú Incluir en

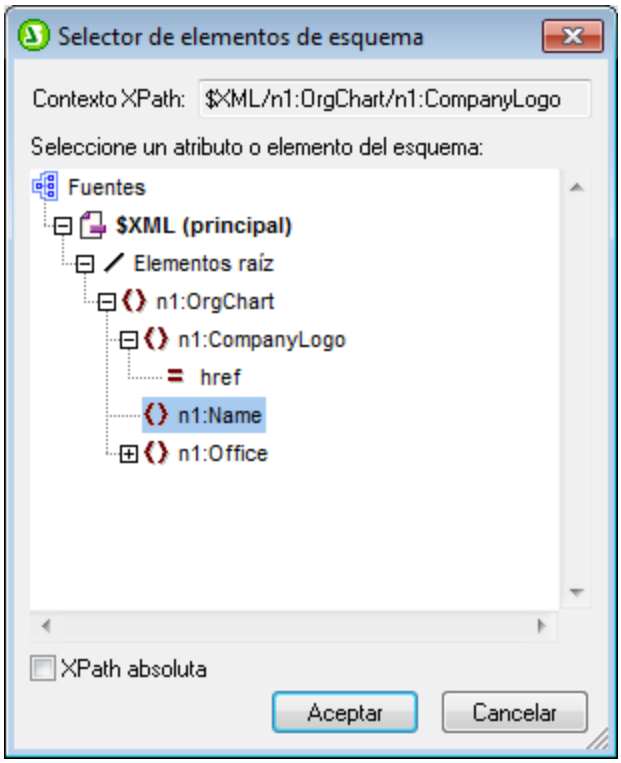
Con los comandos del menú **Incluir en** podrá crear componentes de diseño alrededor del texto o componente seleccionado en la vista **Diseño**. En otras palabras, podrá incluir la selección dentro de un componente de diseño. Algunos de estos comandos están disponibles como iconos en barras de herramientas 748 (también puede encontrar comandos equivalentes en el **Menú Insertar** 812). Además, los comandos del menú **Incluir en** están disponibles en los menús contextuales que aparecen al hacer clic con el botón derecho en la selección en la vista **Diseño**. Los comandos que no corresponden a la selección se deshabilitan.

Nota: los comandos del menú **Incluir en** solamente están disponibles en la vista **Diseño** porque sirven para construir el diseño SPS.

16.9.1 Plantilla

El comando **Incluir en | Plantilla** incluye el componente de diseño o el texto seleccionado dentro de una plantilla para un nodo de esquema determinado. Estas son las instrucciones de uso:

1. Seleccione el componente de diseño o el texto que desea incluir dentro de una plantilla.
2. Haga clic en el comando de menú **Incluir en | Plantilla**. Se abre el cuadro de diálogo "Selector de elementos de esquema" (*imagen siguiente*).



3. En el cuadro *Contexto XPath* aparece el nodo de contexto de la selección, que funcionará como nodo de contexto de la plantilla cuando esta se cree. Seleccione el nodo para el que desea crear la plantilla. En la imagen anterior, por ejemplo, el nodo seleccionado para crear la plantilla es el nodo `n1:Name`.

4. Cuando termine haga clic en **Aceptar**.

Como resultado alrededor de la selección se crea una plantilla para el nodo seleccionado.

16.9.2 Plantilla definida por el usuario

El comando **Incluir en | Plantilla definida por el usuario** incluye la selección dentro de una plantilla para un nodo indicado por el usuario con ayuda de una expresión XPath. Estas son las instrucciones de uso:

1. Seleccione el componente o texto que desea incluir dentro de una plantilla definida por el usuario.
2. Haga clic en el comando de menú **Incluir en | Plantilla definida por el usuario**. Se abre el cuadro de diálogo Editar expresión XPath 729.
3. Escriba la expresión XPath que selecciona el nodo elegido. En este momento debe tener en cuenta un par de cosas: (i) la expresión XPath se evaluará en el contexto del nodo donde se está creando la plantilla definida por el usuario y (ii) la expresión XPath puede seleccionar cualquier nodo del documento, pero también de otros documentos XML.
4. Cuando termine de construir la expresión XPath haga clic en **Aceptar**.

Alrededor de la selección se crea una plantilla definida por el usuario para el nodo elegido.

Para más información consulte el apartado Archivos SPS: estructura | Plantillas y fragmentos de diseño | Plantillas definidas por el usuario 270.

16.9.3 Plantilla variable

El comando **Incluir en | Plantilla variable** incluye la selección dentro de una plantilla para una variable definida en el diseño SPS. Estas son las instrucciones de uso:

1. Seleccione el texto o el componente del diseño que desea incluir en una plantilla variable.
2. Haga clic en el comando **Incluir en | Plantilla variable**. Se abre el cuadro de diálogo Incluir en plantilla variable 273.
3. El cuadro de diálogo enumera las variables del diseño SPS. Seleccione la variable para la que desea crear la plantilla.
4. Haga clic en **Aceptar** para terminar.

Alrededor de la selección se crea una plantilla variable.

16.9.4 Párrafo, párrafo especial

El comando **Párrafo** inserta un elemento de párrafo HTML `<p>` alrededor del componente seleccionado. Se considera que el componente está seleccionado si todo el nodo está seleccionado (tras hacer clic en una de sus etiquetas) o si hay texto estático seleccionado. Si pone el cursor dentro de texto estático, el elemento de párrafo se inserta en esa posición (su etiqueta inicial y final). También puede insertar un párrafo haciendo clic en el icono **Insertar párrafo** de la barra de herramientas Insertar elementos de diseño 755.

 © 2016-2022 Altova GmbH

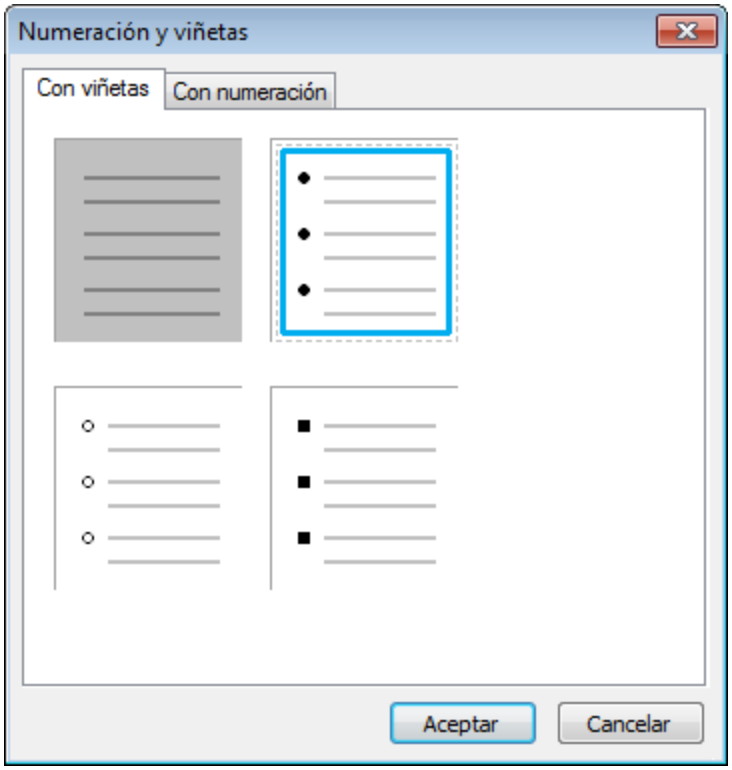
El comando **Párrafo especial** sirve para asignar un formato predefinido al nodo seleccionado. Los formatos predefinidos disponibles también se pueden seleccionar en un cuadro combinado de la barra de herramientas.

Cada tipo de párrafo tiene características de formato propias. Recuerde que con el tipo de formato `pre` puede representar los retornos de carro como tales en el documento de salida, en lugar de normalizarlos como espacios en blanco.

16.9.5 Numeración y viñetas

El comando **Incluir en | Numeración y viñetas** crea una lista estática y elementos de lista alrededor del componente o texto seleccionado. Si la selección contiene un retorno de carro o salto de línea (insertado con la tecla **Entrar**), entonces se crean elementos de lista distintos por cada fragmento de texto que esté separado con un retorno de carro o salto de línea.

Al seleccionar el comando de menú **Incluir en | Numeración y viñetas** aparece el cuadro de diálogo "Numeración y viñetas" (*imagen siguiente*).



Seleccione qué tipo de marcador de elemento de lista desea usar y haga clic en **Aceptar**. La lista se crea alrededor de la selección. El número de elementos de lista equivale al número de retornos de carro y saltos de línea de la selección. Si quiere, puede añadir elementos de lista nuevos pulsando la tecla **Entrar**.

Nota: otra manera de hacer esto es seleccionar el contenido estático y hacer clic en el icono **Lista con viñetas** o **Lista numerada** de la barra de herramientas Insertar elementos de diseño 755.

16.9.6 Marcadores e hipervínculos

Los comandos **Incluir en | Marcador** e **Incluir en | Hipervínculo** se habilitan cuando hay texto o un componente de diseño seleccionado en la vista **Diseño**. Con estos comandos puede crear un marcador o un hipervínculo respectivamente alrededor de la selección. Para más información sobre los marcadores y los hipervínculos, su funcionamiento y cómo trabajar con ellos puede consultar el apartado Características avanzadas | Tabla de contenido, referencias y marcadores 364.

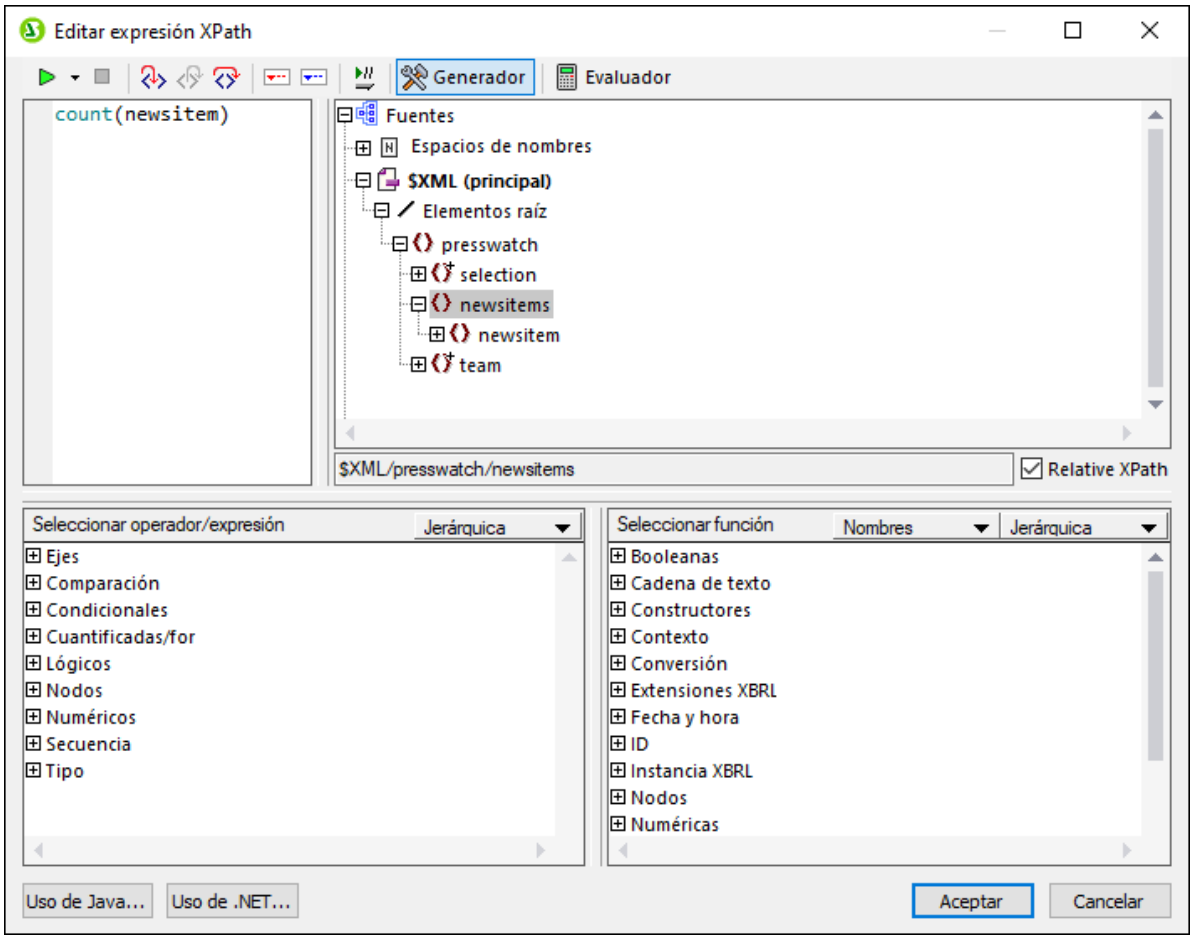
16.9.7 Condición, condición basada en el resultado

El comando **Condición** sirve para insertar una condición en la posición del cursor o alrededor del componente seleccionado. Una condición está compuesta por una o varias ramas y cada rama contiene un conjunto concreto de reglas de procesamiento. De este modo podrá especificar reglas de procesamiento distintas para cada rama. Por ejemplo, una rama puede comprobar si el contenido de un nodo es la cadena `Stop` y, si lo es, marcar el contenido del nodo en color rojo. Otra rama puede comprobar si el contenido del nodo es la cadena `Go` y, si lo es, marcar el contenido del nodo en color verde. La tercera rama puede comprobar si el contenido del nodo no es ni `Stop` ni `Go` y, si así es, marcar el contenido del nodo en color negro.

Siga estos pasos para insertar una condición en el diseño:

1. Ponga el cursor en la posición del diseño elegida para la condición o seleccione el componente que desea convertir en condición.
2. Seleccione el comando de menú **Insertar | Insertar condición** o haga clic con el botón derecho y elija **Insertar condición** en el menú contextual.
3. Se abre el cuadro de diálogo Editar expresión XPath 729 (*imagen siguiente*), donde puede escribir la expresión XPath para la condición.

 © 2016-2022 Altova GmbH



El nodo de contexto para la expresión aparece resaltado en el panel inferior izquierdo, que muestra la estructura del esquema. Puede escribir la expresión XPath directamente en el cuadro de texto *Expresión XPath* o puede hacer doble clic en los elementos de los paneles inferiores para insertarlos en el cuadro de texto. Los nodos de la estructura del esquema del panel izquierdo se insertan en relación al nodo de contexto (si marca la casilla *XPath relativa*) o como expresión absoluta empezando por el nodo de documento (si no marca la casilla *XPath relativa*).

4. Haga clic en **Aceptar** para terminar e insertar la condición, que se crea con una sola rama que comprueba la expresión XPath introducida en el paso anterior.

Insertar condición basada en el resultado

Este comando inserta una condición basada en el resultado en la posición del cursor o alrededor del componente seleccionado. Cada condición de la rama equivale a un formato de salida diferente (vista Authentic, RTF, texto o HTML). Para ver qué tipo de formato de salida representa una rama, pase el puntero del ratón por encima de la etiqueta de la rama o mire qué expresión XPath tiene la rama seleccionada (en la ventana Propiedades, en la entrada rama condición, haga clic en el botón 307 **Edición** 307). Si la condición basada en el resultado se creó en el punto de inserción del cursor, todas las ramas estarán vacías y deberá introducir contenido para cada rama. Si la condición basada en el resultado se creó alrededor de un componente, cada rama contendrá ese componente. Para más información consulte el apartado sobre condiciones basadas en el resultado 307. Estas condiciones se pueden editar, mover y eliminar como si fuera una condición normal y corriente.

Editar las expresiones XPath de las ramas condicionales

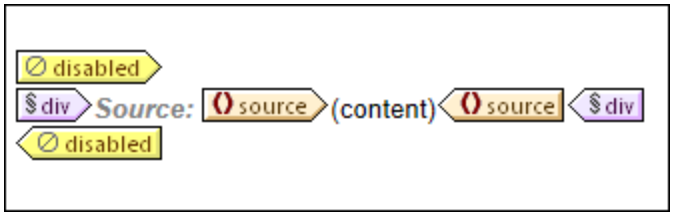
Si quiere editar la expresión XPath de una rama seleccione la rama en la vista **Diseño**. Después, en la ventana Propiedades, seleccione el componente **`rama condicional`** y expanda el grupo de propiedades *cuando*. Haga doble clic en el campo de valor de la propiedad `XPath`. Esto abre el cuadro de diálogo "Editar expresión XPath" (*imagen anterior*), donde puede editar la expresión. Cuando termine haga clic en **Aceptar**.

Agregar ramas, cambiar el orden de las ramas y eliminar ramas

Para agregar ramas nuevas, cambiar el orden de las ramas y eliminar ramas puede hacer clic con el botón derecho en la rama pertinente y seleccionar el comando correspondiente del menú contextual.

16.9.8 Deshabilitado

El comando **Deshabilitado** encierra el contenido seleccionado (incluidos los componentes de diseño) en un componente de tipo deshabilitado (*imagen siguiente*).



El contenido que se ecuentra dentro del componente de tipo deshabilitado (*disabled*) se ignora en el resultado. Esto quiere decir que puede usar los componentes de tipo deshabilitado para omitir en el resultado el contenido que quiere ignorar.

Si quiere restaurar ese contenido solo tiene que eliminar las etiquetas Disabled entre las que se encuentra ese contenido. Para ello, seleccione el componente de tipo deshabilitado, haga clic con el botón derecho y seleccione **Eliminar solo etiqueta**.

16.9.9 Marcadores TDC y niveles TDC

Los componentes del diseño se pueden incluir dentro de componentes de una tabla de contenido (TDC). A continuación enumeramos los comandos del **Incluir en** para crear componentes de TDC alrededor de la selección. Para más información sobre estos comandos haga clic en los enlaces.

- Marcador TDC 344
- Marcador TDC (Asistente) 340
- Nivel de TDC 341
- Referencia de nivel de TDC 349

© 2016-2022 Altova GmbH

Nota: estos comandos también están disponibles en el menú contextual que aparece al hacer clic con el botón derecho en el diseño. Sin embargo, no todos los comandos están siempre disponibles, depende de la posición del diseño donde haga clic con el botón derecho.

16.9.10 Documento nuevo

El comando **Incluir en | Documento nuevo** incluye la selección actual dentro de una plantilla de documento nuevo (*imagen siguiente*).

La plantilla de documento nuevo contiene una sección inicial del documento que ya incluye el texto/componente que estaba seleccionado cuando ejecutó el comando **Incluir en | Documento nuevo**. En la imagen anterior, por ejemplo, se seleccionó una TDC y se incluyó dentro de una plantilla de documento nuevo. Ahora puede insertar contenido en esta sección inicial del documento o anexar secciones de documento a la sección inicial del documento con ayuda del comando **Insertar | Insertar página, columna o sección de documento**.

La plantilla de documento nuevo crea un documento nuevo de salida. Es decir, los resultados estarán compuestos por varios documentos de salida.

16.9.11 Elemento definido por el usuario

El comando **Incluir en | Elemento definido por el usuario** crea un elemento definido por el usuario 151 alrededor del componente/texto seleccionado en la vista **Diseño**. Para más información sobre los elementos definidos por el usuario y cómo usarlos consulte el apartado Archivos SPS: contenido | Elementos definidos por el usuario 151.

16.10 Menú Tabla

El menú **Tabla** ofrece todos los comandos necesarios para insertar una tabla estática o dinámica y para cambiar la estructura y las propiedades de estas tablas. La estructura de las tablas se puede modificar anexando, insertando, eliminando, combinando y dividiendo filas y columnas. Además, las propiedades de las tablas y de sus columnas, celdas y filas se definen con ayuda de estilos CSS 165 y propiedades HTML 165.

Además de en el menú **Tabla**, estos comandos están disponibles como iconos en la barra de herramientas Tabla 751. No obstante, recuerde que los comandos que se habilitan en cada momento dependen de la posición del cursor. Las tablas se pueden insertar en cualquier posición del diseño SPS con el comando **Insertar tabla** 844. Para editar la estructura de la tabla debe poner el cursor en la celda, columna o fila pertinente y seleccionar el comando de edición necesario. Para editar una propiedad de formato debe poner el cursor en la celda, columna o fila pertinente y definir la propiedad necesaria para el componente de tabla en la ventana Estilos 165 o en la ventana Propiedades 165.

Estos son los comandos del menú Tabla:

- **Insertar tabla, Eliminar tabla** 844
- **Agregar columna/fila de encabezado/pie de tabla** 845
- **Anexar/Insertar fila/columna** 845
- **Eliminar fila, Eliminar columna** 846
- **Combinar celda hacia la izquierda/derecha/arriba/abajo** 846
- **Dividir la celda horizontalmente/verticalmente** 846
- **Ver bordes de la celda, Ver marcado de la tabla** 847
- **Propiedades de la tabla** 847
- **Alineación vertical del contenido de la celda** 848

Encabezados y pies de tabla

En las tablas dinámicas puede indicar si quiere incluir encabezados o pies. Los pies de tabla solamente están permitidos cuando la tabla aumenta de arriba a abajo. En las tablas estáticas puede crear un encabezado o pie de tabla insertando a mano una fila al principio y otra al final. La estructura de los encabezados y pies de tabla tanto en las tablas estáticas como en las dinámicas se puede modificar con solo dividir y combinar celdas.

Navegar por las tablas
Puede recorrer las celdas de la tabla con ayuda de las teclas de dirección y con el tabulador.

Agregar contenido a las celdas
En las celdas puede insertar cualquier tipo de componente de diseño. No obstante, debe dar formato al componente usando las herramientas de formato estándar.

16.10.1 Insertar tabla, eliminar tabla

El comando **Insertar tabla** inserta una tabla vacía en el diseño. Al hacer clic en este comando se abre un cuadro de diálogo donde puede elegir entre crear una tabla estática o dinámica.

© 2016-2022 Altova GmbH

- Si elige la opción **Tabla estática**, aparece otro cuadro de diálogo donde puede definir el tamaño de la tabla (el número de filas y columnas).
- Si elige la opción **Tabla dinámica**, aparece el cuadro de diálogo "Selector de XPath" donde puede definir a partir de qué nodo se crea la tabla dinámica. Al hacer clic en **Aceptar** en este diálogo se abre el diálogo "Crear tabla dinámica" donde puede seleccionar qué nodos secundarios se usan para las filas/columnas de la tabla. Para más información consulte el apartado Crear tablas dinámicas 158.

Una vez creada la tabla, puede cambiar su estructura anexando, insertando y eliminando filas y columnas.

El comando **Eliminar tabla** elimina la tabla estática o dinámica donde está el cursor.

16.10.2 Agregar encabezado y pie de tabla

Los encabezados de tabla pueden ser una fila de encabezado (situada encima del cuerpo de la tabla) o una columna de encabezado (a la izquierda del cuerpo de la tabla, aunque parece que está dentro del cuerpo de la tabla). Igualmente, los pies de tabla pueden ser una fila (situada debajo del cuerpo de la tabla) o una columna (situada a la derecha del cuerpo de la tabla, aunque parece que está dentro del cuerpo de la tabla).

Nota: en los documentos de salida HTML los encabezados de tabla están dentro de elementos `th` y aparecen en negrita porque ese es el formato inherente de los elementos `th`).

Estos son los comandos del menú **Tabla** con los que puede agregar encabezados y pies de tabla:

	Agregar columna de encabezado de tabla	Añade una columna de encabezado a la izquierda del cuerpo de la tabla.
	Agregar columna de pie de tabla	Añade una columna de pie a la derecha del cuerpo de la tabla.
	Agregar fila de encabezado de tabla	Añade una fila de encabezado encima del cuerpo de la tabla.
	Agregar fila de pie de tabla	Añade una fila de pie de tabla debajo del cuerpo de la tabla.

16.10.3 Anexar/insertar fila/columna

El comando **Anexar fila** anexa una fila a la tabla estática o dinámica donde está el cursor.

El comando **Insertar fila** inserta una fila encima de la fila donde está el cursor. Este comando se puede usar en tablas estáticas y dinámicas.

El comando **Anexar columna** anexa una columna a la tabla estática o dinámica donde está el cursor.

El comando **Insertar columna** inserta una columna a la izquierda de la columna donde está el cursor. Este comando se puede usar en tablas estáticas y dinámicas.

16.10.4 Eliminar fila/columna

El comando **Eliminar fila** elimina la fila donde está el cursor. Este comando se puede usar en tablas estáticas y dinámicas.

El comando **Eliminar columna** elimina la columna donde está el cursor. Este comando se puede usar en tablas estáticas y dinámicas.

16.10.5 Combinar celda hacia la izquierda, derecha, arriba, abajo

El comando **Combinar hacia la izquierda** une la celda donde está el cursor con la celda situada inmediatamente a su izquierda. El contenido de las celdas combinadas se concatena en la celda nueva. Todos los valores de las propiedades de la celda situada a la izquierda se pasan a la celda nueva. Este comando se puede usar en tablas estáticas y dinámicas.

El comando **Combinar hacia la derecha** une la celda donde está el cursor con la celda situada inmediatamente a su derecha. El contenido de las celdas combinadas se concatena en la celda nueva. Todos los valores de las propiedades de la celda situada a la izquierda se pasan a la celda nueva. Este comando se puede usar en tablas estáticas y dinámicas.

El comando **Combinar hacia abajo** une la celda donde está el cursor con la celda situada inmediatamente debajo. El contenido de las celdas combinadas se concatena en la celda nueva. Todos los valores de las propiedades de la celda situada arriba se pasan a la celda nueva. Este comando se puede usar en tablas estáticas y dinámicas.

El comando **Combinar hacia arriba** une la celda donde está el cursor con la celda situada inmediatamente arriba. El contenido de las celdas combinadas se concatena en la celda nueva. Todos los valores de las propiedades de la celda situada arriba se pasan a la celda nueva. Este comando se puede usar en tablas estáticas y dinámicas.

16.10.6 Dividir la celda horizontal y verticalmente

El comando **Dividir la celda horizontalmente** crea una celda nueva a la derecha de la celda donde está el cursor. El contenido de la celda original se queda en la celda de la izquierda. Todas las propiedades de la celda original se pasan a la celda nueva. Este comando se puede usar en tablas estáticas y dinámicas.

© 2016-2022 Altova GmbH

El comando **Dividir la celda verticalmente** crea una celda nueva debajo de la celda donde está el cursor. El contenido de la celda original se queda en la celda de arriba. Todas las propiedades de la celda original se pasan a la celda nueva. Este comando se puede usar en tablas estáticas y dinámicas.

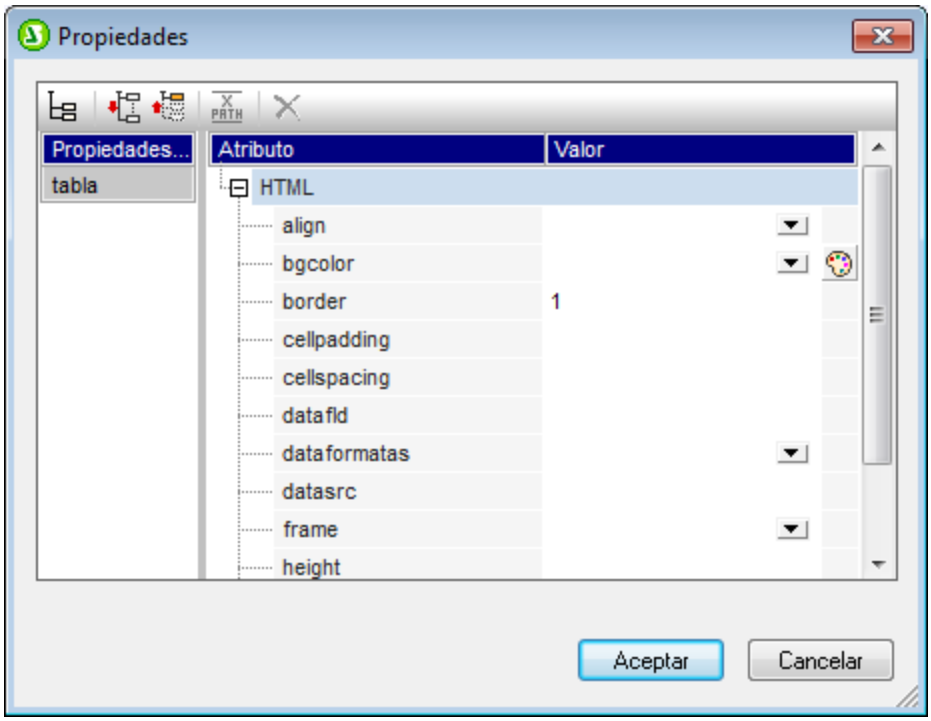
16.10.7 Ver bordes de la celda, Ver marcado de la tabla

Los comandos **Ver bordes de la celda** y **Ver marcado de la tabla** sirven para ver el contorno de las celdas y el marcado de las columnas y filas de la tabla. Si activa estos dos comandos, podrá comprender mucho mejor la estructura de las tablas. Si las desactiva, podrá ver el contenido de la tabla con mayor precisión.

	Ver bordes de la celda	Muestra/oculta el contorno de la tabla (los bordes) si la tabla tiene una propiedad `border` con valor `0` (grupo de propiedades *HTML*).
	Ver marcado de la tabla	Muestra/oculta los marcadores de columna y fila azules.

16.10.8 Propiedades de la tabla

El comando **Propiedades de la tabla** se habilita cuando el cursor está dentro de una tabla estática o dinámica 155. Al hacer clic en este comando se abre la ventana Propiedades, donde aparecen las propiedades del componente **`tabla`** (*imagen siguiente*).

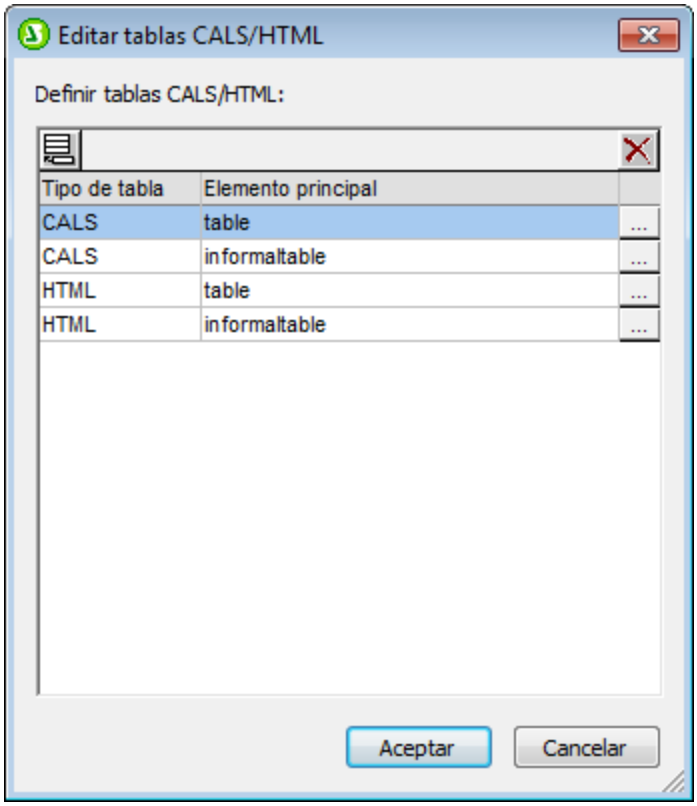


En esta ventana puede editar las propiedades de la tabla. Cuando termine haga clic en **Aceptar**.

16.10.9 Editar tablas CALS/HTML

A veces, el documento XML contiene estructuras de datos que siguen el modelo de tabla CALS o el modelo de tablas HTML. Con el comando **Editar tablas CALS/HTML** puede crear tablas en el documento de salida a partir de este tipo de estructuras en forma de tabla en los documentos de salida. El marcado de la tabla en los formatos de salida se deriva directamente del documento XML. Sin embargo, puede añadir más estilos a la tabla desde el diseño SPS.

Al hacer clic en el comando **Editar tablas CALS/HTML** aparece el cuadro de diálogo "Editar tablas CALS/HTML" (*imagen siguiente*).



Consulte la sección Tablas 155 para aprender a trabajar con tablas CALS/HTML.

16.10.10 Alineación vertical del contenido de la celda

La barra de herramientas **Tabla** ofrece varios comandos para configurar la alineación vertical del contenido de las celdas de la tabla. Primero debe poner el cursor dentro de la celda pertinente y después hacer clic en los iconos.

Alinear arriba: este comando alinea el contenido de la celda en la parte superior de la celda.

© 2016-2022 Altova GmbH

Centrar verticalmente: este comando alinea el contenido en el centro de la celda.

Alinear abajo: este comando alinea el contenido de la celda en la parte inferior de la celda.

16.11 Menú Authentic

Los comandos del menú **Authentic** sirven para:

- Personalizar algunos aspectos de la vista Authentic de un documento XML, generada a partir del diseño SPS creado en StyleVision.
- Editar documentos en la vista Authentic de StyleVision.

Estos son los comandos del menú **Authentic**:

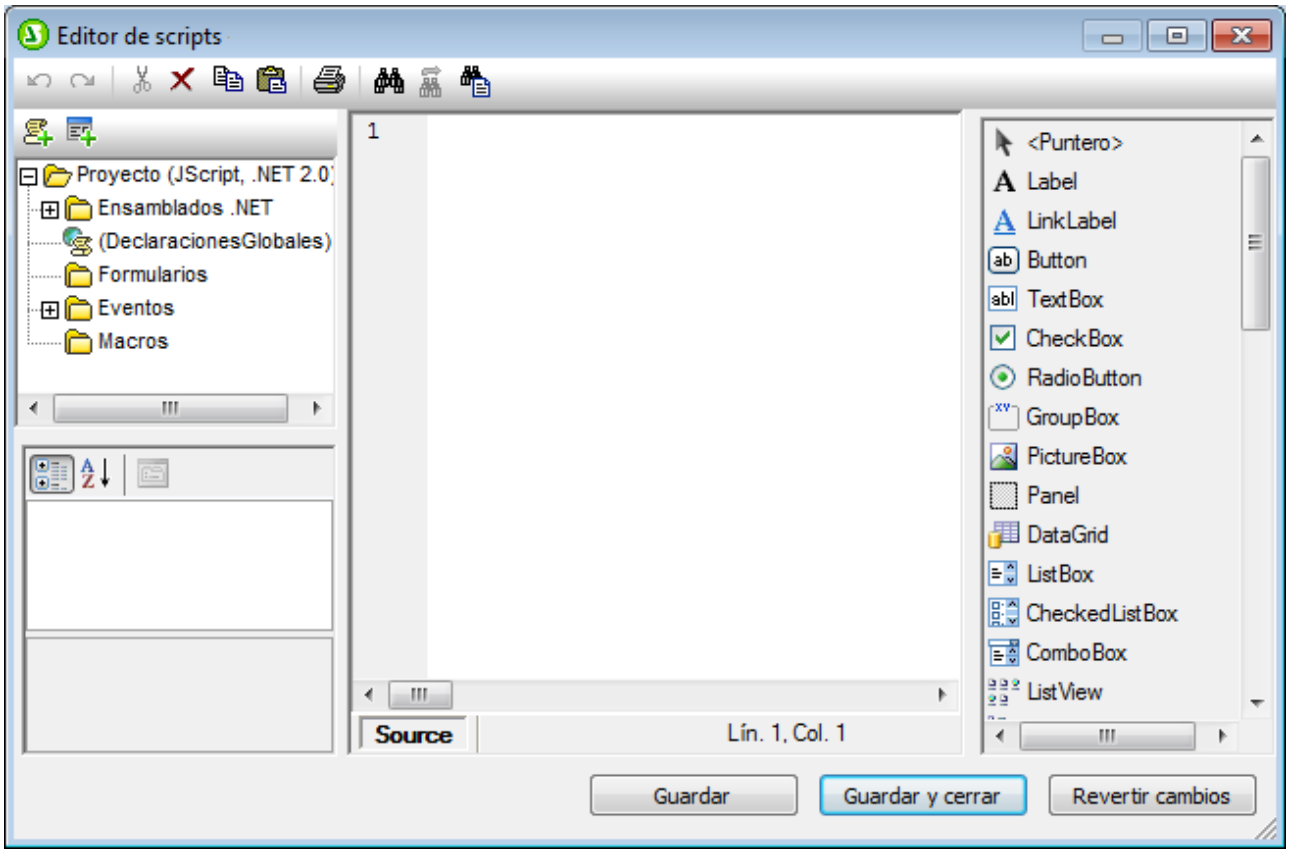
- **Editar scripts de Authentic** 850
- **Personalizar botones de la barra de herramientas** 851
- **Revisar referencias a macros** 854
- **Agregar selector de fecha automáticamente** 855
- **Agregar controles de base de datos automáticamente** 855
- **Volver a cargar la vista Authentic** 856
- **Validar el documento XML** 856
- **Seleccionar y editar una fila nueva con datos XML** 856
- **Definir entidades XML** 857
- **Ver marcado** 858
- **Comandos para operaciones con filas de tablas dinámicas** 859

Todos estos comandos se describen detalladamente en los apartados de esta sección.

16.11.1 Editar scripts de Authentic

El comando **Editar scripts de Authentic** abre el Editor de scripts de StyleVision (*imagen siguiente*), donde puede crear formularios, eventos y macros para usarlos en la vista Authentic.

© 2016-2022 Altova GmbH



16.11.2 Personalizar botones de la barra de herramientas

Al hacer clic en el comando **Personalizar botones de la barra de herramientas** se abre el cuadro de diálogo "Botones personalizados de Authentic" (*imagen siguiente*), donde puede diseñar una barra de herramientas para Authentic. Después de guardarla con el diseño SPS, la barra de herramientas aparecerá en la vista Authentic de las ediciones Enterprise y Professional de los productos de Altova en los que se abra un archivo XML asociado a este SPS.

Nota: estos productos de Altova incluyen la vista Authentic: XMLSpy Enterprise, XMLSpy Professional, StyleVision Enterprise, StyleVision Professional y Authentic Desktop Enterprise y el complemento de Authentic para exploradores Enterprise Edition.

Agregar un botón

Para agregar un botón o separador a la barra de herramientas de Authentic, haga clic en el botón **Agregar** situado en la esquina superior derecha del diálogo (*imagen anterior*). Al hacer clic en ese botón aparece un menú donde puede seleccionar qué tipo de elemento desea añadir a la barra de herramientas: (i) un **icono de estado del texto**, (ii) un **botón personalizado** o (iii) un **separador**. Más abajo explicamos qué son los iconos de estado del texto 852 y los botones personalizados 853. Los separadores sirven para separar los iconos y botones de la barra de herramientas.

Mover y eliminar botones

Para mover un botón o separador a otra posición, selecciónelo, arrástrelo y suéltelo en su nuevas posición. Para eliminar un botón o un separador, selecciónelo y haga clic en el botón **Eliminar** situado en la esquina superior derecha del diálogo (*imagen anterior*).

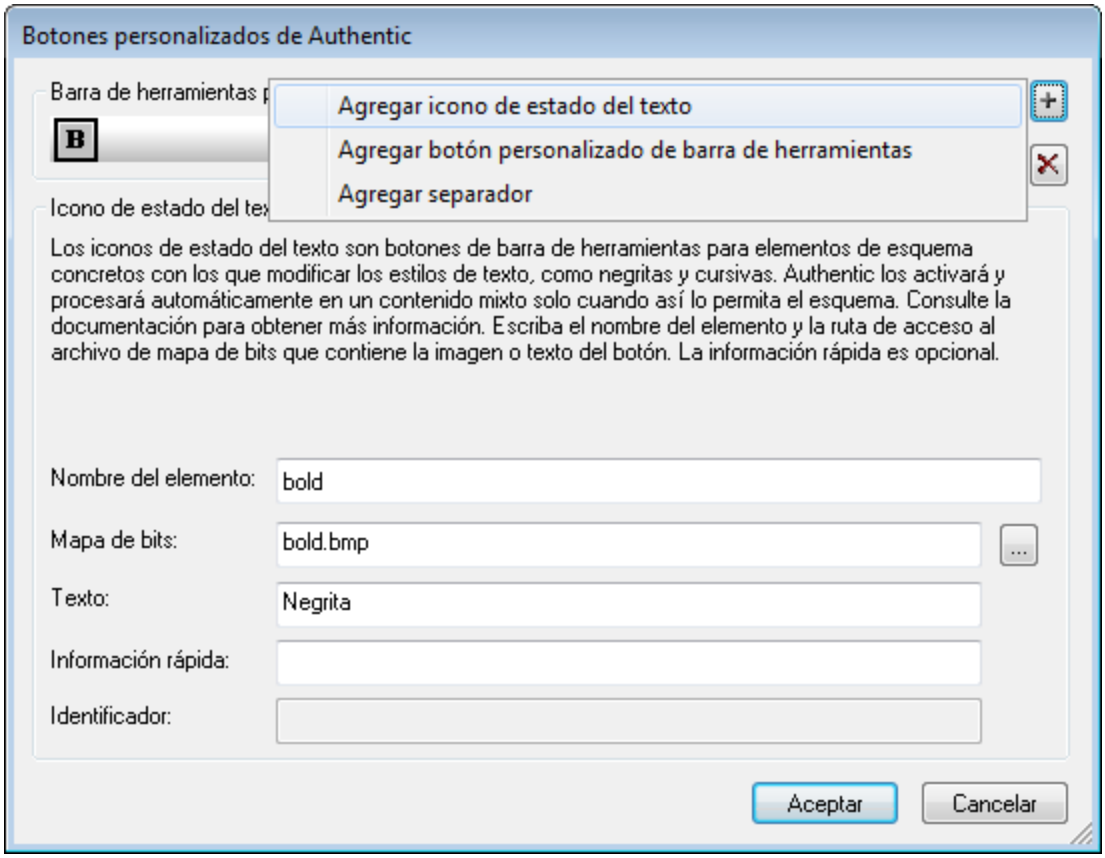
Iconos de estado del texto

Un icono de estado del texto es un icono para un elemento global del esquema. Cuando el usuario de la vista Authentic selecciona texto en el documento de la vista Authentic y hace clic en un icono de estado del texto, el elemento global elegido se inserta alrededor del texto seleccionado. Estos iconos están pensados para dar formato al texto.

Para añadir un icono de estado del texto en la barra de herramientas Authentic, haga clic en el botón **Agregar** situado en la esquina superior derecha del diálogo "Botones personalizados de Authentic" (*imagen siguiente*) y después seleccione **Agregar icono de estado del texto**. Después escriba el nombre del elemento XML para

 © 2016-2022 Altova GmbH

el icono y busque una imagen para el botón o escriba un texto simplemente. Si quiere, también puede insertar información rápida, que aparecerá cuando el usuario de la vista Authentic pase el puntero del ratón encima del icono. Haga clic en **Aceptar** para agregar el botón a la barra de herramientas.



La imagen anterior, por ejemplo, muestra el icono de estado del texto para el elemento `bold`. Este icono usa el archivo de imagen `bold.bmp`. El texto del botón se introdujo como reserva. Es decir, si no se encuentra la imagen indicada, en el botón aparecerá el texto `Negrita`.

Botones personalizados

Cuando el usuario de la vista Authentic hace clic en un botón personalizado de la barra de herramientas de Authentic se ejecuta una macro. Es decir, puede crear un botón personalizado en el diseño SPS y especificar qué macro dispara el botón.

Para añadir un botón personalizado a la barra de herramientas Authentic, haga clic en el botón **Agregar** situado en la esquina superior derecha del cuadro de diálogo "Botones personalizados de Authentic" (*imagen siguiente*). Después seleccione la opción **Agregar botón personalizado de barra de herramientas**.

Los botones personalizados toman estos parámetros:

- La ubicación de la imagen para el botón (campo *Mapa de bits*) o el texto para el botón (campo *Texto*).
- En el cuadro combinado *Nombre de la macro* debe seleccionar una macro de la lista desplegable. Las macros de la lista son las que están guardadas con el SPS. Haga clic en el botón **Crear** para abrir el Editor de scripts 907 de StyleVision y crear una macro para el SPS. Haga clic en el botón **Editar** para abrir el Editor de scripts 907 y editar la macro seleccionada.
- También puede insertar un texto de *Información rápida* que ayude al usuario de la vista Authentic. Este texto aparece cuando el usuario pasa el puntero del ratón encima del botón.
- En el campo *Identificador* escriba una cadena de texto que sirva de identificador para el botón. Este identificador se puede usar en el código del script.

En el ejemplo de la imagen anterior se creó un botón personalizado para la macro `ChangeAddressToUS`. Este botón usa una imagen llamada `US.bmp`. Además se creó un texto de información rápida y se definió el identificador `Dirección_US`. Estos ejemplos están en el archivo `ToolbarButtons.sps` de la carpeta `Authentic\Scripting` del proyecto `Examples` que viene con StyleVision.

16.11.3 Revisar referencias a macros

El comando **Revisar referencias a macros** comprueba que las referencias a macros de los botones de la barra de herramientas y de los scripts del SPS son correctas. Si se detectan referencias incorrectas, se emite un mensaje de error.

 © 2016-2022 Altova GmbH

16.11.4 Agregar selector de fecha automáticamente

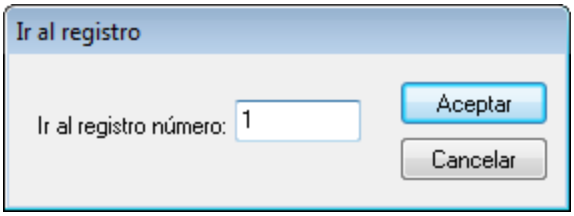
El comando **Agregar selector de fecha automáticamente** es un conmutador que activa/desactiva los selectores de fecha automáticos. Si activa esta característica e inserta el marcador de posición `(contenido)` o un campo de entrada creados con un elemento de tipo `xs:date` o `xs:dateTime`, entonces se crea automáticamente un selector de fecha entre las etiquetas del elemento y después del marcador de posición `(contenido)` o del campo de entrada.

16.11.5 Agregar controles de base de datos automáticamente

El comando **Agregar controles de base de datos automáticamente** es un conmutador que activa/desactiva los controles de BD automáticos. Si activa esta característica e inserta un elemento de tabla de BD en el diseño, el panel de control de BD (*imagen siguiente*) se inserta inmediatamente antes del elemento secundario `Fila` del elemento de tabla de BD que acaba de insertar.

Con el panel de control de BD el usuario de la vista Authentic puede recorrer las filas de la tabla de la BD en la vista Authentic:

- El primer botón del panel le lleva al primer registro de la tabla.
- El segundo le lleva al registro anterior al actual.
- El tercer botón abre el cuadro de diálogo "Ir al registro" (*imagen siguiente*) donde puede insertar el número del registro al que desea ir.
- El cuarto botón le lleva al registro siguiente al actual.
- El quinto y último botón le lleva al último registro de la tabla.



Si desactiva esta característica, el panel de control de BD no se inserta al crear una tabla de BD en el documento de diseño.

Nota: estos botones de navegación se pueden insertar a mano. Primero debe poner el cursor en cualquier posición entre la etiqueta inicial y final de la tabla de BD. Después haga clic en **Insertar | Insertar control de base de datos** y seleccione un control del submenú que aparece. El submenú ofrece tres opciones: insertar (i) un panel con cuatro botones de navegación (**Navegación**), (ii) un panel con cuatro botones de navegación y el botón **Ir a** (**Navegación+Ir a**) y (iii) un botón de consulta.

16.11.6 Volver a cargar la vista Authentic, Validar el documento XML

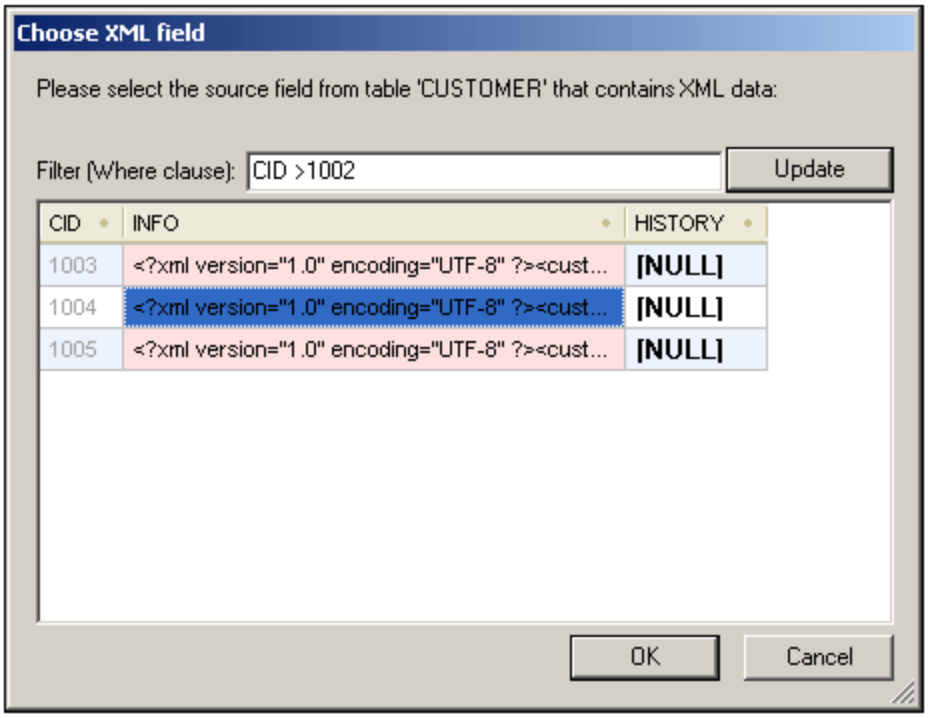
Con el comando **Volver a cargar** puede volver a cargar el archivo de datos XML de la vista Authentic. Esto puede ser necesario si el archivo se modifica fuera de StyleVision, sobre todo si hay varios usuarios trabajando en el mismo archivo.

El comando **Validar el documento XML (F8)** comprueba si el archivo XML es válido con respecto al esquema asociado. El validador XSD que se debe utilizar (XSD 1.0 o XSD 1.1) se define en el cuadro de diálogo Propiedades 787). También se comprueba la validez de los nodos en los que añadieron requisitos de validación adicional (en la ventana Propiedades, expanda el grupo de propiedades *Authentic* y edite la propiedad `validación adicional`). El resultado de la validación aparece en un mensaje.

16.11.7 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.

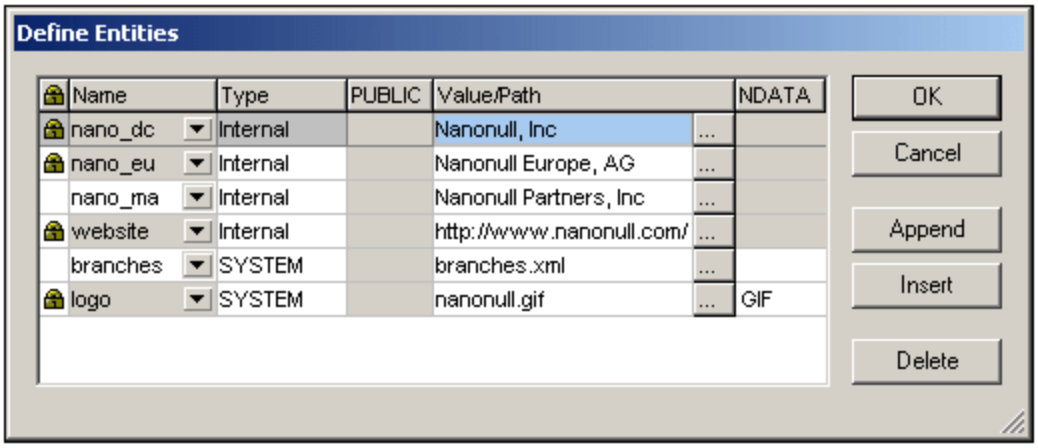
16.11.8 Definir entidades XML

El comando **Definir entidades XML** solamente está disponible en la vista Authentic. Con este comando puede definir qué entidades desea añadir al **documento XML**. Una vez definida, la entidad se puede insertar en el documento XML. Para ello haga clic con el botón derecho donde desea insertar la entidad y, en el menú contextual, seleccione **Insertar entidad** y el nombre de la entidad pertinente.

Con este comando puede definir tres tipos de entidades:

- *Entidades analizadas internas*: el valor de la entidad es una cadena de texto que aparece frecuentemente en el documento. El uso de una entidad garantiza que todas las apariciones usen el mismo valor definido aquí.
- *Entidades analizadas externas*: se trata de un archivo XML externo que reemplazará a la entidad cada vez que aparezca. El valor de la entidad es el URI del archivo XML externo.
- *Entidades externas sin analizar*: se trata de un recurso externo al que se llamará cuando se procese la entidad. El valor de la entidad es el URI del recurso externo.

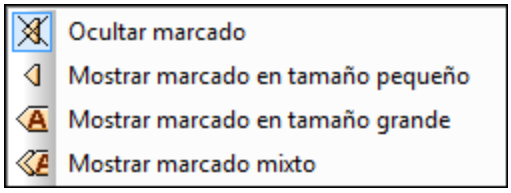
Al hacer clic en el comando se abre el cuadro de diálogo "Definir entidades" (*imagen siguiente*).



Para aprender a usar este cuadro de diálogo consulte el apartado Definir entidades 683 de la documentación de la vista Authentic.

16.11.9 Ver marcado

El comando **Ver marcado** tiene un submenú con varios comandos (*imagen siguiente*) que permiten controlar el marcado del documento XML en la vista Authentic, es decir las etiquetas de nodo que aparecen en la vista. A continuación describimos estos comandos.



El comando **Ocultar marcado** oculta los símbolos de marcado del documento en la vista Authentic.

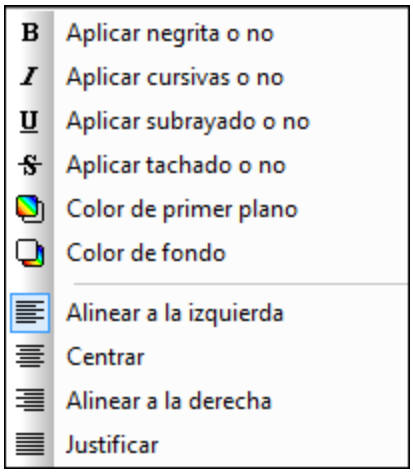
El comando **Mostrar marcado en tamaño pequeño** muestra símbolos pequeños de marcado en la vista Authentic.

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.

16.11.10 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.

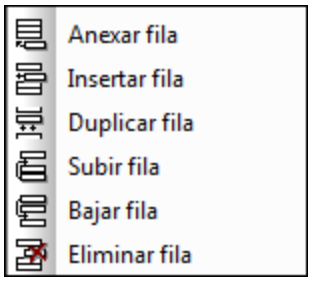


 © 2016-2022 Altova GmbH

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.

16.11.11 Comandos de fila de tabla dinámica

Los comandos **Anexar fila**, **Insertar fila**, **Duplicar fila**, **Subir fila**, **Bajar fila** y **Eliminar fila** del menú **Authentic** se habilitan cuando el usuario de la vista Authentic pone el cursor dentro de una fila de una tabla dinámica. Con estos comandos el usuario puede manipular las filas de la tabla. Puesto que una fila de una tabla dinámica representa una estructura de datos fija, el usuario de la vista Authentic en realidad está manipulando unidades de una estructura de datos en el contexto de la estructura de datos representada por la tabla dinámica.



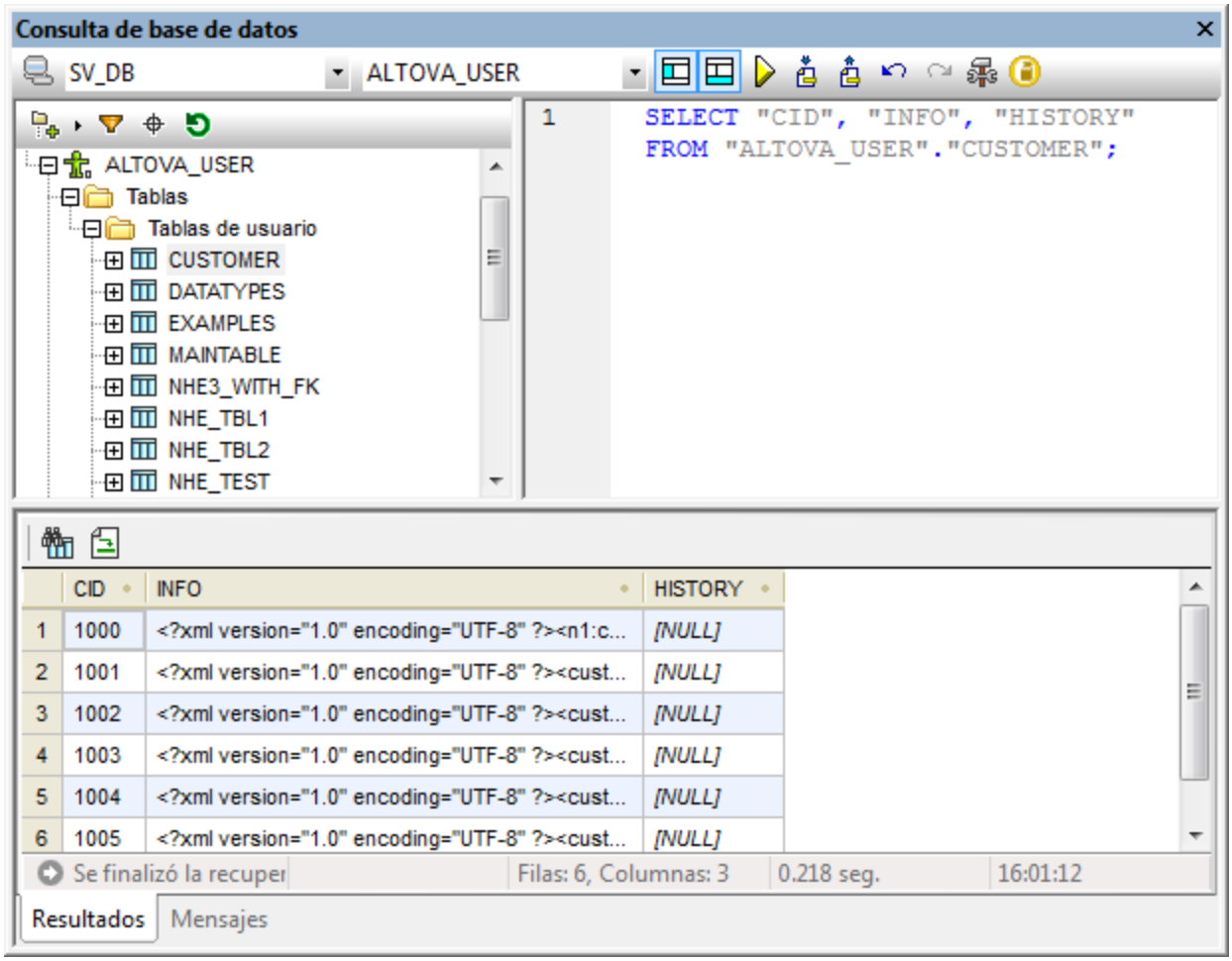
16.12 Menú Base de datos

Los comandos del menú **Base de datos** sirven para consultar la BD que está conectada y editar y borrar los filtros de BD que se aplicaron a la BD.

- El comando **Consultar la base de datos** 860 inicia el proceso de conexión con la BD 523 y abre la ventana "Consulta de base de datos".
- Los comandos **Editar filtro de la BD** 861 y **Borrar filtro de la base de datos** 861 abren sendos cuadros de diálogo donde puede editar y borrar los filtros de la base de datos.

16.12.1 Consultar la base de datos

El comando **Consultar la base de datos** abre la ventana Consulta de base de datos 633 (*imagen siguiente*). Desde esta ventana puede conectarse a una BD y consultarla. Para más información sobre cómo usar esta ventana consulte el apartado Consultar la base de datos 633 del manual.



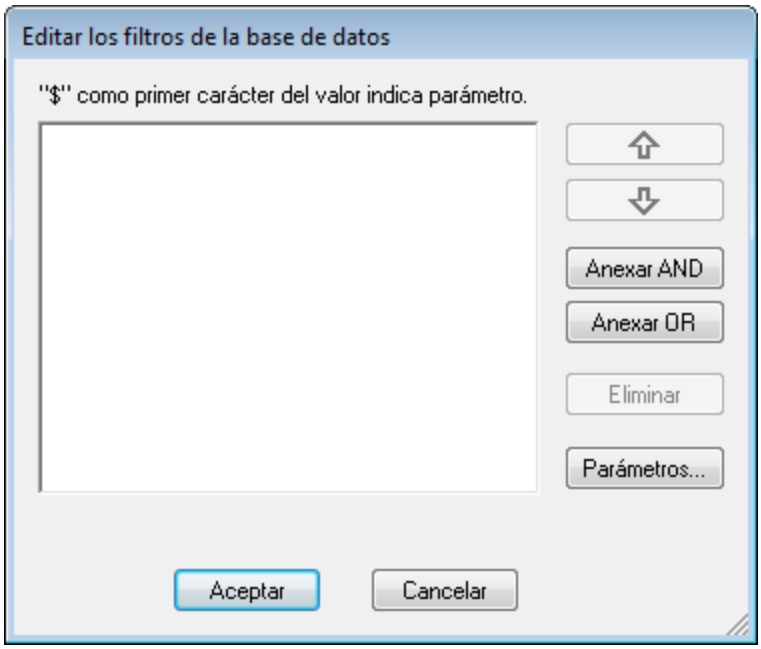
Esta ventana puede abrirse y cerrarse haciendo clic en el comando **Consultar la base de datos**.

16.12.2 Editar/borrar filtro de la base de datos

El comando **Editar filtro de la base de datos** sirve para crear y editar filtros para una tabla de la BD (un filtro de BD). El filtro determina qué datos de la tabla de la BD seleccionada se importan y presentan en el diseño. Los filtros están compuestos por uno o varios criterios, que se especifican mediante expresiones. Cada expresión está compuesta por operadores (= o >) y valores (texto o números). Además, los criterios se pueden unir con los operadores lógicos `AND` y `OR`.

Siga estas instrucciones para crear o editar un filtro de BD:

1. Seleccione el elemento de tabla de datos de nivel superior para el que desea crear o editar el filtro de BD. Es decir, haga clic en la etiqueta del elemento en la vista **Diseño** o en el nombre del elemento en la ventana Estructura del esquema.
2. Seleccione el comando **Base de datos | Editar filtro de la base de datos** o haga clic en el icono correspondiente de la barra de herramientas. Esto abre el cuadro de diálogo "Editar los filtros de la base de datos" (*imagen siguiente*).



3. Use los botones **Anexar AND** y **Anexar OR** para añadir criterios al filtro. Para subir o bajar un criterio use los botones que tienen forma de flecha. Para eliminar un criterio haga clic en el botón **Eliminar**.
4. Especifique los criterios para el filtro de BD. Cada criterio tiene tres partes: `Nombre del campo + Operador + Valor`. Las opciones para los campos `Nombre del campo` y `Operador` están disponibles en cuadros combinados. El valor de la expresión debe insertarse a mano y puede ser un parámetro (que se indica con el prefijo `$`).

Borrar el filtro de la base de datos

El comando **Borrar el filtro de la base de datos** elimina el filtro pero antes pregunta si está seguro de que desea borrarlo.

© 2016-2022 Altova GmbH

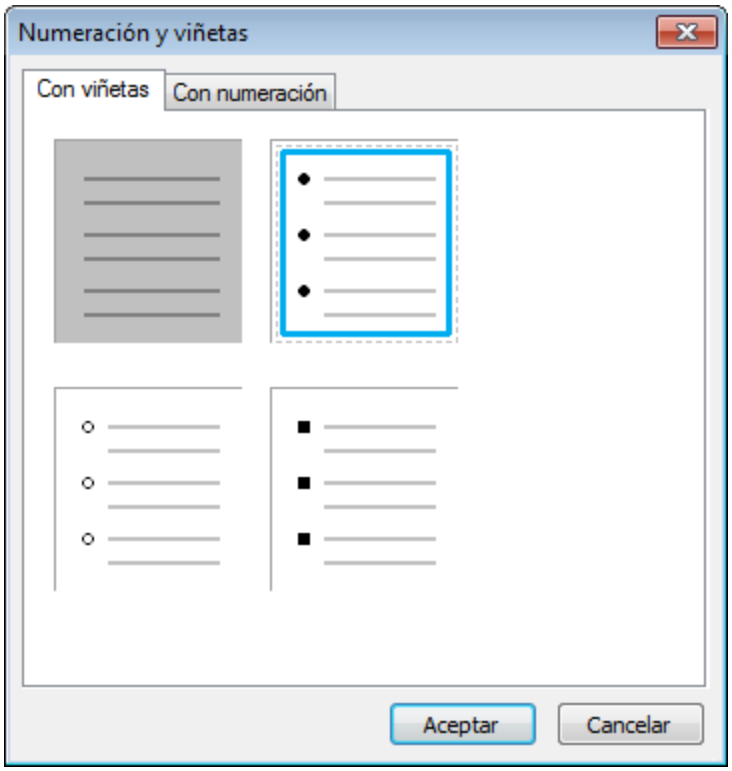
16.13 Menú Propiedades

Los comandos del menú **Propiedades** sirven para insertar listas y definir formatos de tipo de datos para la función de formato de entrada 380. Los apartados siguientes describen en detalle cada comando:

- **Editar numeración y viñetas** 863
- **Cadenas de formato de valor predefinidas** 863: define los formatos de tipo de datos numérico para el SPS.

16.13.1 Editar numeración y viñetas

El comando **Editar numeración y viñetas** sirve para insertar una lista en la posición del cursor. Al hacer clic en este comando aparece el cuadro de diálogo "Numeración y viñetas" (*imagen siguiente*), donde puede seleccionar el estilo de la lista. Si elige crear una lista numerada, también puede elegir el número por el que debe empezar la lista.



16.13.2 Cadenas de formato de valor predefinidas

Todos los marcadores de posición `(contenido)`, campos de entrada y cálculos automáticos que sean de tipo de datos `numeric`, `date`, `time`, `dateTime` o `duration` pueden tener asignado un formato personalizado.

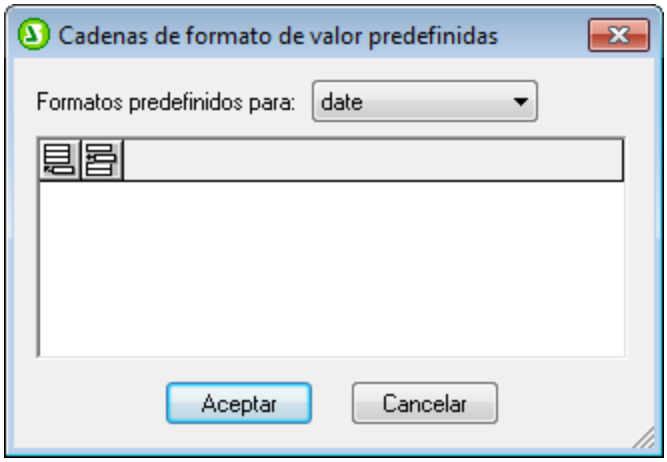
Esto se hace en el cuadro de diálogo Formato de valor 380, donde puede crear un formato directamente o seleccionar un formato de una lista de formatos predefinidos.

En el cuadro de diálogo "Formato de valor", la lista desplegable de formatos predefinidos incluye formatos de dos tipos:

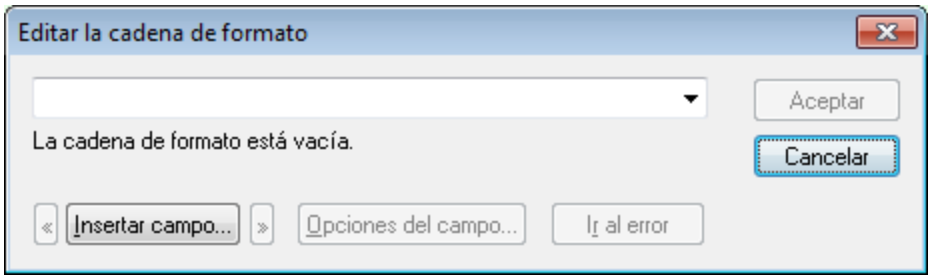
- Formatos predefinidos que vienen con StyleVision
- Formatos definidos por el usuario con el comando **Cadenas de formato de valor predefinidas**. Estos formatos de valor se crean para el diseño SPS activo solamente y no para toda la aplicación. Tras crear los formatos de valor predefinidos, debe guardar el diseño SPS para que los formatos estén disponibles cuando vuelva a abrir el archivo.

Crear una cadena de formato de valor predefinida
Las cadenas de formato de valor predefinidas son específicas del tipo de datos. Para crear una cadena de formato de valor predefinida:

1. Haga clic en el comando **Propiedades | Cadenas de formato de valor predefinidas**. Se abre este cuadro de diálogo:

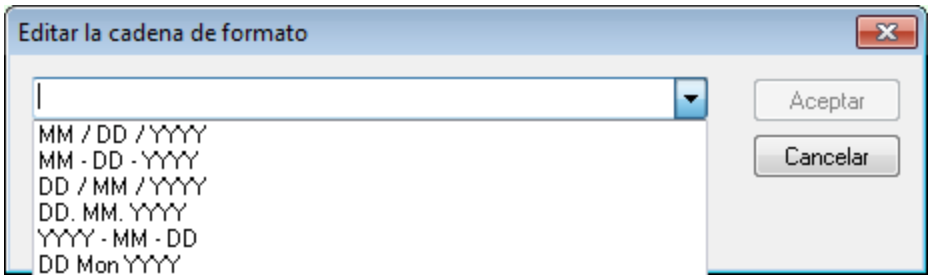


2. Seleccione un tipo de datos de la lista desplegable del cuadro combinado y haga clic en el icono **Anexar** o **Insertar** según corresponda. Esto abre el cuadro de diálogo "Editar la cadena de formato":



Abra la lista desplegable del cuadro combinado de este diálogo, que contiene los formatos predefinidos que vienen con StyleVision para el tipo de datos elegido en el paso anterior (*imagen siguiente*).

 © 2016-2022 Altova GmbH



Ahora puede seleccionar un formato de la lista y modificarlo o introducir un formato directamente en el campo de entrada. La sintaxis para definir un formato se explica en la sección Formato de valor 380, pero también puede obtener ayuda sobre la sintaxis haciendo clic en los botones **Insertar campo** y **Opciones de campo**.

3. Tras definir el formato haga clic en **Aceptar** y guarde el archivo SPS. La cadena de formato se añade a la lista de formatos predefinidos del tipo de datos y en adelante aparecerá como opción en el cuadro de diálogo "Formato de valor" (del archivo SPS activo) cuando el elemento seleccionado tenga ese tipo de datos.

Notas:

- Por cada tipo de datos puede añadir tantas cadenas de formato personalizadas como quiera.
- El orden de las cadenas de formato en el cuadro de diálogo "Cadenas de formato predefinidas" determina el orden en que esta cadenas aparecen en el cuadro de diálogo "Formato de valor". Las cadenas de formato personalizadas aparecen antes que los formatos predefinidos que vienen con StyleVision.
- Para editar una cadena de formato haga doble clic en su entrada en el diálogo "Cadenas de formato predefinidas".
- Para eliminar una cadena de formato selecciónela y haga clic en el icono **Eliminar** del diálogo "Cadenas de formato predefinidas".

16.14 Menú Herramientas

El menú **Herramientas** contiene comandos para la revisión ortográfica y otros comandos con los que podrá personalizar StyleVision.

Los apartados siguientes describen todos los comandos del menú **Herramientas**:

- Ortografía 866
- Opciones de ortografía 867
- Recursos globales 870
- Configuración activa 871
- Gestor de esquemas XML 232
- Personalizar 887
- Opciones 894

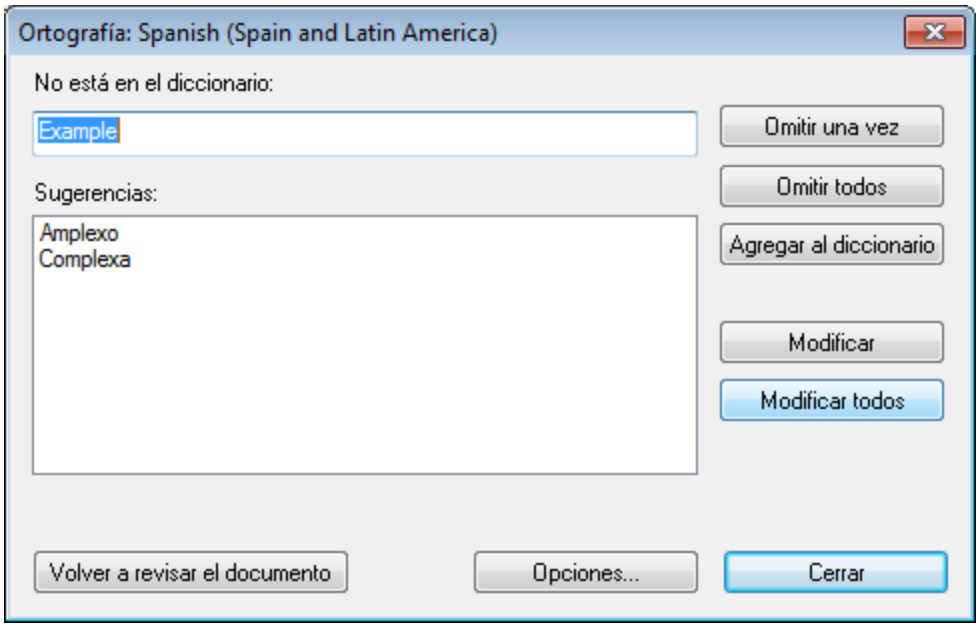
16.14.1 Ortografía

El comando **Herramientas | Ortografía** ejecuta una revisión ortográfica del diseño SPS (vista **Diseño**) o el documento de la vista Authentic, dependiendo de cuál de las dos vistas esté activa. La aplicación incluye una opción para elegir el idioma de la revisión ortográfica.

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. 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.

Al hacer clic en este comando aparece el cuadro de diálogo "Ortografía". Las palabras que no estén en el diccionario elegido aparecen una por una, por orden de documento, en el cuadro *No está en el diccionario* del cuadro de diálogo y se resaltan en la vista.

© 2016-2022 Altova GmbH

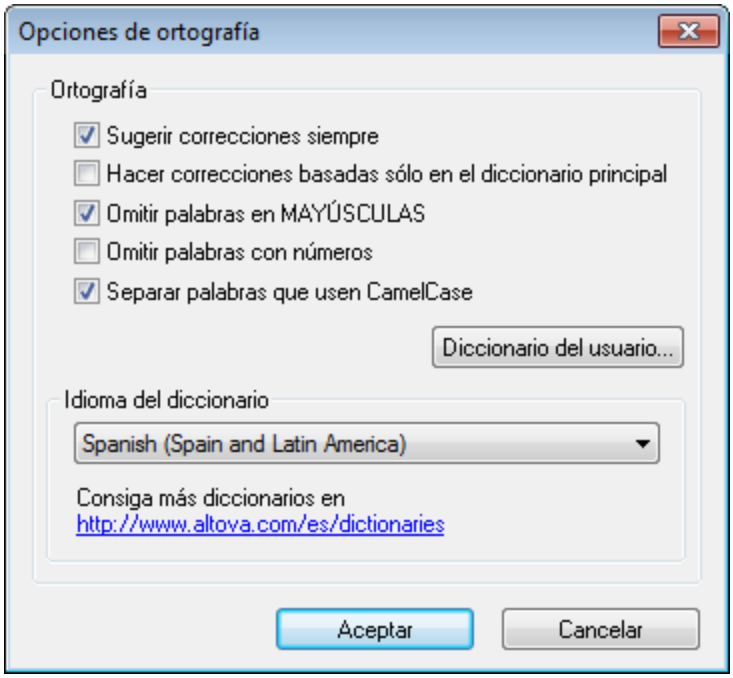


Puede seleccionar una entrada de la lista del panel *Sugerencias* y después hacer clic en **Modificar** o **Modificar todos** para cambiar la instancia resaltada de la palabra o todas sus instancias respectivamente. Haga doble clic en una palabra del panel *Sugerencias* para reemplazar la palabra que no está en el diccionario. Si lo prefiere, puede omitir *esta instancia* de la palabra desconocida (**Omitir una vez**) o puede omitir *todas las instancias* de esta palabra (**Omitir todos**). También puede agregar la palabra desconocida al diccionario del usuario (**Agregar al diccionario**) para que el corrector ortográfico considere que la palabra es correcta y pase a la siguiente palabra desconocida. Por último puede volver a revisar el documento desde el principio (**Volver a revisar el documento**) o cerrar el cuadro de diálogo (**Cerrar**).

El botón **Opciones** abre el cuadro de diálogo Opciones de ortografía 867, donde puede configurar el corrector ortográfico.

16.14.2 Opciones de ortografía

El comando **Herramientas | Opciones de ortografía** abre un cuadro de diálogo (*imagen siguiente*) donde puede definir las opciones de la revisión ortográfica.



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:

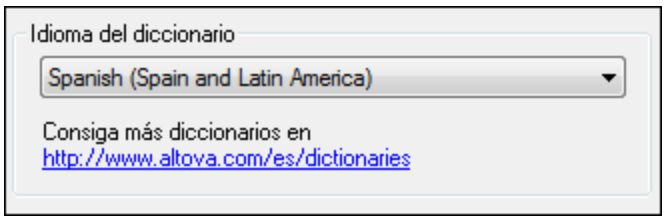
```
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)
\en_GB.aff
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)
\en_GB.dic
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US)\en_US.aff
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (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:

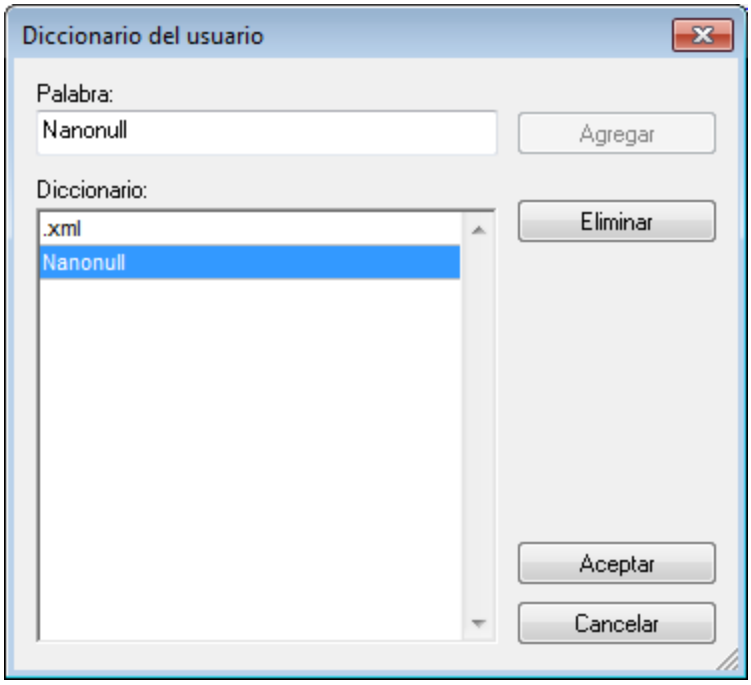
- Puede añadir diccionarios Hunspell a una subcarpeta nueva de la carpeta `Lexicons`. Los diccionarios Hunspell se pueden descargar desde https://wiki.services.openoffice.org/wiki/Dictionaries o desde https://extensions.services.openoffice.org/en/dictionaries, por ejemplo. (Recuerde que OpenOffice utiliza 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 866. Pulse el botón **Agregar al diccionario** para añadir la palabra desconocida al diccionario del usuario.

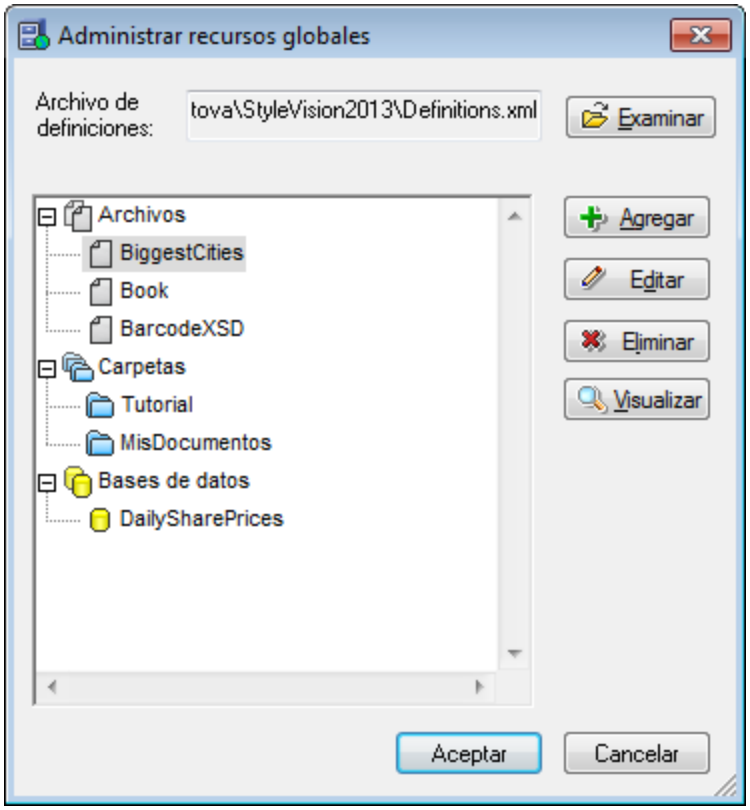
El diccionario del usuario se encuentra en este directorio: `C:\Users\<usuario>\Documentos\Altova\SpellChecker\Lexicons\user.dic`

16.14.3 Recursos globales

El comando **Herramientas | Recursos globales** abre el cuadro de diálogo "Administrar recursos globales" (*imagen siguiente*). En este cuadro de diálogo puede:

 © 2016-2022 Altova GmbH

- 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.

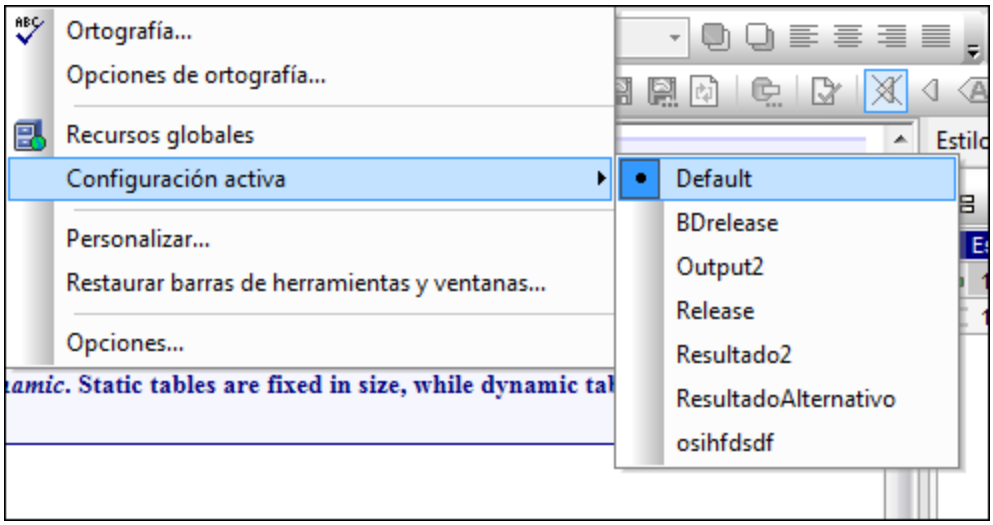


Para más información consulte la sección Definir recursos globales 444.

Nota: también puede abrir el cuadro de diálogo "Administrar recursos globales" desde la barra de herramientas Recursos globales 809 (**Vista | Barras de herramientas | Recursos globales**).

16.14.4 Configuración activa

Al pasar el puntero del ratón sobre el comando **Herramientas | Configuración activa** aparece un submenú con todas las configuraciones definidas en el archivo XML de recursos globales 444 activo (*imagen siguiente*).



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.

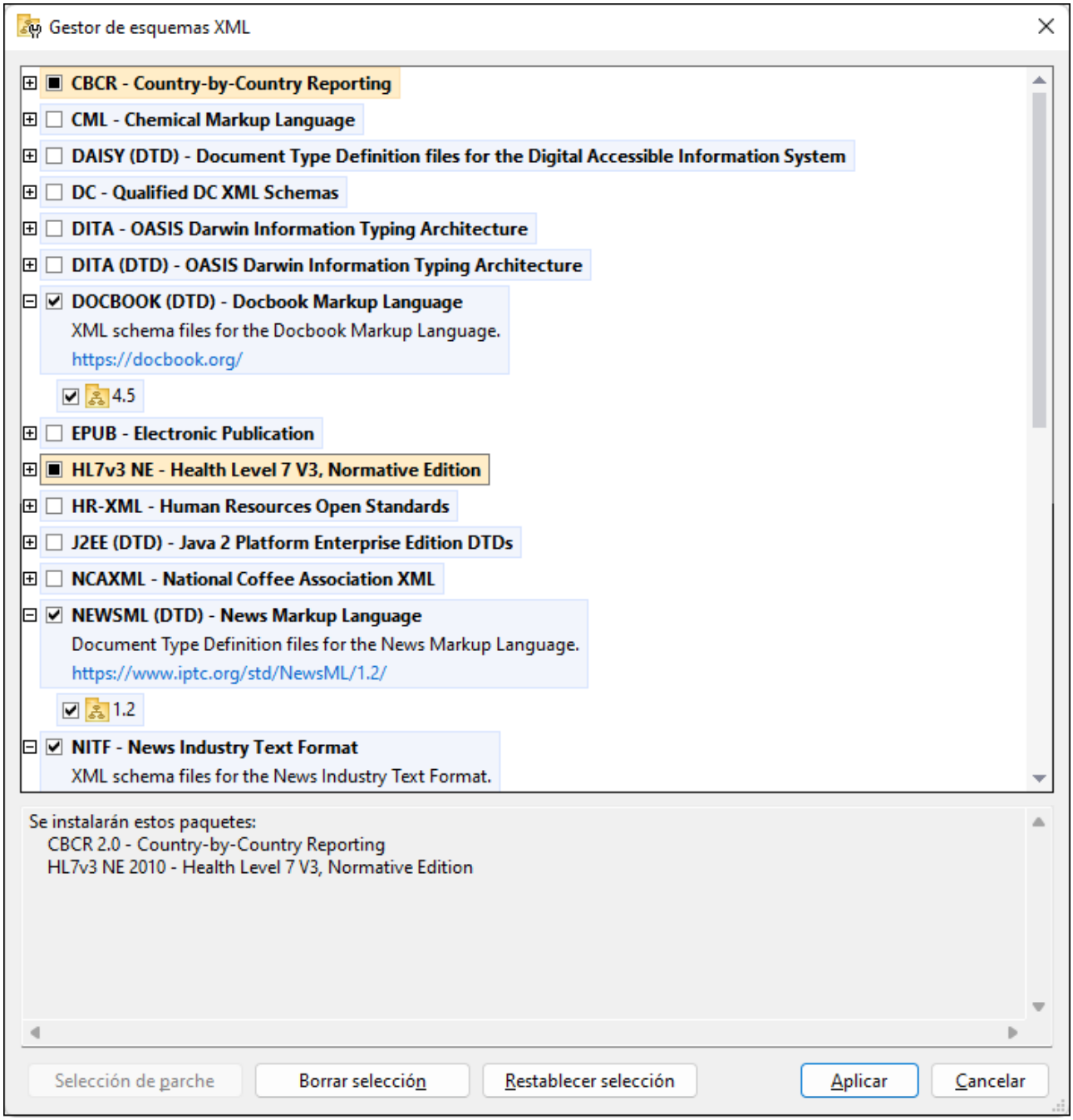
Nota: también puede cambiar de configuración en la barra de herramientas Recursos globales 809 (**Vista | Barras de herramientas | Recursos globales**).

16.14.5 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 StyleVision.

- 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` 243, 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 StyleVision por primera vez en un mismo día natural.
- si StyleVision 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** 246 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** 244 desde la línea de comandos y después ejecute el comando **initialize** 242.

16.14.5.1 Ejecutar el gestor de esquemas

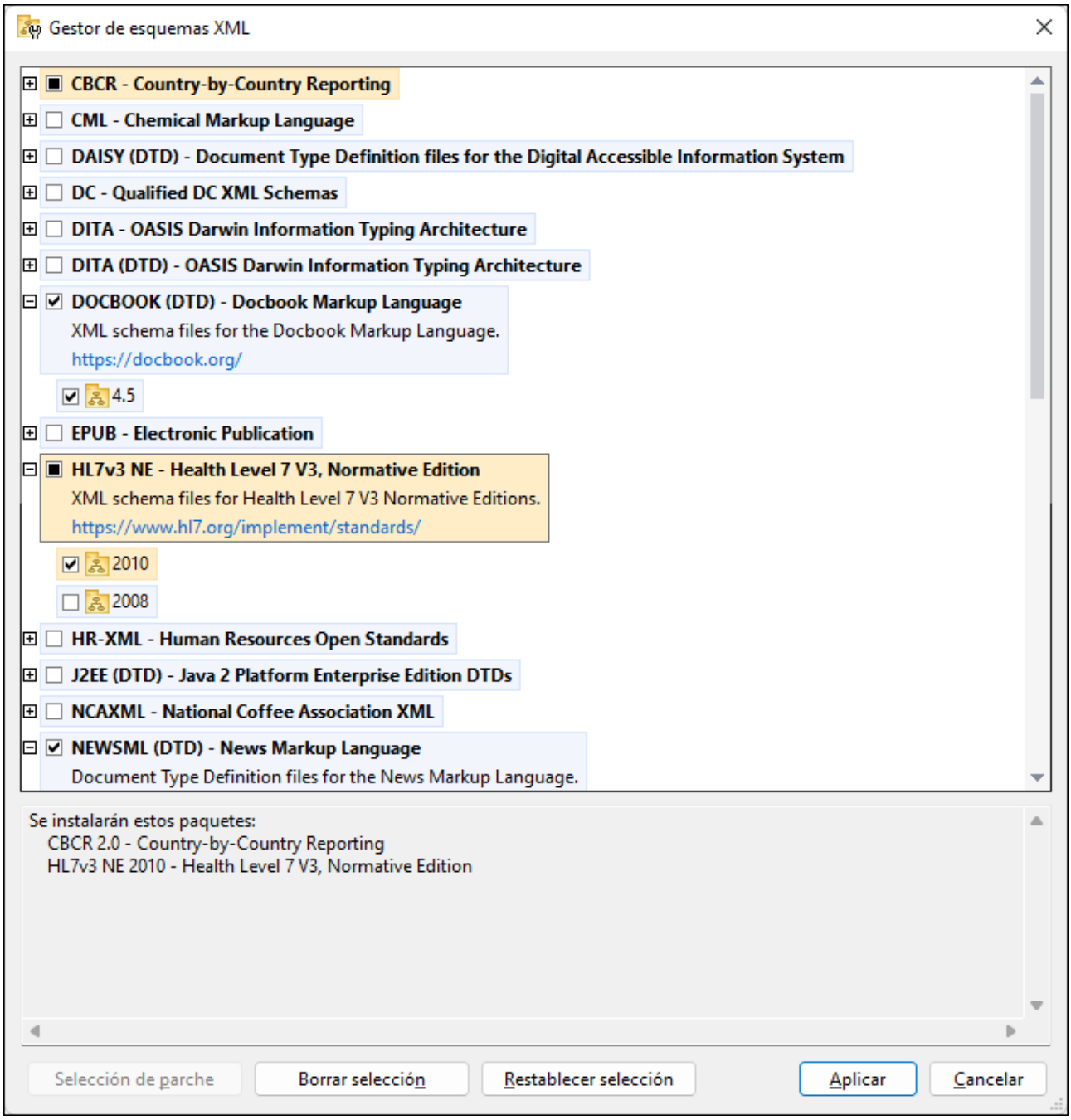
Interfaz gráfica del usuario

Hay varias formas de acceder a la IGU del Gestor de esquemas:

- *Durante la instalación de StyleVision:* 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 StyleVision:* 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 241, a continuación.

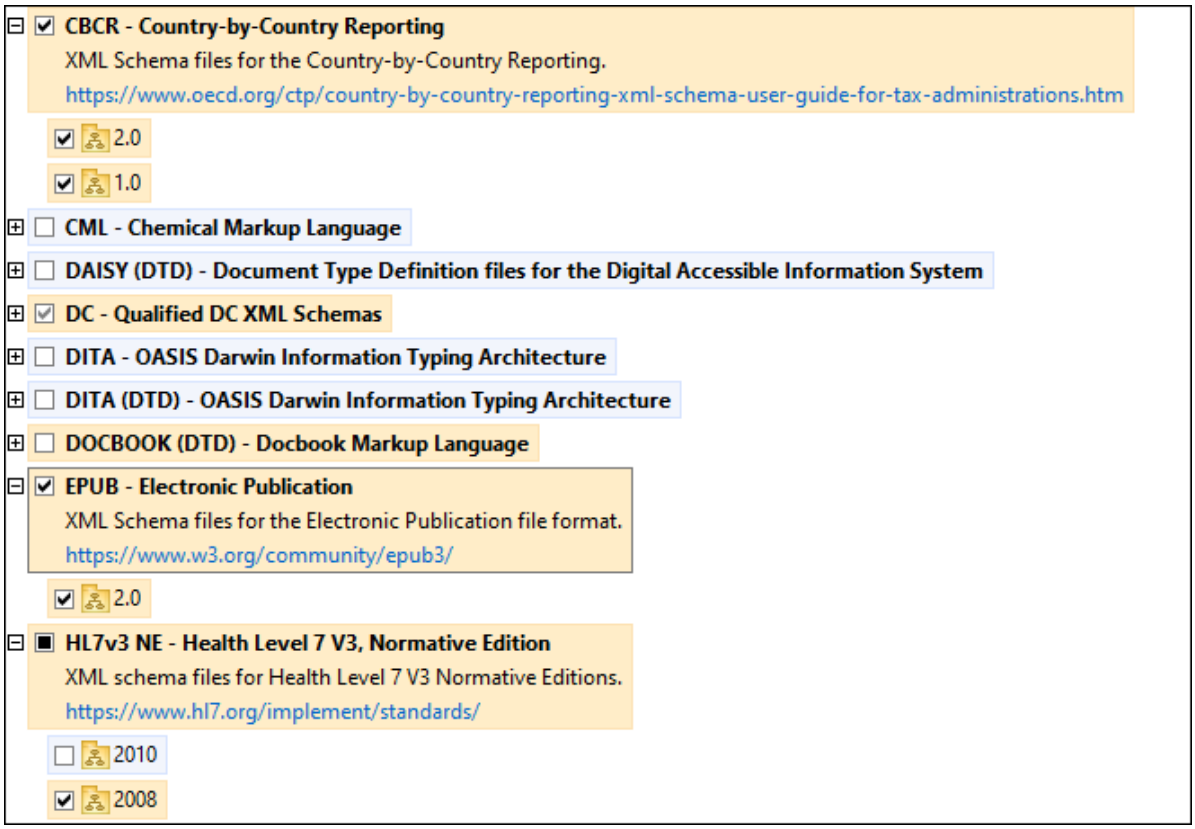
Para mostrar al ayuda de los comandos ejecute:

- *en Windows:* `xmlschemamanager.exe --help`
- *en Linux o macOS (solo para aplicaciones de servidor):* `sudo ./xmlschemamanager --help`

16.14.5.2 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

16.14.5.3 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* 237.) 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` 244. Aparece una lista con los esquemas para las que hay parches disponibles.
1. Ejecute el comando `upgrade` 247 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` 244. Puede introducir tantos esquemas como quiera. Para más detalles consulte la descripción del comando `install` 243.

Nota: en Linux o macOS, use el comando **`sudo ./xmlschemamanager`**.

Instalar un esquema requerido
Si ejecuta un comando en StyleVision y StyleVision 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**.

16.14.5.4 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` 245:

```
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` 244. Puede introducir tantos esquemas como quiera. Para más detalles consulte la descripción del comando `uninstall` 245.

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` 244.

 © 2016-2022 Altova GmbH

Una vez haya ejecutado este comando, asegúrese de que ejecuta también el comando initialize 242 para recrear el directorio caché. También puede ejecutar el comando reset 244 con la opción -i.

Recuerde que reset -i 244 restaura la instalación original del producto, por lo que es recomendable ejecutar el comando update 246 después de restaurar el gestor. Puede ejecutar el comando reset 244 con las opciones -i o -u.

16.14.5.5 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.

16.14.5.5.1 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
```

16.14.5.5.2 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`[244].)
- 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
```

16.14.5.5.3 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
```

16.14.5.5.4 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` 244. 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* 232.

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
```

16.14.5.5.5 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**

16.14.5.5.6 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 242 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 246 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 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**

16.14.5.5.7 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` 244. 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* 232.

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
```

16.14.5.5.8 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` 244 e `initialize` 242.

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

16.14.5.5.9 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`:

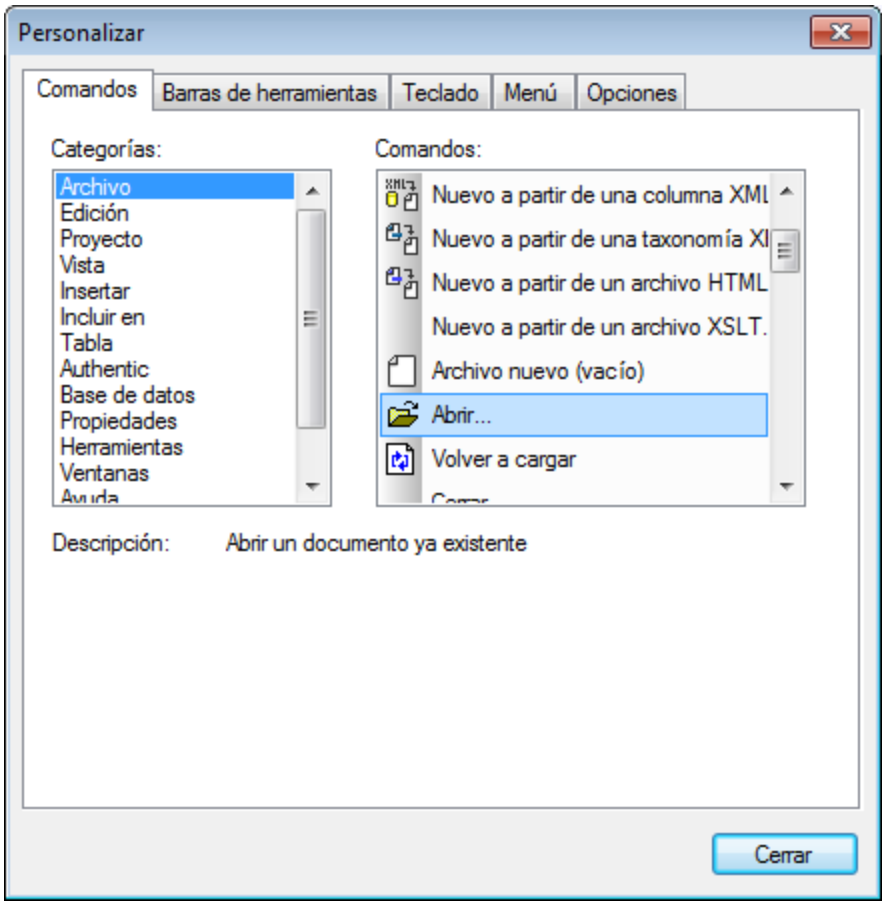
`--help, --h`	Muestra la ayuda sobre este comando en la línea de comandos.
`--silent, --s`	Muestra solamente los mensajes de error. El valor predeterminado es **`false`**.
`--verbose, --v`	Muestra información suplementaria durante la ejecución. El valor predeterminado es **`false`**.

16.14.6 Personalizar

El comando **Herramientas | Personalizar** sirve para personalizar StyleVision y ajustar la aplicación a sus requisitos y preferencias. Al hacer clic en este comando se abre el cuadro de diálogo "Personalizar" que está compuesto por varias pestañas.

Comandos

En la pestaña *Comandos* del cuadro de diálogo "Personalizar" 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.



Para **añadir un comando** a una barra de herramientas o menú:

1. 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*.
2. 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**.
3. 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, 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.

 © 2016-2022 Altova GmbH

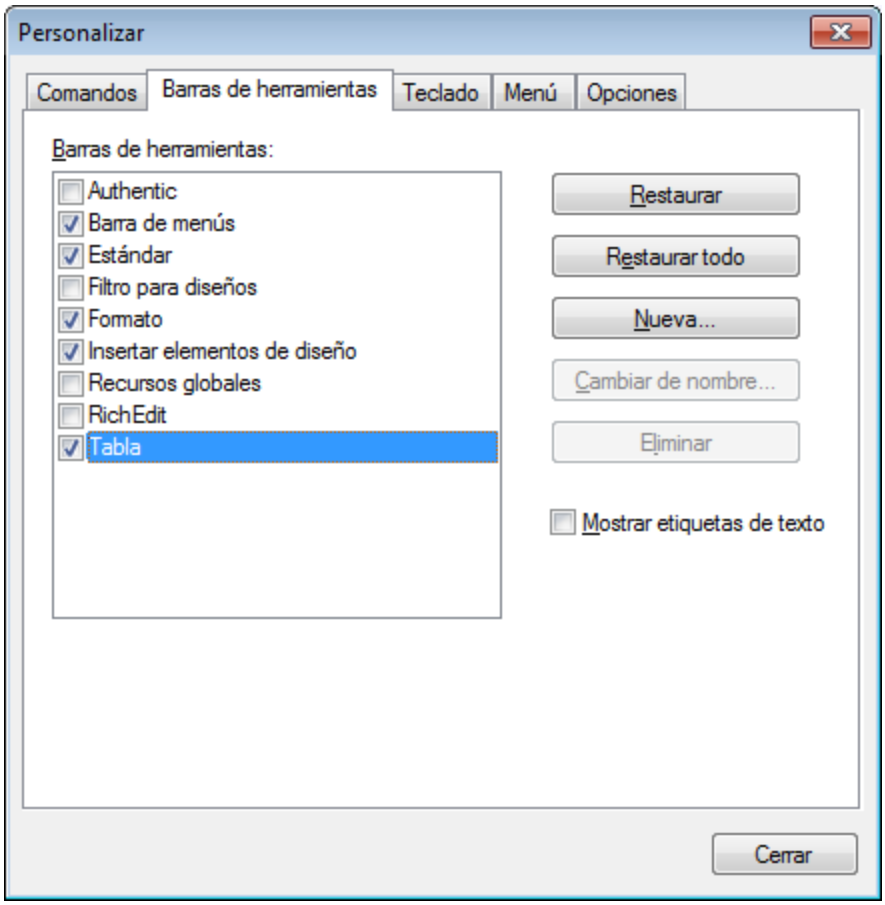
Para eliminar una barra de menús o comando, en el cuadro de diálogo "Personalizar" haga clic con el botón derecho en un menú o comando de menú y seleccione **Eliminar** en el menú contextual que aparece.

Nota:

- Estas operaciones se pueden realizar tanto si hay un documento abierto como si no.
- Para restaurar los menús y las barras de herramientas a su estado inicial de instalación, abra la pestaña *Barras de herramientas* y haga clic en el botón **Restaurar** correspondiente.

Barras de herramientas

En la pestaña *Barras de herramientas* puede (i) activar/desactivar barras de herramientas (es decir, puede decidir qué barras de herramientas aparecen en la interfaz); (ii) definir qué iconos aparecen en cada barra de herramientas; (iii) crear barras de herramientas personalizadas; (iv) restaurar las barras de menús y de herramientas a su estado de instalación.



La interfaz de StyleVision incluye una barra de menús fija y varias barras de herramientas opcionales (Authentic, Filtro para diseños, Formato, Estándar, Tabla y Tabla de contenido).

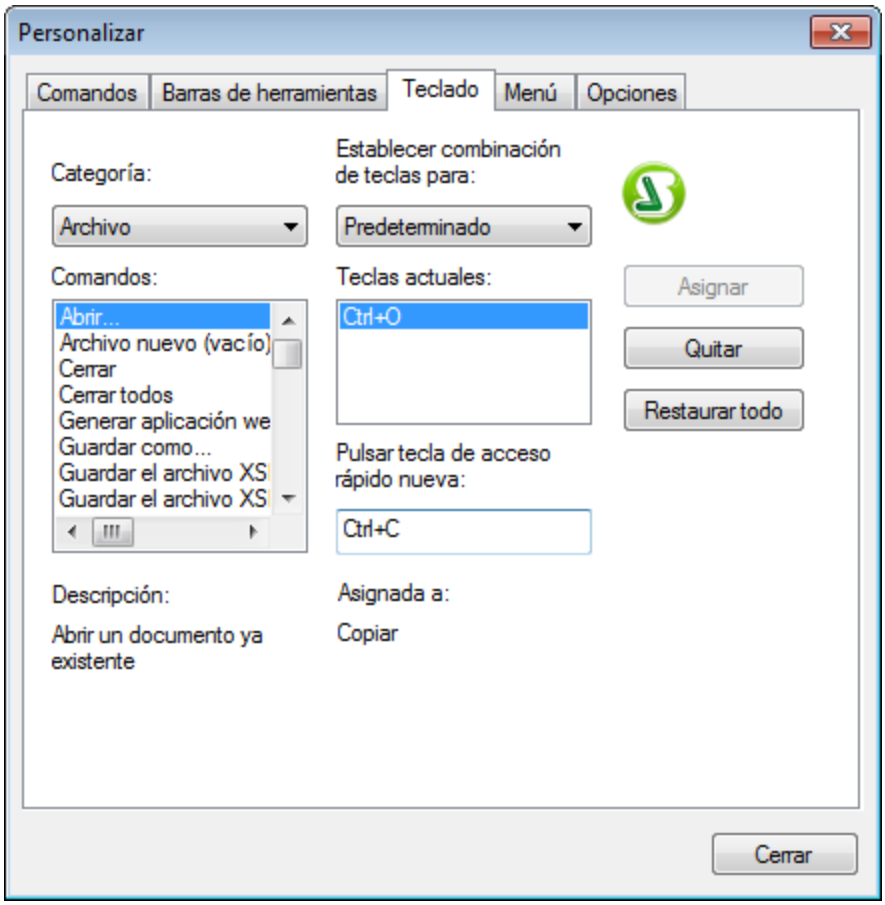
Cada barra de herramientas puede tener varios grupos de comandos y puede añadir más comandos a cada barra de herramientas desde la pestaña *Comandos*. Las barras de herramientas se pueden arrastrar desde su posición actual hasta cualquier posición de la pantalla. Al hacer doble clic en su barra de título, la barra de herramientas se acopla o flota en la pantalla.

En la pestaña *Barras de herramientas* puede activar/desactivar cualquier barra de herramientas con solo activar/desactivar su casilla. También puede elegir si las etiquetas de texto de los elementos de la barra de herramientas se muestran u ocultan. Para ello seleccione la barra de herramientas en la pestaña *Barras de herramientas* y marque la casilla *Mostrar etiquetas de texto.* Para devolver una barra de herramientas a su estado de instalación, haga clic en el botón **Restaurar**. Para devolver todas las barras de herramientas y la barra de menús a su estado de instalación, haga clic en **Restaurar todo**.

Nota: como ya sabe, puede añadir y eliminar comandos en la barra de menús. Para restaurar la barra de menús a su estado de instalación, abra la pestaña *Barras de herramientas* del cuadro de diálogo "Personalizar", seleccione **Barra de menús** y después haga clic en el botón **Restaurar**.

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 StyleVision.



Para asignar una tecla de acceso rápido a un comando:

1. Seleccione la categoría a la que pertenece el comando en el cuadro combinado *Categoría*.
2. Seleccione el comando al que desea asignar una tecla de acceso rápido en el cuadro de lista *Comandos*.
3. Haga clic en campo de entrada *Pulsar tecla de acceso rápido nueva*: y después pulse las teclas que deben activar el comando. La tecla de acceso rápido aparece inmediatamente en el campo de entrada *Pulsar tecla de acceso rápido nueva*. Si esta tecla de acceso rápido ya está asignada a un comando,

el comando aparece debajo del campo de entrada. Por ejemplo, en la imagen anterior, **Ctrl+C** ya está asignada al comando **Copiar** y no se puede asignar al comando **Abrir archivo**. Para borrar el campo de entrada *Pulsar tecla de acceso rápido nueva*, pulse cualquier tecla de control (**Ctrl**, **Alt** o **Mayús**).

4. Haga clic en el botón **Asignar** para asignar la tecla de acceso rápido al comando de forma permanente. La tecla de acceso rápido aparece ahora en la lista *Teclas actuales*.

Para eliminar una tecla de acceso rápido

1. Seleccione el comando cuya tecla de acceso rápido quiere eliminar.
2. En la lista *Teclas actuales* haga clic en la tecla de acceso rápido que desea eliminar.
3. Ahora haga clic en el botón **Quitar**.

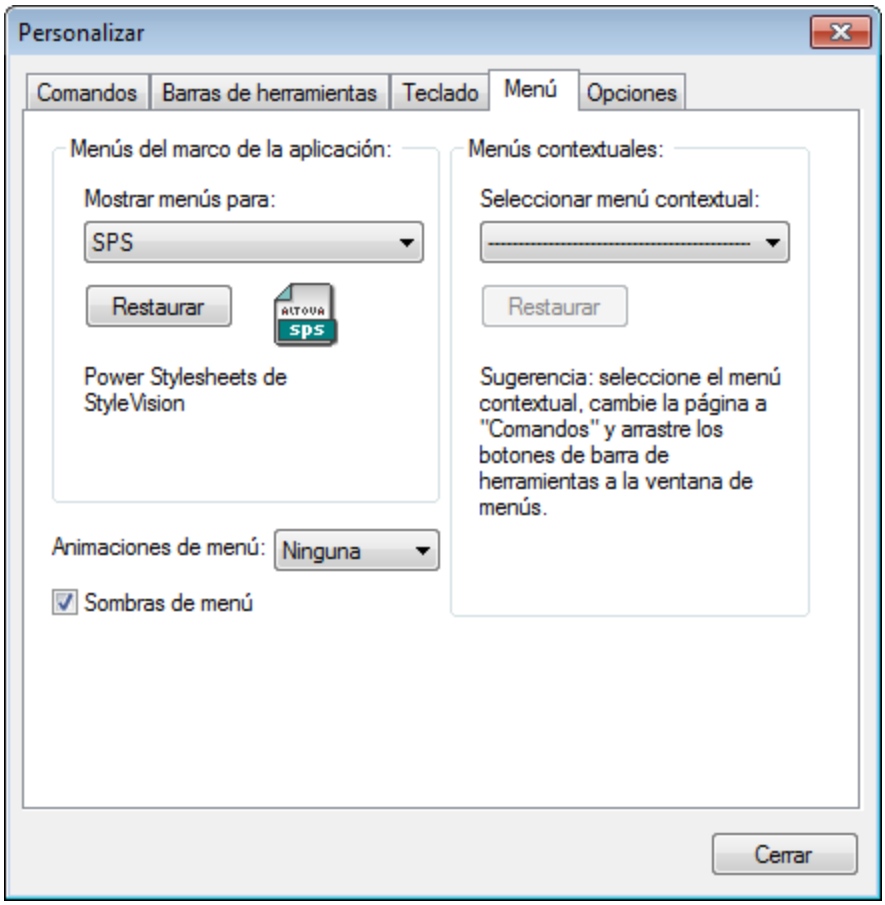
Para restaurar todas las asignaciones de teclas de acceso rápido

1. Haga clic en el botón **Restaurar todo** para devolver las asignaciones de teclas de acceso rápido a su estado de instalación. Aparece un diálogo donde debe confirmar que deesa restaurar todas las asignaciones de teclas de acceso rápido.
2. Haga clic en **Sí** para restaurar todas las asignaciones de teclas de acceso rápido.

Establecer combinación de teclas para
Esta opción no tiene ninguna función actualmente

Menú

En la pestaña *Menú* puede personalizar las dos barras de menús principales (la barra de menús predeterminada y la barra de menús de la aplicación) así como los menús contextuales de la aplicación. Hay dos tipos de barra de menús principal: *predeterminada* (que aparece cuando no hay ningún documento abierto) y *SPS* (que aparece cuando hay abierto un documento SPS).



Para personalizar un menú

1. En la pestaña *Menú* seleccione la barra de menús que desea personalizar (p. ej. en la imagen anterior seleccionamos el menú *SPS*).
2. Ahora cambie a la pestaña *Comandos* y arrastre comandos desde el cuadro de lista *Comandos* hasta la barra de menús.

Para eliminar comandos de un menú

1. Haga clic con el botón derecho en el comando o icono de la barra de herramientas que representa el comando.
2. Seleccione la opción **Eliminar** del menú emergente o arrastre el comando fuera de la barra de menús.

Para restaurar las barras de menús

1. Seleccione la entrada de menú en el cuadro combinado.
2. Haga clic en el botón **Restaurar** situado justo debajo del nombre del menú. Aparece un diálogo donde debe confirmar que quiere restaurar la barra de menús.

Para personalizar un menú contextual (los que aparecen al hacer clic con el botón derecho)

1. Seleccione el menú contextual en el cuadro combinado *Menús contextuales*.
2. Ahora cambie a la pestaña *Comandos* y arrastre comandos desde el cuadro de lista *Comandos* hasta el menú contextual.

© 2016-2022 Altova GmbH

Para eliminar comandos de un menú contextual

1. Haga clic con el botón derecho en el comando o icono que representa al comando.
2. Seleccione la opción **Eliminar** del menú emergente o arrastre el comando fuera de la barra de menús.

Para restaurar un menú contextual

1. Seleccione el menú contextual en el cuadro combinado *Menús contextuales*.
2. Haga clic en el botón **Restaurar** situado justo debajo del nombre del menú. Aparece un diálogo donde debe confirmar que quiere restaurar la barra de menús.

Para cerrar la ventana de un menú contextual

- haga clic en el icono **Cerrar** situado en la esquina superior derecha de la barra de título o
- haga clic en el botón **Cerrar** del cuadro de diálogo "Personalizar".

Animaciones de menú

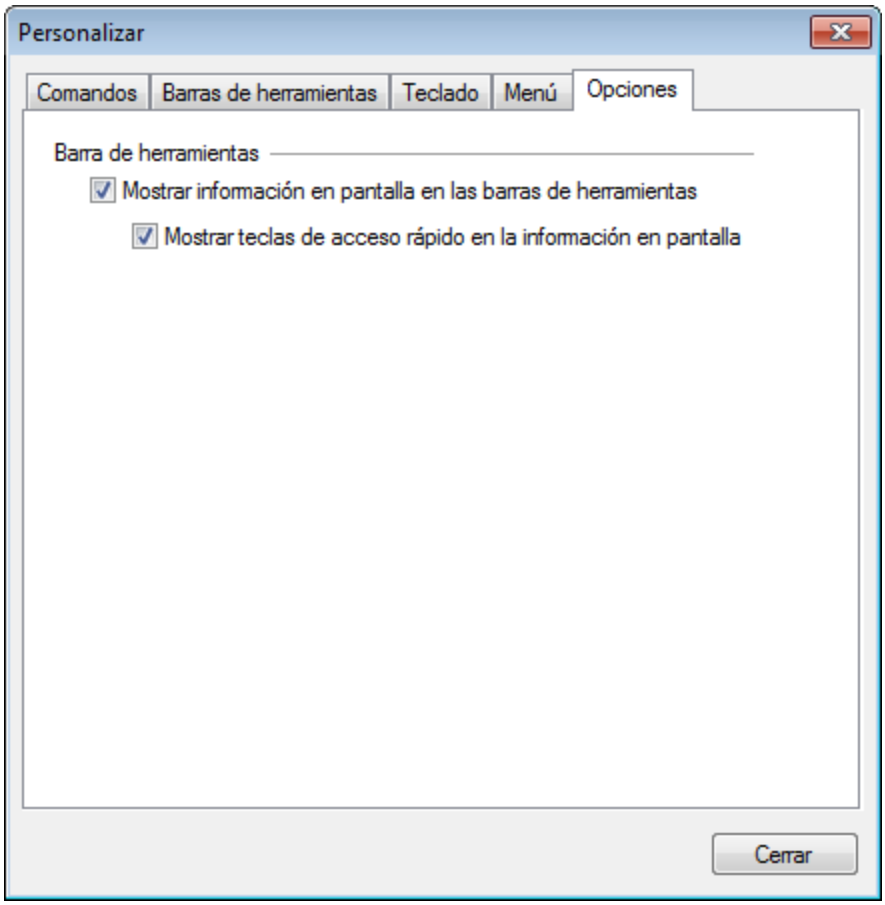
Esta opción especifica cómo deben aparecer los menús cuando se hace clic en ellos. Seleccione una opción de la lista desplegable.

Sombras de menú

Marque esta casilla si quiere que los menús tengan una sombra alrededor.

Opciones

En la pestaña *Opciones* puede personalizar otras características de la barra de herramienta.



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.

16.14.7 Restaurar barras de herramientas y ventanas

El comando **Herramientas | Restaurar barras de herramientas y ventanas** cierra StyleVision y lo reinicia con su configuración predeterminada. Antes de cerrarse, StyleVision le pregunta si desea cerrar o no la aplicación.

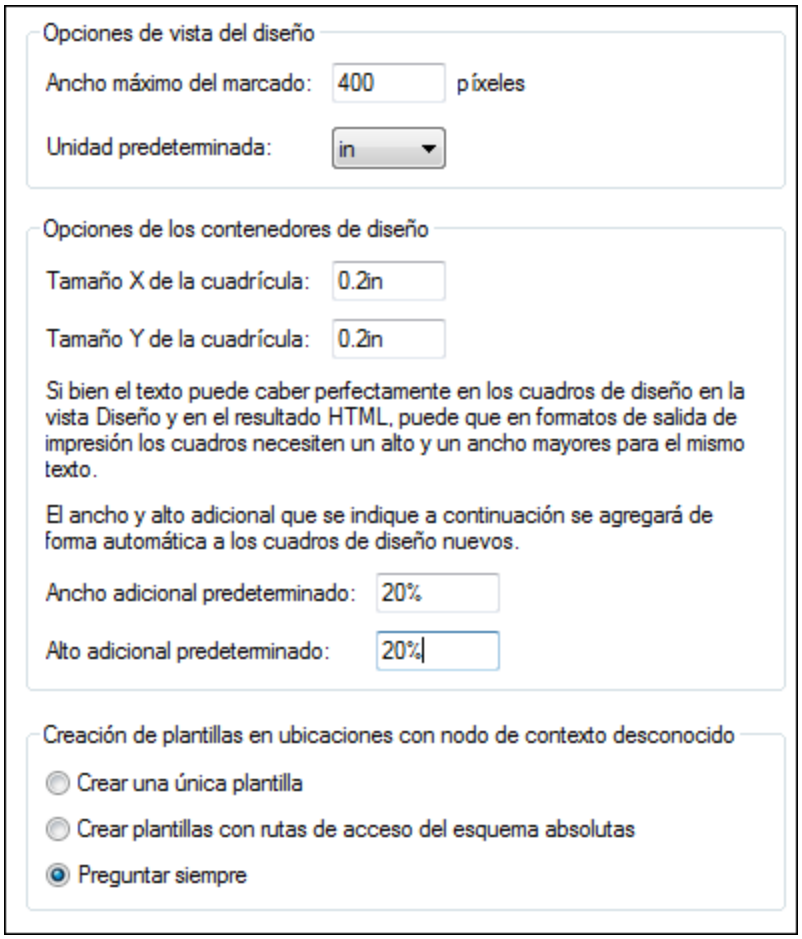
16.14.8 Opciones

El comando **Opciones** abre un cuadro de diálogo (*imagen siguiente*) donde puede configurar la codificación del archivo HTML de salida.

Diseño

En la pestaña *Diseño* del cuadro de diálogo "Opciones" (*imagen siguiente*) puede definir opciones generales de configuración para la vista **Diseño**.

© 2016-2022 Altova GmbH



Estas son las opciones disponibles:

- *Ancho máximo del marcado (en píxeles)*. Aquí debe introducir un número entero positivo.
- *Tamaño de la cuadrícula de los contenedores de diseño (en unidades de longitud absolutas)*. La longitud especificada es la distancia entre dos puntos en el eje correspondiente de la cuadrícula.
- *Ancho y alto adicional predeterminado para los cuadros de diseño*. Estas medidas adicionales se añaden a todos los cuadros de diseño porque a menudo se necesita un poco más de espacio para representar texto en los formatos para medios impresos. Estos valores pueden indicarse como porcentajes o como unidades de longitud absolutas.
- *Comportamiento predeterminado cuando se crea una plantilla de nodo en una posición donde se desconoce el nodo de contexto*. Esta opción suele afectar a las plantillas definidas por el usuario creadas para elementos cuyo contexto se desconoce. Si crea un nodo dentro de una plantilla de este tipo, entonces hay tres opciones: (i) el nodo se crea solo con su nombre, (ii) el nodo se crea con su ruta de acceso completa o (iii) StyleVision pregunta siempre qué quiere hacer en cada situación. La opción predeterminada es *Preguntar siempre.*

Esquema

En la pestaña *Esquema* puede configurar el orden en que aparecen los elementos y atributos en la ventana Estructura del esquema. La opción predeterminada es que los atributos aparezcan por orden alfabético y que los elementos sigan el orden de la estructura del esquema siempre que sea posible.

Proyecto

En la pestaña *Proyecto* puede especificar qué ocurre cuando se hace doble clic en un archivo XML o XSD en la ventana Proyecto. Hay tres posibilidades: (i) el archivo se abre en XMLSpy, donde puede editarlo; (ii) se crea un diseño nuevo basado en el archivo seleccionado; (iii) StyleVision pregunta al usuario qué acción se debe ejecutar.

Codificación predeterminada (de los archivos de salida)

En la pestaña *Codificación predeterminada* puede configurar la codificación de los diferentes formatos de salida que StyleVision puede generar. La codificación indica los puntos de código para los diferentes juegos de caracteres. Cada formato de salida ofrece una lista desplegable con todas las opciones de codificación disponibles. Seleccione la codificación de cada formato de salida y haga clic en **Aceptar**. A partir de ese momento, cada archivo SPS que cree tendrá definidas estas opciones de codificación de salida.

En los archivos XSLT-para-HTML, la información de codificación de salida se registra aquí:

- En el atributo `encoding` del elemento `xsl:output` de la hoja de estilos:
  `<xsl:output version="1.0" encoding="UTF-8" indent="no" omit-xml-declaration="no" media-type="text/html" />`
- En el atributo `charset` del elemento `content-type meta` del encabezado HTML:
  `<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />`

En los archivos XSLT-para-RTF, la información de codificación de salida se registra en el atributo `encoding` del elemento `xsl:output` de la hoja de estilos:

```
<xsl:output version="1.0" encoding="ISO-8859-1" indent="no" method="text" omit-xml-declaration="yes" media-type="text/rtf" />
```

Nota: estas son las opciones de codificación de salida que se usarán para los archivos SPS nuevos. Es decir, la codificación de los archivos SPS ya existentes no se pueden cambiar en este cuadro de diálogo. La codificación de salida de los archivos SPS ya existentes se cambia con el comando Archivo | Propiedades 787.

Opciones de XSL

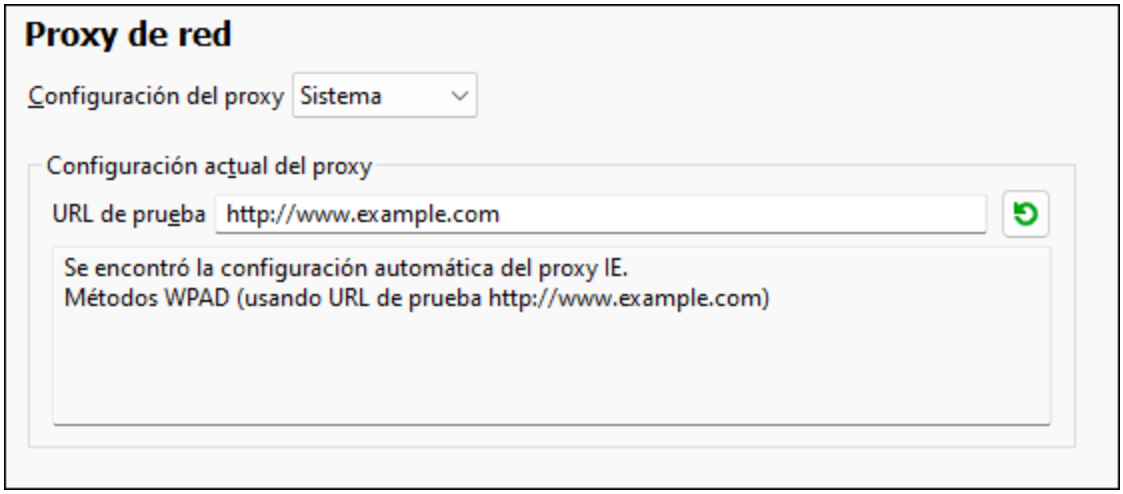
Si usa la versión de prueba de StyleVision, el comentario *Generado con StyleVision* aparecerá en los archivos HTML y PDF de salida. Si adquiere una licencia de StyleVision, en la pestaña Opciones de XSL puede elegir si se incluye o no este comentario.

Opciones del 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.

 © 2016-2022 Altova GmbH



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 (`.`).

Nota: si ha configurado un servidor de proxy y quiere implementar una transformación en Altova FlowForce Server, debe seleccionar la opción *No usar el servidor proxy para direcciones locales*.

Configuración actual del proxy
Proporciona un registro detallado de la detección del proxy. Se puede actualizar con el botón **Actualizar** a la derecha de *URL de prueba* (por ejemplo, al cambiar la URL de prueba o cuando se ha cambiado la configuración del proxy).

- *URL de prueba:* una URL de prueba se puede usar para ver qué proxy hay que usar para esa URL en concreto. No se trata de una URL de entrada/salida. Este campo no debe estar vacío si se ha optado por la configuración automática del proxy (seleccionando *Usar la configuración del proxy del sistema* o *Configuración automática del proxy*).

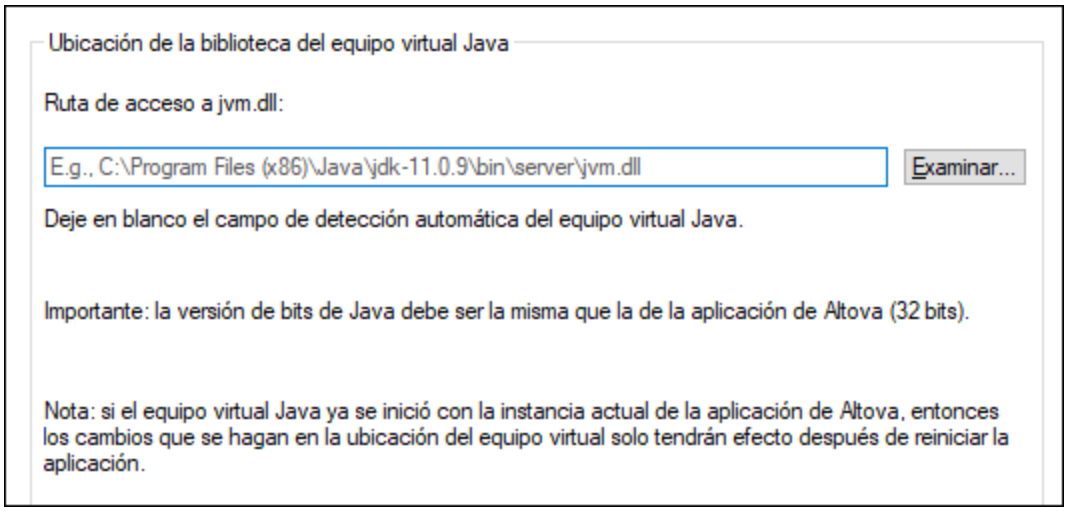
Opciones Java

En la pestaña *Java* puede introducir la ruta de acceso a un equipo virtual java en su sistema de archivos.

 © 2016-2022 Altova GmbH

Tenga en cuenta que no siempre es necesario agregar una ruta de acceso personal a un equipo virtual. Por defecto, StyleVision 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 StyleVision haya detectado automáticamente.



Observe lo siguiente:

- 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 StyleVision (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 StyleVision 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
- compatibilidad con códigos de barras

16.15 Menú Ventanas

Los comandos del menú **Ventanas** sirven para personalizar la disposición de las ventanas de la interfaz gráfica de StyleVision (en cascada, en mosaico o maximizadas). También puede maximizar una ventana haciendo clic en el botón **Maximizar** situado en la esquina superior derecha de la ventana.

Además, el menú **Ventanas** enumera todas las ventanas de documento que están abiertas en cada momento y la ventana activa aparece marcada en la lista. Para pasar a otra ventana haga clic en el nombre de la ventana correspondiente. El nombre de la ventana es el nombre del documento que está abierto en esa ventana.

Cuadro de diálogo "Ventanas"
Al final de la lista de ventanas abiertas aparece el comando **Ventanas...**, que abre el cuadro de diálogo del mismo nombre. El cuadro de diálogo "Ventanas" enumera todas las ventanas que están abiertas e incluye botones para realizar operaciones con esta ventanas.

Advertencia: para salir del cuadro de diálogo "Ventanas" debe hacer clic en **Aceptar**. NO haga clic en **Cerrar ventanas** porque ese botón cierra las ventanas que estén seleccionadas en la lista del diálogo.

© 2016-2022 Altova GmbH

16.16 Menú Ayuda

Los comandos del menú **Ayuda** sirven para abrir la ayuda en pantalla de StyleVision, para obtener información sobre la aplicación y para abrir páginas de soporte y ayuda en el sitio web de Altova. Este menú también incluye un comando para abrir el cuadro de diálogo Activación del software 901, donde puede introducir los datos de su código de licencia.

Los comandos del menú **Ayuda** se describen uno a uno en los apartados siguientes:

- **Contenido, Índice, Buscar** 901
- **Activación del software, Formulario de pedido, Registro del software, Buscar actualizaciones** 901
- **Enlaces al sitio web de Altova** 905

16.16.1 Contenido, Índice, Buscar

Contenido

Abre la ayuda en pantalla de StyleVision 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 StyleVision 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 StyleVision 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.

16.16.2 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:

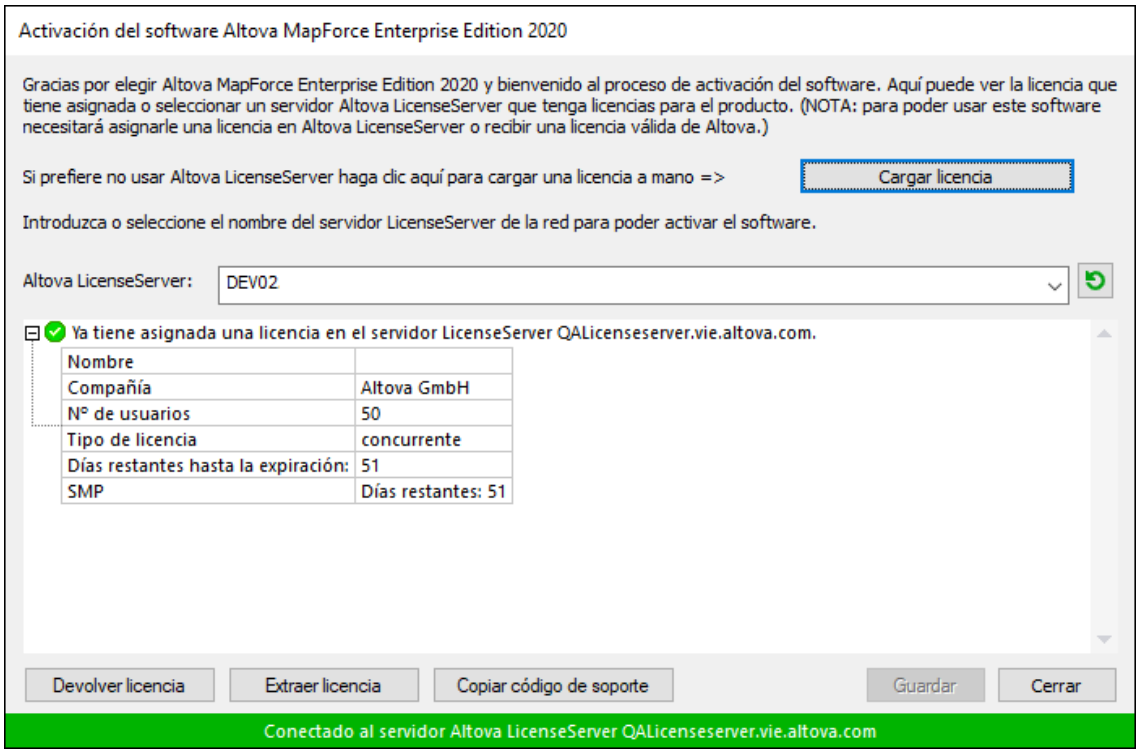
- Guardar el archivo de licencia (**.altova_licenses**) en su equipo, hacer doble clic en el archivo de licencia, introducir los detalles necesarios en el cuadro de diálogo que aparece y finalmente hacer clic en **Aplicar claves**.
- Guardar el archivo de licencia (**.altova_licenses**) en su equipo. 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:

 © 2016-2022 Altova GmbH

- 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.



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.

© 2016-2022 Altova GmbH

⊞ Buscar actualizaciones

Comprueba si existe una versión más reciente del producto en el servidor de Altova y emite un mensaje a tal efecto.

16.16.3 Enlaces al sitio web de Altova

⊟ 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.

⊟ 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.

⊟ StyleVision en Internet

Es un enlace al sitio web de Altova, donde encontrará más información sobre StyleVision, otros productos de Altova y tecnologías relacionadas.

⊟ Acerca de StyleVision

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.

17 Referencia del programador

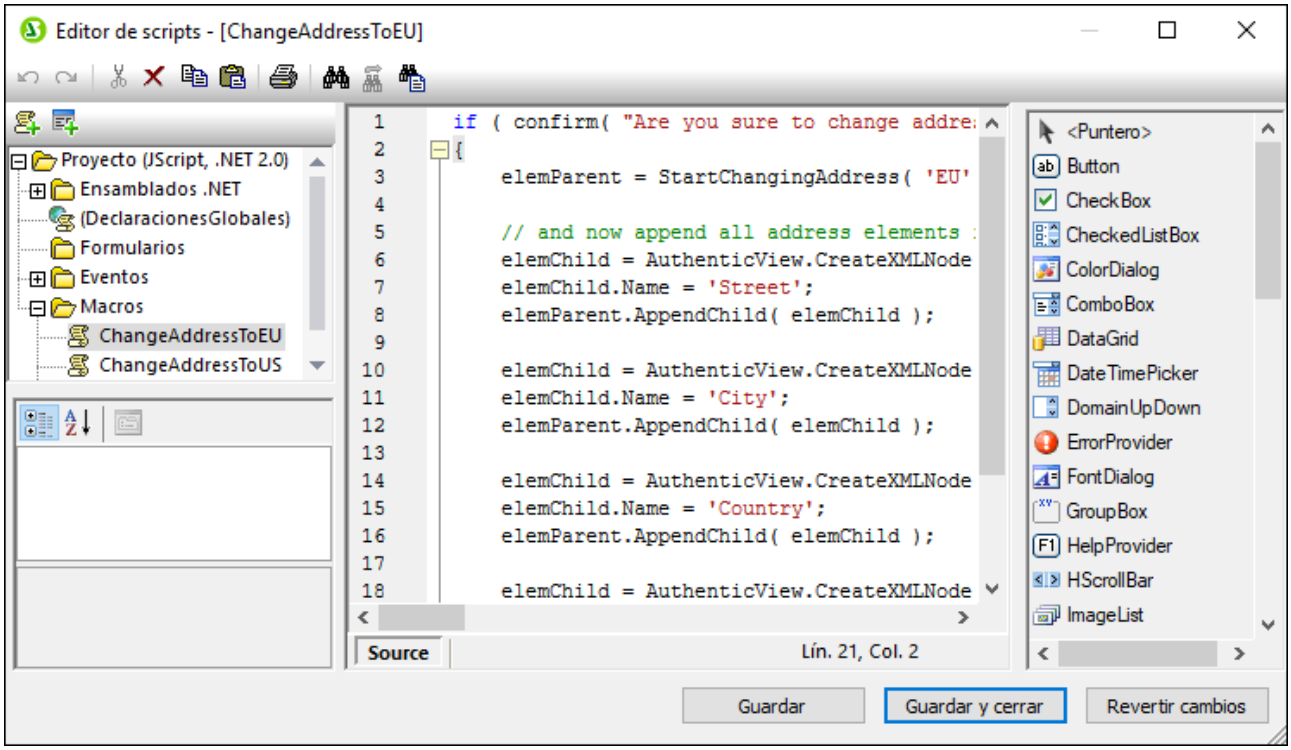
StyleVision como servidor de automatización
StyleVision es un servidor de automatización. Esto quiere decir que es una aplicación que expone objetos programables a otras aplicaciones (llamados *clientes de automatización*). Como resultado, un cliente de automatización puede acceder a los objetos y funciones que pone a su disposición el servidor de automatización. Esto ofrece grandes ventajas al cliente de automatización porque puede servirse de las funciones de StyleVision. Por ejemplo, un cliente de automatización puede generar un archivo XSLT a partir de un SPS con ayuda de StyleVision. Por tanto, los desarrolladores pueden mejorar sus aplicaciones usando las funciones ya disponibles de StyleVision.

Los objetos programables de StyleVision se ponen a disposición de los clientes de automatización a través de la API de StyleVision, que es una API de COM. En esta documentación se ofrece una descripción completa de todos los objetos disponibles (consulte el apartado *Application API* 931).

 © 2016-2022 Altova GmbH

17.1 Editor de scripts

El Editor de scripts es un entorno de desarrollo integrado en StyleVision desde donde puede personalizar las funciones de la Vista Authentic con ayuda de los scripts JScript o VBScript. Por ejemplo, puede añadir a la barra de herramientas un botón nuevo que ejecute una tarea personalizada, como dar formato al texto, o puede hacer que StyleVision muestre un cuadro de mensaje de notificaciones cada vez que se ejecute una acción en la Vista Authentic.Para ello debe crear proyectos de scripting, que son archivos con la extensión `.asprj` (Altova Scripting Project).



Editor de scripts

Los proyectos de scripting son portátiles; puede crearlos como parte de un archivo .sps (StyleVision Power Stylesheet). Cualquier aplicación de Altova que consuma ese archivo .sps puede ejecutar las macros dentro del proyecto. La aplicación no tiene por qué ser StyleVision, también puede ser XMLSpy o Authentic Desktop.

> El Editor de scripts necesita que instale .NET Framework 2.0 o más avanzado antes de instalar StyleVision.

17.1.1 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

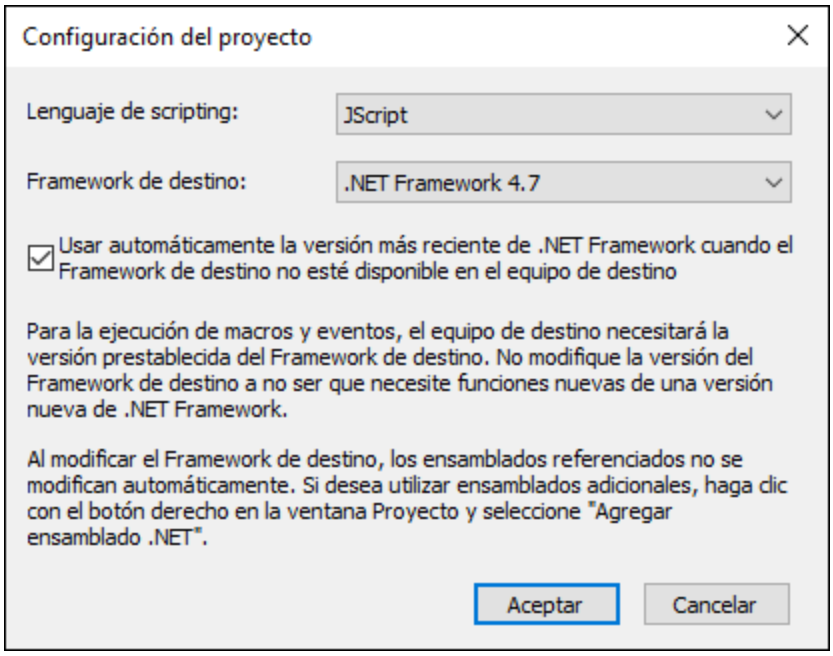
de eventos). También puede añadir variables globales y funciones a un script "Declaraciones globales", que hace que esas variables y funciones sean accesibles en todo el proyecto. Para empezar un proyecto nuevo :

1. Cree o abra el archivo .sps del que el proyecto de scripting debe formar parte.
2. Elija una de estas opciones:
   - En el menú **Authentic** haga clic en **Scripts de Authentic**.
   - Haga clic en la pestaña *Diseño*, en la parte inferior de la ventana principal, y después seleccione **Script de Authentic**.

Cada archivo .sps puede tener asignado un único proyecto de scripting. Para abrir un proyecto de scripting asociado a un archivo .sps puede seguir los pasos indicados en el punto anterior, con los que se abre la ventana Editor de scripts. Conforme va editando el proyecto de scripting puede guardar los cambios temporalmente en la memoria haciendo clic en el botón **Guardar**, en la parte inferior del Editor de scripts. Para guardar el proyecto de scripting en disco primero debe hacer clic en **Guardar y cerrar** en el Editor de scripts y después guardar el archivo .sps.

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).

 © 2016-2022 Altova GmbH

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 ensamblados de caché de ensamblados global de .NET (GAC por sus siglas en inglés) o archivos .dll personalizados. Puede importar ensamblados:

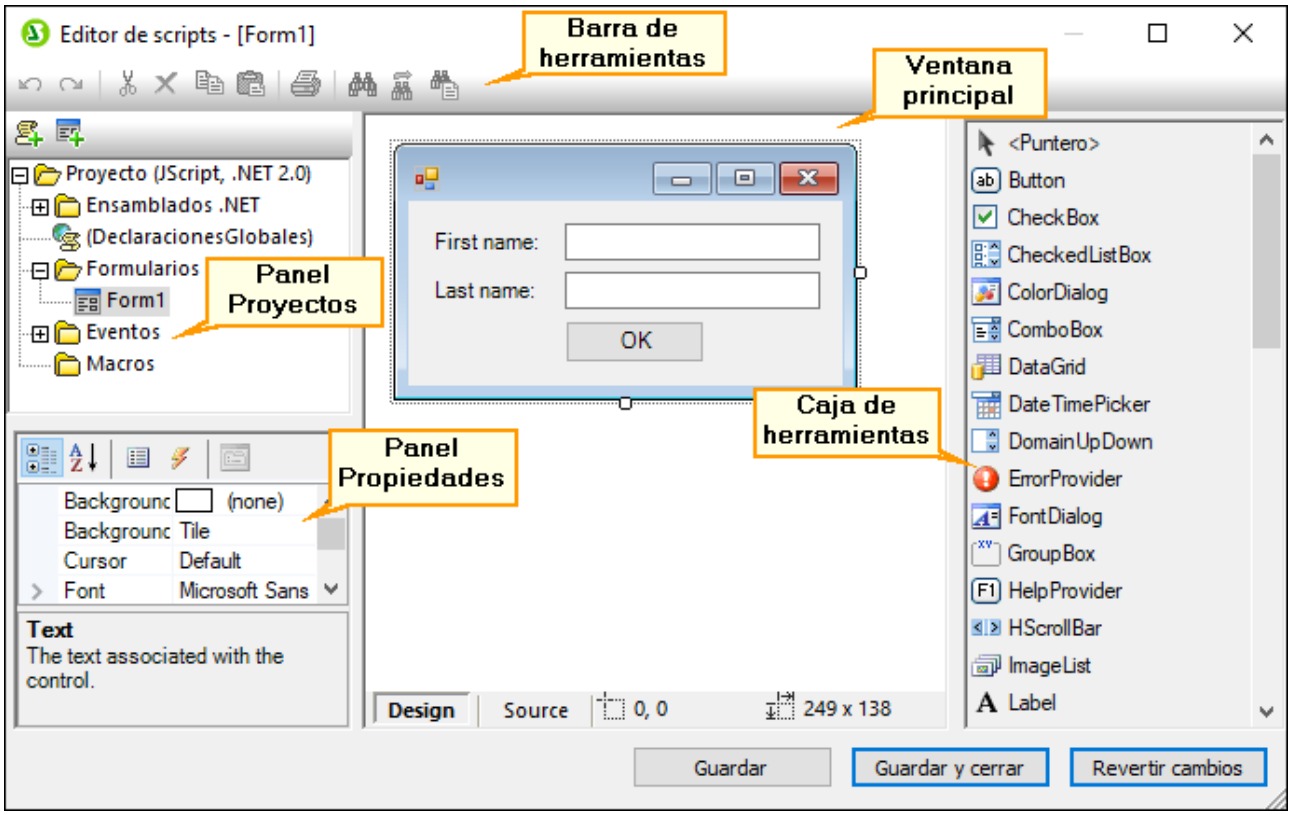
1. De forma estática, añadiéndolos manualmente al proyecto. Haga clic con el botón derecho en el panel superior izquierdo y seleccione **Agregar ensamblado .NET** en el menú contextual.
2. De forma dinámica, en tiempo de ejecución, llamando al comando `CLR.LoadAssembly` 923 desde el código.

Las secciones siguientes se centran en las partes de un proyecto de scripting: declaraciones globales, macros, formularios y eventos.Para más información sobre scripting y proyectos de scripting de ejemplo consulte Scripting con Authentic 687.

17.1.1.1 Información general

El Editor de scripts contiene estas partes:

- Barra de herramientas
- Panel Proyectos
- Panel Propiedades
- Ventana principal
- Cuadro de herramientas



El proyecto de scripting se guarda cuando guarde el archivo .sps del que el proyecto de scripting forma parte. El botón **Guardar** que se ve en la imagen anterior solamente crea una copia temporal en la memoria interna.

Barra de herramientas

La barra de herramientas incluye los comandos **Deshacer (Ctrl+Z)** y **Rehacer (Ctrl+Y)** 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 StyleVision 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 StyleVision. Las macros no tienen parámetros o valores de 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**.

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

 © 2016-2022 Altova GmbH

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.

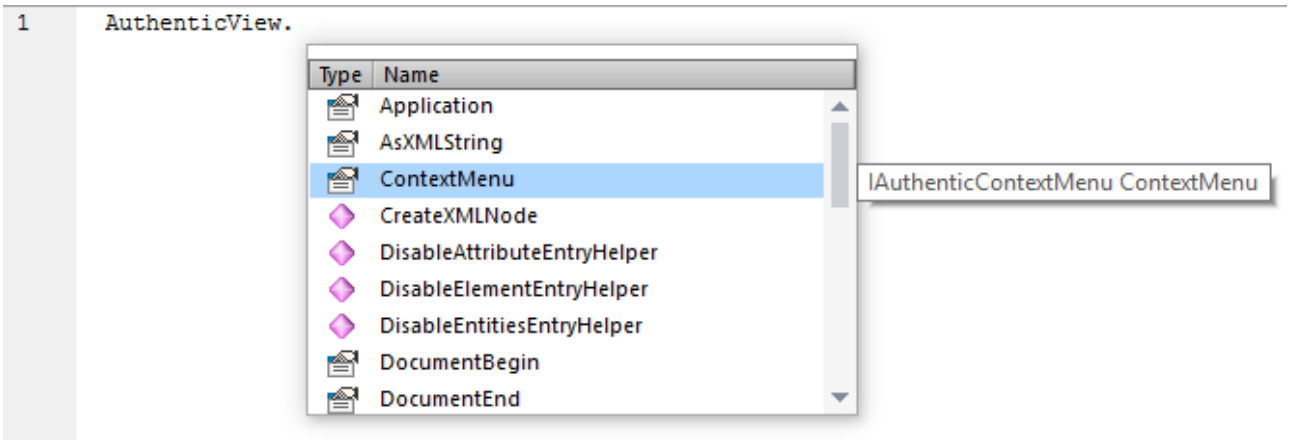
Cuando se selecciona una propiedad o un evento aparece una breve descripción de ese elemento en la parte inferior del panel Propiedades.

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 920 integrados de StyleVision. El método completo o la firma de la propiedad aparecen junto al ayudante de entrada de finalización automática.



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.

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 915.

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.

17.1.1.2 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.

 © 2016-2022 Altova GmbH

17.1.1.3 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 914).

```
// 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 920.

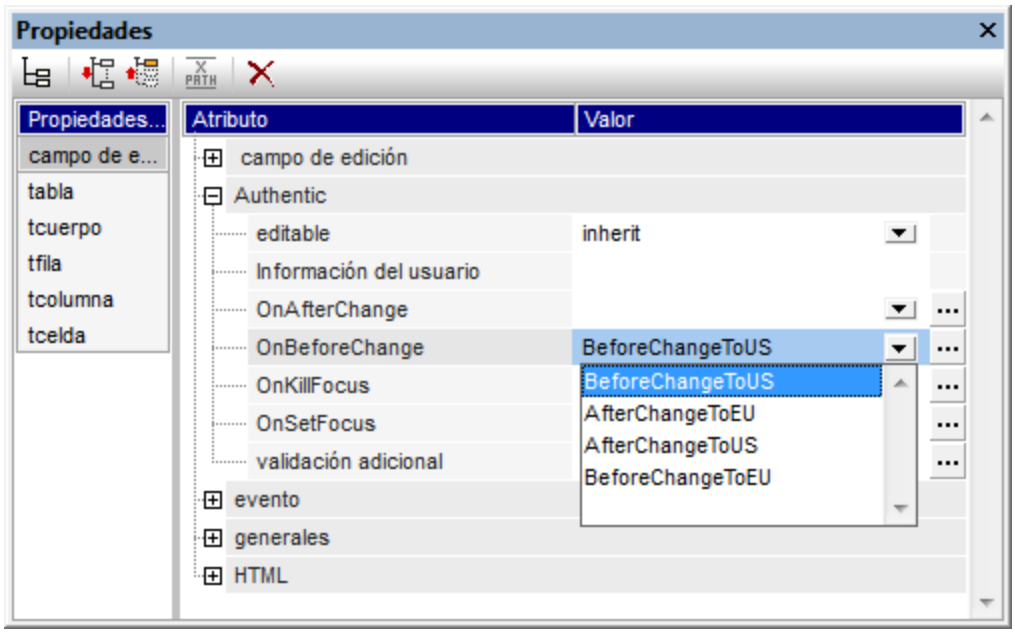
Solo se puede ejecutar una macro a la vez. Una vez se ha ejecutar la macro (o el evento) el script se cierra y las variables globales pierden sus valores.

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 .

Ejecutar macros

Para ejecutar una macro primero debe vincular la macro a un evento de Authentic. La macro se ejecutará cuando el evento de Authentic se desencadene. Por ejemplo, imagine que quiere ejecutar una macro antes de editar texto en Authentic. Estas son las instrucciones que debe seguir:

1. En la vista **Diseño** seleccione el contenido editable para el que desea crear la macro.
2. En la ventana Propiedades (*imagen siguiente*) expanda el grupo de propiedades *Authentic* del componente de diseño seleccionado.



3. Seleccione el evento Authentic que debe desencadenar la macro y abra la lista desplegable del cuadro combinado *Valor*. Esta lista incluye todas las macros definidas en el proyecto de scripting del diseño SPS. Seleccione la macro que desea vincular con esta función.
4. A partir de ahora, cuando en la vista Authentic se desencadene un evento que tenga asignada una macro, la macro se ejecutará.

Para más información consulte el apartado Scripting con Authentic >> Macros 691.

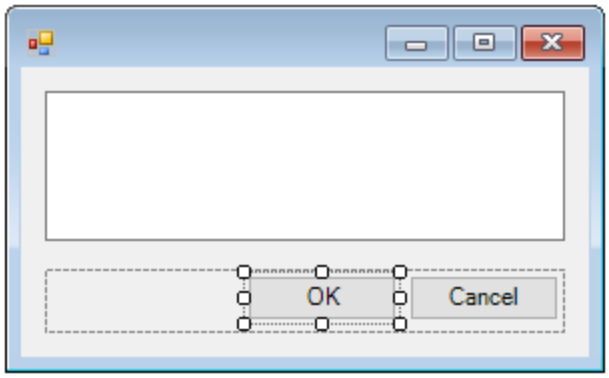
17.1.1.4 Formularios

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.

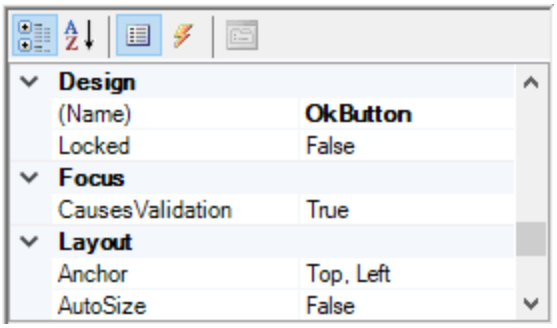
Para añadir un formulario haga clic con el botón derecho en el panel Proyecto y seleccione **Agregar formulario** en el menú contextual. Para añadir un control a un formulario arrástrelo desde la Caja de herramientas que está a la derecha del Editor de scripts y suéltela en el formulario.

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:

 © 2016-2022 Altova GmbH



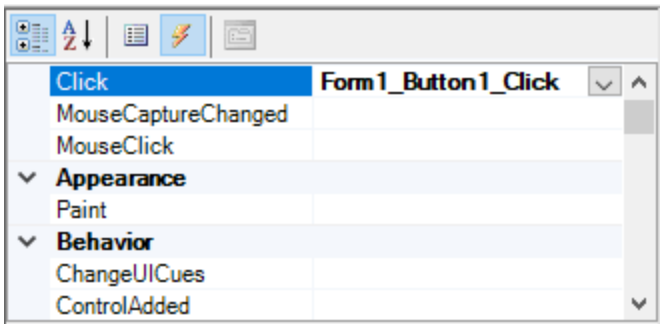
Todos los controles de los formularios tienen propiedades que puede ajustar en el panel Propiedades. Para ello, seleccione primero el control en el formulario y después edite las propiedades en cuestión en el panel Propiedades.



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 StyleVision 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:

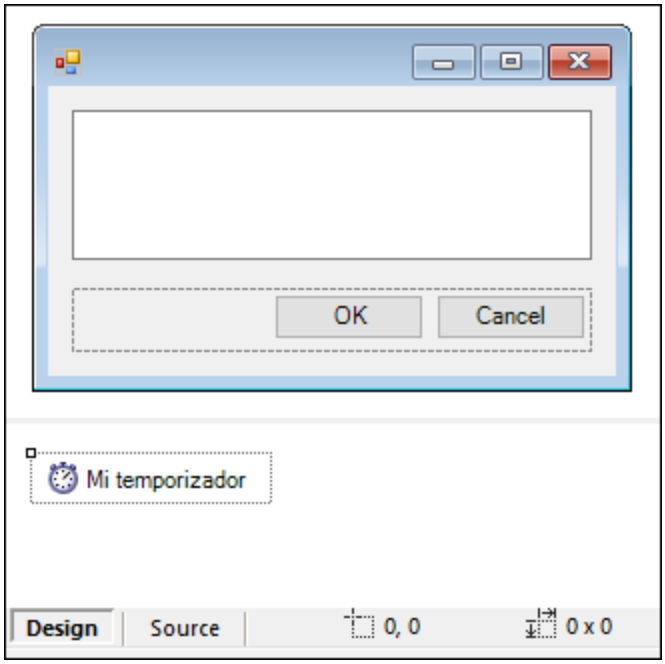
```
// MiFormulario
//   PanelBotón
//      BotónAceptar
//      BotónCancelar
//   EditorTexto
//      ReproductorMediaAx1
// ComponentesBandeja
//   MiTemporizador
```

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;
```

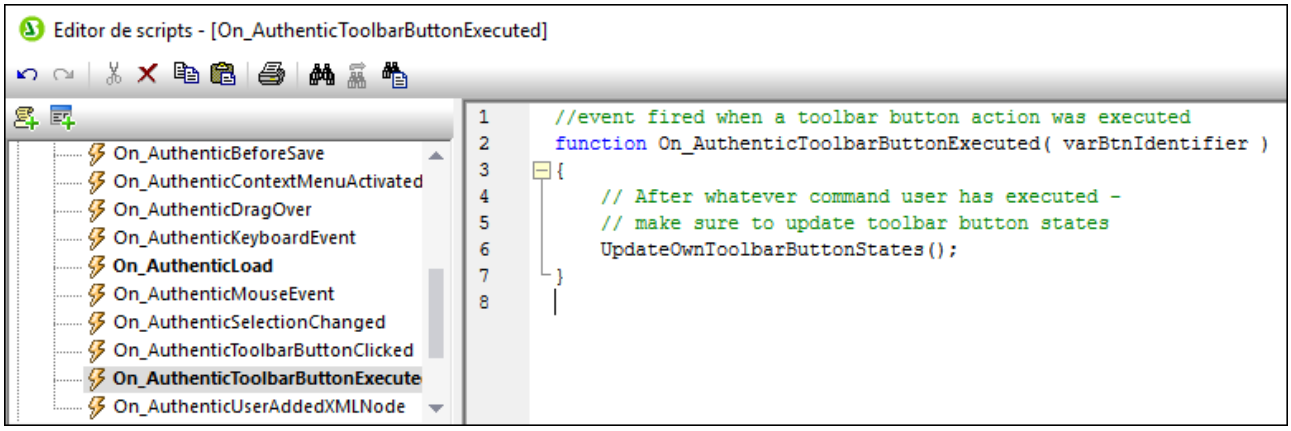
En el caso de los controles ActiveX puede acceder al objeto COM subyacente con la propiedad OCX:

```
var ocx = lastform.AxMediaPlayer1.OCX; // obtener objeto COM subyacente
ocx.enableContextMenu = true;
ocx.URL = "mms://apasf.apa.at/fm4_live_worldwide";
```

17.1.1.5 Eventos

Un proyecto de scripting puede incluir scripts que controlen eventos en la Vista Authentic, como cortar, copiar o pegar texto o abrir un menú contextual. Estos eventos los suministra la API COM de StyleVision y los puede encontrar en la carpeta "Eventos" del proyecto de scripting. Tenga en cuenta que estos eventos son eventos específicos de StyleVision, en contraposición a los eventos de formulario.

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 llama a una función cuando se ejecuta algún botón de la barra de herramientas. La definición de la función está en el script Declaraciones globales 912, por lo que está disponible para el script del controlador de eventos.



> No debe cambiar el nombre de la función del controlador de eventos; de lo contrario no se llamará al script del controlador de eventos.

También puede definir variables locales y funciones de ayuda dentro del script del controlador, por ejemplo:

```
var local;

function On_AuthenticLoad(  )
{
   local = "OnAuthenticLoad";
   Helper();
}

function Helper()
{
   alert("I'm a helper function for " + local);
}
```

17.1.1.6 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 StyleVision.

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
```


© 2016-2022 Altova GmbH

```
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 StyleVision (no necesita conocer su valor numérico) puede usar:

```
objExportXMIFileDlg.XMIType = eXMI21ForUML23;
```

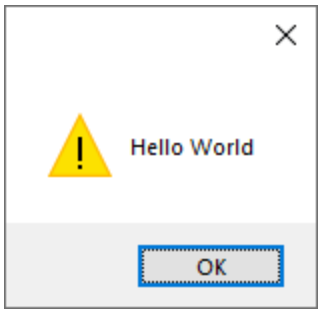
17.1.2 Comandos integrados

En este apartado se explican todos los comandos que puede usar el el Editor de scripts de StyleVision.

- alert 920
- confirm 921
- CLR.Create 922
- CLR.Import 922
- CLR.LoadAssembly 923
- CLR.ShowImports 923
- CLR.ShowLoadedAssemblies 924
- CLR.Static 925
- CreateForm 926
- doevents 927
- lastform 927
- prompt 928
- ShowForm 928
- watchdog 929

17.1.2.1 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
```

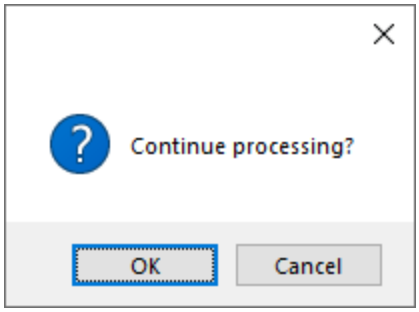
Ejemplo

Est código JScript muestra una caja de mensaje con el texto "Hello, World".

```
alert("Hello World");
```

17.1.2.2 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
```

17.1.2.3 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>
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 > >" );
```

17.1.2.4 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 923.

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 );
```

 © 2016-2022 Altova GmbH

```
}
```

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 );
}
```

17.1.2.5 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" );
}
```

17.1.2.6 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.

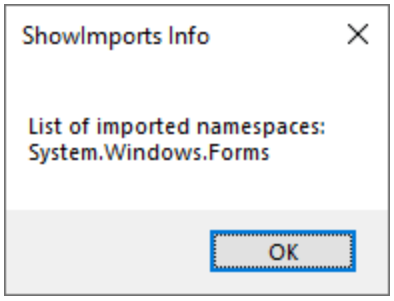
Firma

```
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();
```



17.1.2.7 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.

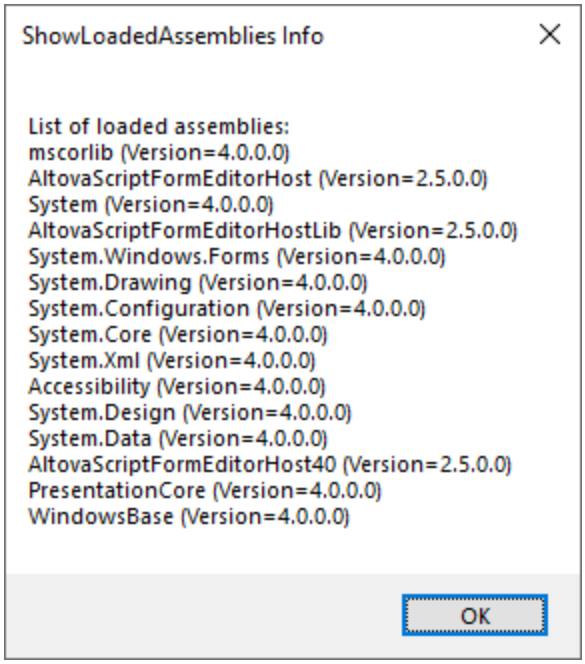
Firma

```
CLR.ShowLoadedAssemblies() -> void
```

Ejemplo

```
CLR.ShowLoadedAssemblies();
```


© 2016-2022 Altova GmbH



17.1.2.8 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" );
```

17.1.2.9 CreateForm

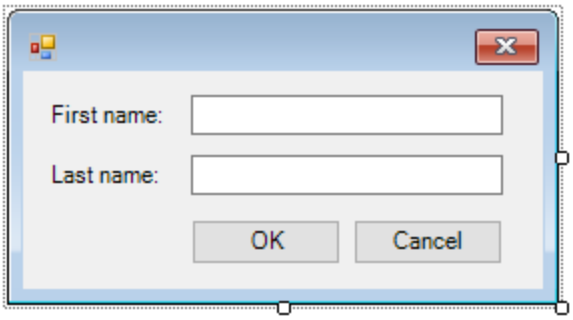
Instancia el objeto formulario `Form` identificado por el nombre dado como argumento. El formulario debe existir en la carpeta "Formularios" del proyecto de scripting. Devuelve el objeto formulario (`System.Windows.Forms.Form`) que corresponde al nombre dato o `null` si no existe ningún formulario con ese nombre.

Firma

```
CreateForm (strFormName : String) -> System.Windows.Forms.Form | null
```

Ejemplo

Imagine que en el proyecto de scripting existe un formulario llamado "FormName".



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").

© 2016-2022 Altova GmbH

17.1.2.10 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
}
```

17.1.2.11 lastform

Este es un campo global que devuelve una referencia al último objeto formulario que se creó con `CreateForm()` o `ShowForm()`.

Firma

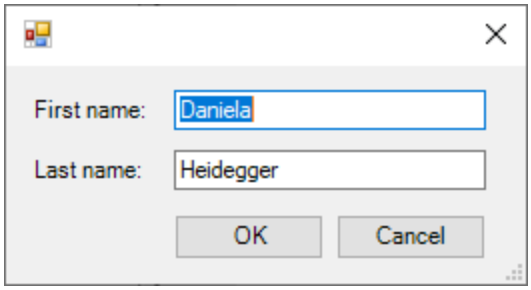
```
lastform -> formObj : System.Windows.Forms.Form
```

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`.



17.1.2.12 prompt

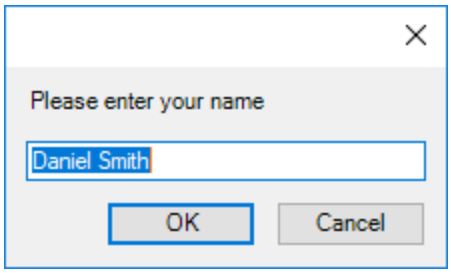
Abre un cuadro de diálogo en el que aparece un mensaje y un control de campo de texto con una respuesta predeterminada. Esto permite al usuario introducir un valor simple de cadena de texto. El valor de retorno es una cadena que contiene el valor del campo de texto o `null` si el usuario seleccionó "Cancelar".

Firma

```
prompt(strMessage : String, strDefault : String) -> val : String
```

Ejemplo

```
var name = prompt( "Please enter your name", "Daniel Smith" );
if ( name != null )
    alert( "¡Hola, " + name + "!" );
```



17.1.2.13 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).

© 2016-2022 Altova GmbH

Firma

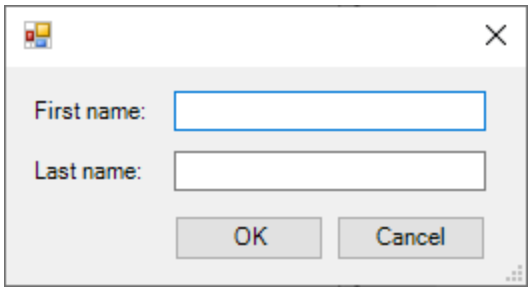
```
ShowForm(strFormName : String) -> result : Integer
```

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").

17.1.2.14 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
```


© 2016-2022 Altova GmbH

17.2 Application API

The COM-based API of StyleVision (also called the Application API from now on) enables other applications to use the functionality of StyleVision.

StyleVision 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 931 and in the Overview 932 part of this section)
- From within the built-in Scripting Editor of StyleVision. For a description of the scripting environment, see the section, Scripting Editor 907.
- Via an ActiveX Control, which is available if the integration package 1051 is installed. For more information, see the section ActiveX Integration 1051.

External programs

In the Overview 932 part of this section, we describe how the functionality of StyleVision can be accessed and automated from external programs.

Using the Application API from outside StyleVision requires an instance of StyleVision to be started first. How this is done depends on the programming language used. See the section, Programming Languages 932, for information about individual languages.

Essentially, StyleVision will be started via its COM registration. Then the `Application` object associated with the StyleVision instance is returned. Depending on the COM settings, an object associated with an already running StyleVision 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 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# 936 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 945: 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.

Programming points

The following limitations must be considered in your client code:

- Be aware that if your client code crashes, instances of StyleVision 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.

17.2.1 Overview

This overview of the Application API is organized as follows:

- The Object Model 932 describes the relationships between the objects of the Application API.
- Programming Languages 932 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.

17.2.1.1 Object Model

The starting point for every application which uses the Application API is the `Application` 952 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 932 section.

The chart below shows the links between the main objects of the Application API

```
Application
|
| -- Documents
|     |
|     | -- Document
|     |     |
|     |     | -- Methods for assigning working XML file, saving files
|     |     | -- Methods for saving generated files
|     |     | -- Methods for activating, saving, closing SPS file
```

In addition there are objects for schema sources and properties, as well as methods for AuthenticView events.

Once you have created an `Application` object you can start using the functionality of StyleVision.

17.2.1.2 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:

 © 2016-2022 Altova GmbH

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Users\<usuario>\Documentos\Altova\StyleVision\2023\%APPNAME%>Examples`

JScript

The JScript listings demonstrate the following basic functionality:

- Start application or attach to a running instance 934
- Simple document access 935
- Iteration 936

VBScript

VBScript is different than JScript only syntactically; otherwise it works in the same way. For more information, refer to the JScript examples 933.

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 StyleVision 941: Starts StyleVision, which is registered as an automation server, or activates StyleVision if it is already running.
- Open OrgChart.pxf 942: Locates one of the example documents installed with StyleVision and opens it. If this document is already open it becomes the active document.
- OnDocumentOpened Event On/Off 943: Shows how to listen to StyleVision events. When turned on, a message box will pop up after a document has been opened.
- Open ExpReport.xml 942: Opens another example document.
- Shutdown StyleVision 941: Stops StyleVision.

Java

The StyleVision API can be accessed from Java code. The Java sub-section of this section 945 explains how some basic StyleVision functionality can be accessed from Java code. It is organized into the following sub-sections:

- Mapping Rules for the Java Wrapper 945
- Example Java Project 946
- Application Startup and Shutdown 950
- Simple Document Access 951
- Iterations 951
- Event Handlers 951

17.2.1.2.1 JScript

This section contains listings of JScript code that demonstrate the following basic functionality:

- Start application or attach to a running instance 934

- Simple document access 935
- Iteration 936

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\StyleVision\2023\%APPNAME%>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).

17.2.1.2.1.1 Start Application

The JScript below starts the application and shuts it down. If the COM object of the 32-bit StyleVision 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 StyleVision, el nombre registrado o identificador programático (ProgId) del objeto COM es `StyleVision.Application`. Para la versión de 64 bits de StyleVision, el nombre es `StyleVision_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 StyleVision. Por eso, si tiene instaladas tanto la versión de 32 bits como la versión de 64 bits de StyleVision 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"
```

```
// 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.
```

 © 2016-2022 Altova GmbH

```
try {   objStyleVision = WScript.GetObject("", "StyleVision.Application");   }
catch(err) {}

if( typeof( objStyleVision ) == "undefined" )
{
   try   {   objStyleVision = WScript.GetObject("", "StyleVision_x64.Application")   }
   catch(err)
   {
      WScript.Echo( "Can't access or create StyleVision.Application" );
      WScript.Quit();
   }
}

// if newly started, the application will start without its UI visible. Set it to
visible.
objStyleVision.Visible = true;

WScript.Echo(objStyleVision.Edition + " has successfully started. ");

objStyleVision.Visible = false; // will shutdown application if it has no more COM
connections
//objStyleVision.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 934).

17.2.1.2.1.2 Simple Document Access

After you have started the application as shown in Start Application 934, 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 StyleVision Examples folder.

```
// Locate examples via USERPROFILE shell variable. The path needs to be adapted to major
release versions.
objWshShell = WScript.CreateObject("WScript.Shell");
majorVersionYear = objStyleVision.MajorVersion + 1998
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\My
Documents\\Altova\\StyleVision" + majorVersionYear + "\\StyleVisionExamples\\";

// Tell StyleVision to open two documents. No dialogs
objDoc1 = objStyleVision.Documents.OpenDocument(strExampleFolder + "OrgChart.pxf");
objStyleVision.Documents.OpenDocument(strExampleFolder +
"BiggestCitiesPerContinent.sps");
```

The JScript code listed above is available in the sample file `DocumentAccess.js` (see Example Files 934).

17.2.1.2.1.3 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
for (var iterDocs = new Enumerator(objStyleVision.Documents); !iterDocs.atEnd();
iterDocs.moveNext())
{
   objName = iterDocs.item().Name;
   WScript.Echo("Document name: " + objName);
}

// go through all open documents using index-based access to the document collection
for (i = objStyleVision.Documents.Count; i > 0; i--)
   objStyleVision.Documents.Item(i).Close();
```

The JScript code listed above is available in the sample file `DocumentAccess.js` (see Example Files 934).

17.2.1.2.2 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 StyleVision Type Library by clicking **Project | Add Reference**. The Add Reference dialog appears. Browse for the StyleVision Type Library component, which is located in the StyleVision application folder, and add it.
3. Enter the code you want.
4. Compile the code and run it.

Example C# project

Your StyleVision 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\StyleVision\2023\%APPNAME%>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.

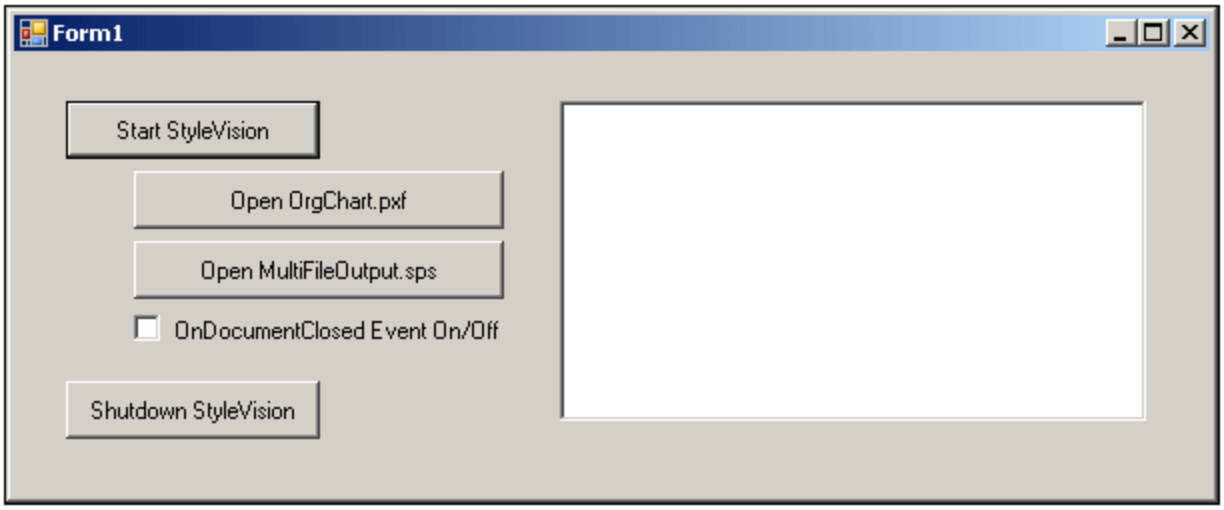
 © 2016-2022 Altova GmbH

Platform configuration

If you have a 64-bit operating system and are using a 32-bit installation of StyleVision, 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 StyleVision operations:



- Start StyleVision 941: Starts StyleVision, which is registered as an automation server, or activates the application if it is already running.
- Open OrgChart.pxf 942: Locates one of the example documents installed with StyleVision and opens it. If this document is already open it becomes the active document.
- Open MultiFileOutput.sps 942: Opens another example document.
- Shut down StyleVision 941: Stops StyleVision.

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 `AutomateStyleVision_VS2008.sln` or the file `AutomateStyleVision_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 `AutomateStyleVision` 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.

```
namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        // An instance of StyleVision is accessed via its automation interface.
        StyleVisionLib.Application StyleVision;

        // Location of examples installed with StyleVision
        String strExamplesFolder;

        private void Form1_Load(object sender, EventArgs e)
        {
            // Locate examples installed with StyleVision.
            // REMARK: You might need to adapt this if you have a different major version
of the product.
            strExamplesFolder = Environment.GetEnvironmentVariable("USERPROFILE") + "\\My
Documents\\Altova\\StyleVision2012\\StyleVisionExamples\\";
        }

        // Handler for the "Start StyleVision" button
        private void StartStyleVision_Click(object sender, EventArgs e)
        {
            if (StyleVision == null)
            {
                Cursor.Current = Cursors.WaitCursor;

                // If there is no StyleVision instance, create one and make it visible.
                StyleVision = new StyleVisionLib.Application();
                StyleVision.Visible = true;

                Cursor.Current = Cursors.Default;
            }
            else
            {
                // If a StyleVision instance is already running, make sure it's visible.
                if (!StyleVision.Visible)
                    StyleVision.Visible = true;
            }
        }

        // Handler for the "Open OrgChart.pxf" button
        private void openOrgChart_Click(object sender, EventArgs e)
        {
            // Make sure there's a running StyleVision instance, and that it's visible
            StartStyleVision_Click(null, null);
```

 © 2016-2022 Altova GmbH

```
            // Open a sample files installed with the product.
            StyleVision.Documents.OpenDocument(strExamplesFolder + "OrgChart.pxf");
            updateListBox();
        }

        // Handler for the "Open MultiFileOutput.sps" button
        private void openMultiFileOutput_Click(object sender, EventArgs e)
        {
            // Make sure there's a running StyleVision instance, and that it's visible
            StartStyleVision_Click(null, null);

            // Open one of the sample files installed with the product.
            StyleVision.Documents.OpenDocument(strExamplesFolder + "MultiFileOutput.sps");
            updateListBox();
        }

        // Handler for the "Shutdown StyleVision" button
        // Shut down the application instance by explicitly releasing the COM object.
        private void shutdownStyleVision_Click(object sender, EventArgs e)
        {
            if (StyleVision != null)
            {
                // Allow shut-down of StyleVision by releasing UI
                StyleVision.Visible = false;

                // Explicitly release the COM object
                try
                {
                    int i =
System.Runtime.InteropServices.Marshal.ReleaseComObject(StyleVision);
                    while
(System.Runtime.InteropServices.Marshal.ReleaseComObject(StyleVision) > 0) ;
                }
                finally
                {
                    // Disallow subsequent access to this object.
                    StyleVision = null;
                }
            }
        }

        delegate void addListBoxItem_delegate(string sText);
        // Called from the UI thread
        private void addListBoxItem(string sText)
        {
            listBoxMessages.Items.Add(sText);
        }
        // Wrapper method to call UI control methods from a worker thread
        void syncWithUIthread(Control ctrl, addListBoxItem_delegate methodToInvoke, String
sText)
        {
            // Control.Invoke: Executes on the UI thread, but calling thread waits for
completion before continuing.
            // Control.BeginInvoke: Executes on the UI thread, and calling thread doesn't
wait for completion.
            if (ctrl.InvokeRequired)
```

```
                ctrl.BeginInvoke(methodToInvoke, new Object[] { sText });
        }

        // Event handler for OnDocumentClosed event
        private void handleOnDocumentClosed(StyleVisionLib.Document i_ipDocument)
        {
            String sText = "";

            if (i_ipDocument.Name.Length > 0)
                sText = "Document " + i_ipDocument.Name + " was closed!";

            // Synchronize the calling thread with the UI thread because
            // COM events are triggered from a working thread
            addListBoxItem_delegate methodToInvoke = new
addListBoxItem_delegate(addListBoxItem);
            // Call syncWithUIthread with the following arguments:
            // 1 - listBoxMessages - list box control to display messages from COM events
            // 2 - methodToInvoke  - a C# delegate which points to the method which will be
called from the UI thread
            // 3 - sText           - the text to be displayed in the list box
            syncWithUIthread(listBoxMessages, methodToInvoke, sText);
        }



        private void updateListBox()
        {
            // Iterate through all open documents
            listBoxMessages.Items.Clear();


            for (int i = 1; i <= StyleVision.Documents.Count; i++)
            {
                StyleVisionLib.Document doc = StyleVision.Documents[i];

                if (doc != null)
                {
                    if (checkBoxEventOnOff.Checked)
                        doc.OnDocumentClosed += new
StyleVisionLib._IDocumentEvents_OnDocumentClosedEventHandler(handleOnDocumentClosed);
                    else
                        doc.OnDocumentClosed -= new
StyleVisionLib._IDocumentEvents_OnDocumentClosedEventHandler(handleOnDocumentClosed);

                    listBoxMessages.Items.Add(doc.Name);
                    StyleVisionLib.SchemaSources sources = doc.SchemaSources;

                    for (int j = 1; j <= sources.Count; j++)
                    {
                        StyleVisionLib.SchemaSource source = sources[j];

                        if (source != null)
                        {
                            listBoxMessages.Items.Add("\tSchema      file name : " +
source.SchemaFileName + "\n");
```

```
                            listBoxMessages.Items.Add("\tWorking XML file name : " +
source.WorkingXMLFileName + "\n");
                            listBoxMessages.Items.Add("\tIs main schema source : " +
source.IsMainSchemaSource + "\tType name : " + source.TypeName + "\n");
                        }
                    }
                }
            }
        }
    }
}
```

17.2.1.2.2.1 Add Reference to StyleVision 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 `StyleVision.tlb`, and is located in the StyleVision application folder.

Then declare a variable to access the StyleVision API:

```
// An instance of StyleVision is accessed via its automation interface.
   StyleVisionLib.Application StyleVision;
```

17.2.1.2.2.2 Application Startup and Shutdown

In the code snippets below, the methods `StartStyleVision_Click` and `ShutdownStyleVision_Click` are those assigned to buttons in the AutomateStyleVision example 936 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\StyleVision\2023\%APPNAME%>Examples`

You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022.

Starting StyleVision

The following code snippet from the AutomateStyleVision example 936 shows how to start up the application.

```
// Handler for the "Start StyleVision" button
private void StartStyleVision_Click(object sender, EventArgs e)
{
    if (StyleVision == null)
    {
        Cursor.Current = Cursors.WaitCursor;

        // If there is no StyleVision instance, create one and make it visible.
        StyleVision = new StyleVisionLib.Application();
```

```
                StyleVision.Visible = true;

                Cursor.Current = Cursors.Default;
            }
            else
            {
                // If an instance of StyleVision is already running, make sure it's visible
                if (!StyleVision.Visible)
                    StyleVision.Visible = true;
            }
        }
```

Shutting down StyleVision

The following code snippet from the AutomateStyleVision example 936 shows how to shut down the application.

```
        // Handler for the "Shutdown StyleVision" button
        // Shut down the application instance by explicitly releasing the COM object.
        private void shutdownStyleVision_Click(object sender, EventArgs e)
        {
            if (StyleVision != null)
            {
                // Allow shut-down of StyleVision by releasing UI
                StyleVision.Visible = false;

                // Explicitly release the COM object
                try
                {
                    int i =
System.Runtime.InteropServices.Marshal.ReleaseComObject(StyleVision);
                    while
(System.Runtime.InteropServices.Marshal.ReleaseComObject(StyleVision) > 0) ;
                }
                finally
                {
                    // Disallow subsequent access to this object.
                    StyleVision = null;
                }
            }
        }
```

17.2.1.2.2.3 Opening Documents

The code snippets below (from the AutomateStyleVision example 936) 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: Documents.OpenDocument(string) 1026.

The AutomateStyleVision example 936 (*see the file Form1.cs*) is located in the C# subfolder of the API Examples folder:

Windows 7, Windows 8,	`C:\Users\<usuario>\Documentos\`

Windows 10, Windows 11	`Altova\StyleVision\2023\%APPNAME%>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 StyleVision instance, and that it's visible
     StartStyleVision_Click(null, null);

     // Open a sample files installed with the product.
     StyleVision.Documents.OpenDocument(strExamplesFolder + "OrgChart.pxf");
     updateListBox();
 }

 // Handler for the "Open MultiFileOutput.sps" button
 private void openMultiFileOutput_Click(object sender, EventArgs e)
 {
     if (StyleVision == null)
         StartStyleVision_Click(null, null);

     // Open one of the sample files installed with the product.
     StyleVision.Documents.OpenDocument(strExamplesFolder + "MultiFileOutput.sps");
     updateListBox();
 }
```

The file opened last will be the active file.

17.2.1.2.2.4 Events

The code snippet below (from the AutomateStyleVision example [936]) lists the code for two event handlers. The AutomateStyleVision example [936] (*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\StyleVision\2023\%APPNAME%>Examples`

You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022.

Code snippet

```
delegate void addListBoxItem_delegate(string sText);
// Called from the UI thread
private void addListBoxItem(string sText)
```

```
        {
            listBoxMessages.Items.Add(sText);
        }
        // Wrapper method to call UI control methods from a worker thread
        void syncWithUIthread(Control ctrl, addListBoxItem_delegate methodToInvoke, String
sText)
        {
            // Control.Invoke: Executes on the UI thread, but calling thread waits for
completion before continuing.
            // Control.BeginInvoke: Executes on the UI thread, and calling thread doesn't
wait for completion.
            if (ctrl.InvokeRequired)
                ctrl.BeginInvoke(methodToInvoke, new Object[] { sText });
        }

        // Event handler for OnDocumentClosed event
        private void handleOnDocumentClosed(StyleVisionLib.Document i_ipDocument)
        {
            String sText = "";

            if (i_ipDocument.Name.Length > 0)
                sText = "Document " + i_ipDocument.Name + " was closed!";

            // Synchronize the calling thread with the UI thread because
            // COM events are triggered from a working thread
            addListBoxItem_delegate methodToInvoke = new
addListBoxItem_delegate(addListBoxItem);
            // Call syncWithUIthread with the following arguments:
            // 1 - listBoxMessages - list box control to display messages from COM events
            // 2 - methodToInvoke  - a C# delegate which points to the method which will be
called from the UI thread
            // 3 - sText           - the text to be displayed in the list box
            syncWithUIthread(listBoxMessages, methodToInvoke, sText);
        }



        private void updateListBox()
        {
            // Iterate through all open documents
            listBoxMessages.Items.Clear();


            for (int i = 1; i <= StyleVision.Documents.Count; i++)
            {
                StyleVisionLib.Document doc = StyleVision.Documents[i];

                if (doc != null)
                {
                    if (checkBoxEventOnOff.Checked)
                        doc.OnDocumentClosed += new
StyleVisionLib._IDocumentEvents_OnDocumentClosedEventHandler(handleOnDocumentClosed);
                    else
                        doc.OnDocumentClosed -= new
StyleVisionLib._IDocumentEvents_OnDocumentClosedEventHandler(handleOnDocumentClosed);
```


© 2016-2022 Altova GmbH

```
                    listBoxMessages.Items.Add(doc.Name);
                    StyleVisionLib.SchemaSources sources = doc.SchemaSources;

                    for (int j = 1; j <= sources.Count; j++)
                    {
                        StyleVisionLib.SchemaSource source = sources[j];

                        if (source != null)
                        {
                            listBoxMessages.Items.Add("\tSchema      file name : " +
source.SchemaFileName + "\n");
                            listBoxMessages.Items.Add("\tWorking XML file name : " +
source.WorkingXMLFileName + "\n");
                            listBoxMessages.Items.Add("\tIs main schema source : " +
source.IsMainSchemaSource + "\tType name : " + source.TypeName + "\n");
                        }
                    }
                }
            }
        }
```

17.2.1.2.3 Java

The Application API can be accessed from Java code. To allow accessing the StyleVision automation server directly from Java code, the libraries listed below must reside in the `classpath`. They are installed in the folder: `JavaAPI` in the StyleVision 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
- `StyleVisionAPI.jar`: Java classes that wrap the StyleVision automation interface
- `StyleVisionAPI_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 946.

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 StyleVision 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 StyleVision functionality can be accessed from Java code. It is organized into the following sub-sections:

- Example Java Project 946
- Application Startup and Shutdown 950
- Simple Document Access 951
- Iterations 951
- Event Handlers 951

17.2.1.2.3.1 Example Java Project

The StyleVision 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\StyleVision\2023\%APPNAME%>Examples`

© 2016-2022 Altova GmbH

This folder contains Java examples for the StyleVision 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
`StyleVisionAPI.jar`	Java classes of the StyleVision API
`RunStyleVision.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
`StyleVision_JavaDoc.zip`	`Javadoc` file containing help documentation for the Java API

What the example does

The example starts up StyleVision and performs a few operations, including opening and closing documents. When done, StyleVision stays open. You must close it manually.

- Start StyleVision 950: Starts StyleVision, which is registered as an automation server, or activates StyleVision if it is already running.
- Open OrgChart.pxf 951: Locates one of the example documents installed with StyleVision and opens it.
- Iteration and Changing the View Mode 951: Shows how to iterate through open documents.
- Event Handling 951: Shows how to handle StyleVision events.
- Shut down StyleVision 950: Shuts down StyleVision.

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 `RunStyleVision.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

`RunStyleVision` 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 `RunStyleVision.java` is listed below with comments.

```
01 // Access general JAVA-COM bridge classes
02 import com.altova.automation.libs.*;
03
04 // Access StyleVision Java-COM bridge
05 import com.altova.automation.StyleVision.*;
06 import com.altova.automation.StyleVision.Enums.ENUMApplicationStatus;
07
08 /**
09  * A simple example that starts the COM server and performs a View operations on it.
10  * Feel free to extend.
11  */
12 public class RunStyleVision
13 {
14   public static void main(String[] args)
15   {
16     // An instance of the application.
17     Application stylevision = null;
18
19     // Instead of COM error-handling, use Java exception mechanism.
20     try
21     {
22       // Start StyleVision as COM server.
23       stylevision = new Application();
24
25       ENUMApplicationStatus status = ENUMApplicationStatus.eApplicationRunning;
26       do{
27         // Check the application status
28         status = stylevision.getStatus();
29         System.out.println("status : " + status + "\n");
30       } while (status != ENUMApplicationStatus.eApplicationRunning);
31
32       // COM servers start up invisible, so we make the server visible
33       stylevision.setVisible(true);
34
35       // Locate samples installed with the product.
36       String strExamplesFolder = System.getenv("USERPROFILE") + "\\My Documents\\Altova\
\StyleVision2012\\StyleVisionExamples\\";
37
38       // Open two files from the product samples.
39       stylevision.getDocuments().openDocument(strExamplesFolder + "OrgChart.pxf");
40       stylevision.getDocuments().openDocument(strExamplesFolder +
"BiggestCitiesPerContinent.sps");
41
42       // Iterate through all open documents
43       for (Document doc:stylevision.getDocuments())
```

 © 2016-2022 Altova GmbH

```
44        {
45          System.out.println("Document name : " + doc.getName() + "\n");
46          SchemaSources sources = doc.getSchemaSources();
47
48          for (int i = 1; i <= sources.getCount(); i++)
49          {
50            SchemaSource source = sources.getItem(i);
51            System.out.println("\tSchema      file name : " + source.getSchemaFileName() +
"\n");
52            System.out.println("\tWorking XML file name : " +
source.getWorkingXMLFileName() + "\n");
53            System.out.println("\tIs main schema source : " +
source.getIsMainSchemaSource() + "\tType name : " + source.getTypeName() + "\n");
54          }
55        }
56        // The following lines attach to the document events using a default
implementation
57        // for the events and override one of its methods.
58        // If you want to override all document events it is better to derive your
listener class
59        // from DocumentEvents and implement all methods of this interface.
60        Document doc = stylevision.getActiveDocument();
61        doc.addListener(new DocumentEventsDefaultHandler()
62        {
63          @Override
64          public void onDocumentClosed(Document i_ipDoc) throws AutomationException
65          {
66            System.out.println("Document " + i_ipDoc.getName() + " requested closing.");
67          }
68        });
69        doc.close();
70        doc = null;
71
72        System.out.println("Watch StyleVision!");
73      }
74      catch (AutomationException e)
75      {
76        e.printStackTrace();
77      }
78      finally
79      {
80        // Make sure that StyleVision can shut down properly.
81        if (stylevision != null)
82          stylevision.dispose();
83
84        // Since the COM server was made visible and still is visible, it will keep
running
85        // and needs to be closed manually.
86        System.out.println("Now close StyleVision!");
87      }
88    }
89 }
```

17.2.1.2.3.2 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 StyleVision Java-COM bridge
05 import com.altova.automation.StyleVision.*;
06 import com.altova.automation.StyleVision.Enums.ENUMApplicationStatus;
07
08 /**
09  * A simple example that starts the COM server and performs a View operations on it.
10  * Feel free to extend.
11  */
12 public class RunStyleVision
13 {
14   public static void main(String[] args)
15   {
16     // An instance of the application.
17     Application stylevision = null;
18
19     // Instead of COM error-handling, use Java exception mechanism.
20     try
21     {
22       // Start StyleVision as COM server.
23       stylevision = new Application();
24
25       ENUMApplicationStatus status = ENUMApplicationStatus.eApplicationRunning;
26       do{
27         // Check the application status
28         status = stylevision.getStatus();
29         System.out.println("status : " + status + "\n");
30       } while (status != ENUMApplicationStatus.eApplicationRunning);
31
32       // COM servers start up invisible, so we make the server visible
33       stylevision.setVisible(true);
34
35 ...
36     }
37   }
38 }
```

Application shutdown

The application can be shut down as shown below.

```
1 {
2       // Make sure that StyleVision can shut down properly.
3       if (stylevision != null)
```

 © 2016-2022 Altova GmbH

```
4         stylevision.dispose();
5
6       // Since the COM server was made visible and still is visible, it will keep running
7       // and needs to be closed manually.
8       System.out.println("Now close StyleVision!");
9 }
```

17.2.1.2.3.3 Simple Document Access

The code listing below shows how to open a document.

```
1  // Locate samples installed with the product.
2  String strExamplesFolder = System.getenv("USERPROFILE") + "\\My Documents\\Altova\
\StyleVision2012\\StyleVisionExamples\\";
3
4  // We open two files form the product samples.
5  stylevision.getDocuments().openDocument(strExamplesFolder + "OrgChart.pxf");
6  stylevision.getDocuments().openDocument(strExamplesFolder +
"BiggestCitiesPerContinent.sps");
```

17.2.1.2.3.4 Iterations

The listing below shows how to iterate through open documents.

```
01 // Iterate through all open documents
02 for (Document doc:stylevision.getDocuments())
03   {
04      System.out.println("Document name : " + doc.getName() + "\n");
05      SchemaSources sources = doc.getSchemaSources();
06
07      for (int i = 1; i <= sources.getCount(); i++)
08      {
09        SchemaSource source = sources.getItem(i);
10        System.out.println("\tSchema      file name : " + source.getSchemaFileName() +
"\n");
11        System.out.println("\tWorking XML file name : " + source.getWorkingXMLFileName()
+ "\n");
12        System.out.println("\tIs main schema source : " + source.getIsMainSchemaSource()
+ "\tType name : " + source.getTypeName() + "\n");
13      }
14   }
```

17.2.1.2.3.5 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 Document doc = stylevision.getActiveDocument();
06 doc.addListener(new DocumentEventsDefaultHandler()
07 {
08   @Override
09   public void onDocumentClosed(Document i_ipDoc) throws AutomationException
10   {
11     System.out.println("Document " + i_ipDoc.getName() + " requested closing.");
12   }
13 });
14 doc.close();
15 doc = null;
```

17.2.2 Interfaces

This chapter contains the reference of the StyleVision Type Library.

17.2.2.1 Application

Methods

Quit 956

Properties
Application 953
Parent 955

ActiveDocument 953
Documents 953

Status 956
MajorVersion 954
MinorVersion 954
Edition 954
IsAPISupported 954
ServicePackVersion 956

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

 © 2016-2022 Altova GmbH

```
Set objSpy = CreateObject("XMLSpy.Application")
```

17.2.2.1.1 Events

17.2.2.1.1.1 OnShutDown

Event: OnShutDown()

Description
Sent just before StyleVision shuts down.

17.2.2.1.2 ActiveDocument

Property: ActiveDocument as Document 1012

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.

17.2.2.1.3 Application

Property: Application as Application 952 (read-only)

Description
Accesses the StyleVision application object.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.1.4 Documents

Property: Documents as Documents 1024

Description
Collection of all open documents.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.1.5 Edition

Property: Edition as String

Description
Returns the edition of the application, for example `Altova StyleVision Enterprise Edition` for the Enterprise edition.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.1.6 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.

17.2.2.1.7 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.

17.2.2.1.8 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.

© 2016-2022 Altova GmbH

17.2.2.1.9 NewDocument

Method: NewDocument as Document 1012

Return Value
None

Description
Creates a new empty document based on the previous template.

Errors
1000 The application object is invalid.
1005 Error when creating a new document
1006 Cannot create document

17.2.2.1.10 OpenDocument

Method: OpenDocument(strFileName as *String*) as Document 1012

Return Value
None

Description
Opens an existing SPS file.

Errors
1000 The application object is invalid.
1002 Invalid file extension.
1003 Error when opening document.
1004 Cannot open document.

17.2.2.1.11 Parent

Property: Parent as Application 952 (read-only)

Description
Accesses the StyleVision application object.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.1.12 Quit

Method: Quit()

Return Value
None

Description
This method terminates StyleVision. All modified documents will be closed without saving the changes. This is also true for an open project.

If StyleVision 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.

17.2.2.1.13 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.

17.2.2.1.14 Status

Property: Status as ENUMApplicationStatus 1045

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.

17.2.2.1.15 Visible

Property: Visible as VARIANT_BOOL

Description
Sets or gets the visibility attribute of StyleVision.

© 2016-2022 Altova GmbH

Errors

1110 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.2 AppOutputLine

Represents a message line. Its structure is more detailed and can contain a collection of child lines, thereby forming a tree of message lines.

Properties and Methods

Properties to navigate the object model:

Application 958
Parent 961

Line access:

GetLineSeverity 960
GetLineSymbol 960
GetLineText 960
GetLineTextEx 961
GetLineTextWithChildren 961
GetLineTextWithChildrenEx 961

A single AppOutputLine consists of one or more sub-lines.
Sub-line access:

GetLineCount 960

A sub-line consists of one or more cells.
Cell access:

GetCellCountInLine 958
GetCellIcon 958
GetCellSymbol 960
GetCellText 959
GetCellTextDecoration 959
GetIsCellText 959

Below an AppOutputLine there can be zero, one, or more child lines which themselves are of type AppOutputLine, which thus form a tree structure.

Child lines access:

ChildLines 958

17.2.2.2.1 Application

Property: Application as Application 952 (read-only)

Description
Retrieves the application's top-level object.

Errors
4100 The object is no longer valid.
4101 Invalid address for the return parameter was specified.

17.2.2.2.2 ChildLines

Property: ChildLines as AppOutputLines 962 (read-only)

Description
Returns a collection of the current line's direct child lines.

Errors
4100 The application object is no longer valid.
4101 Invalid address for the return parameter was specified.

17.2.2.2.3 GetCellCountInLine

Method: GetCellCountInLine (*nLine* as Long) as Long

Description
Gets the number of cells in the sub-line indicated by *nLine* in the current AppOutputLine.

Errors
4100 The object is no longer valid.
4101 Invalid address for the return parameter was specified.

17.2.2.2.4 GetCellIcon

Method: GetCellIcon (*nLine* as Long, *nCell* as Long) as Long

Description
Gets the icon of the cell indicated by *nCell* in the current AppOutputLine's sub-line indicated by *nLine*.

Errors
4100 The object is no longer valid.
4101 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

17.2.2.2.5 GetCellSymbol

Method: GetCellSymbol (*nLine* as Long, *nCell* as Long) as AppOutputLineSymbol 963

Description
Gets the symbol of the cell indicated by *nCell* in the current AppOutputLine's sub-line indicated by *nLine*.

Errors
4100 The object is no longer valid.
4101 Invalid address for the return parameter was specified.

17.2.2.2.6 GetCellText

Method: GetCellText (*nLine* as Long, *nCell* as Long) as String

Description
Gets the text of the cell indicated by *nCell* in the current AppOutputLine's sub-line indicated by *nLine*.

Errors
4100 The object is no longer valid.
4101 Invalid address for the return parameter was specified.

17.2.2.2.7 GetCellTextDecoration

Method: GetCellTextDecoration (*nLine* as Long, *nCell* as Long) as Long

Description
Gets the decoration of the text cell indicated by *nCell* in the current AppOutputLine's sub-line indicated by *nLine*.
It can be one of the ENUMAppOutputLine_TextDecoration 1046 values.

Errors
4100 The object is no longer valid.
4101 Invalid address for the return parameter was specified.

17.2.2.2.8 GetIsCellText

Method: GetIsCellText (*nLine* as Long, *nCell* as Long) as String

Description
Gets the text of the cell indicated by *nCell* in the current AppOutputLine's sub-line indicated by *nLine*.

Errors
4100 The object is no longer valid.

4101 Invalid address for the return parameter was specified.

17.2.2.2.9 GetLineCount

Method: GetLineCount () as Long

Description
Gets the number of sub-lines the current line consists of.

Errors
4100 The object is no longer valid.
4101 Invalid address for the return parameter was specified.

17.2.2.2.10 GetLineSeverity

Method: GetLineSeverity () as Long

Description
Gets the severity of the line. It can be one of the ENUMAppOutputLine_Severity 1045 values:

Errors
4100 The object is no longer valid.
4101 Invalid address for the return parameter was specified.

17.2.2.2.11 GetLineSymbol

Method: GetLineSymbol () as AppOutputLineSymbol 963

Description
Gets the symbol assigned to the whole line.

Errors
4100 The object is no longer valid.
4101 Invalid address for the return parameter was specified.

17.2.2.2.12 GetLineText

Method: GetLineText () as String

Description
Gets the contents of the line as text.

Errors

 © 2016-2022 Altova GmbH

4100 The object is no longer valid.
4101 Invalid address for the return parameter was specified.

17.2.2.2.13 GetLineTextEx

Method: GetLineTextEx (*psTextPartSeperator* as String, *psLineSeperator* as String) as String

Description
Gets the contents of the line as text using the specified part and line separators.

Errors
4100 The object is no longer valid.
4101 Invalid address for the return parameter was specified.

17.2.2.2.14 GetLineTextWithChildren

Method: GetLineTextWithChildren () as String

Description
Gets the contents of the line including all child and descendant lines as text.

Errors
4100 The object is no longer valid.
4101 Invalid address for the return parameter was specified.

17.2.2.2.15 GetLineTextWithChildrenEx

Method: GetLineTextWithChildrenEx (*psPartSep* as String, *psLineSep* as String, *psTabSep* as String, *psItemSep* as String) as String

Description
Gets the contents of the line including all child and descendant lines as text using the specified part, line, tab and item separators.

Errors
4100 The object is no longer valid.
4101 Invalid address for the return parameter was specified.

17.2.2.2.16 Parent

Property: Parent as AppOutputLines 962 (read-only)

Description

The parent object according to the object model.

Errors

4100 The object is no longer valid.
4101 Invalid address for the return parameter was specified.

17.2.2.3 AppOutputLines

Represents a collection of `AppOutputLine` message lines.

Properties and Methods

Properties to navigate the object model:

`Application` 962
`Parent` 963

Iterating through the collection:

`Count` 962
`Item` 963

17.2.2.3.1 Application

Property: Application as Application 952 (read-only)

Description

Retrieves the application's top-level object.

Errors

4000 The object is no longer valid.
4001 Invalid address for the return parameter was specified.

17.2.2.3.2 Count

Property: Count as Integer (read-only)

Description

Retrieves the number of lines in the collection.

Errors

4000 The object is no longer valid.
4001 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

17.2.2.3.3 Item

Property: Item (*nIndex* as Integer) as AppOutputLine 957 (read-only)

Description
Retrieves the line at nIndex from the collection. Indices start with 1.

Errors
4000 The object is no longer valid.
4001 Invalid address for the return parameter was specified.

17.2.2.3.4 Parent

Property: Parent as AppOutputLine 957 (read-only)

Description
The parent object according to the object model.

Errors
4000 The object is no longer valid.
4001 Invalid address for the return parameter was specified.

17.2.2.4 AppOutputLineSymbol

An `AppOutputLineSymbol` represents a link in an `AppOutputLine` message line which can be clicked in the StyleVision Messages window. It is applied to a cell of an `AppOutputLine` or to the whole line itself.

Properties and Methods
Properties to navigate the object model:

`Application` 963
`Parent` 965

Access to AppOutputLineSymbol methods:

`GetSymbolHREF` 964
`GetSymbolID` 964
`IsSymbolHREF` 964

17.2.2.4.1 Application

Property: Application as Application 952 (read-only)

Description
Retrieves the application's top-level object.

Errors
4200 The object is no longer valid.
4201 Invalid address for the return parameter was specified.

17.2.2.4.2 GetSymbolHREF

Method: GetSymbolHREF () as String

Description
If the symbol is of type URL, returns the URL as a string.

Errors
4200 The object is no longer valid.
4201 Invalid address for the return parameter was specified.
.

17.2.2.4.3 GetSymbolID

Method: GetSymbolHREF () as Long

Description
Gets the ID of the symbol.

Errors
4200 The object is no longer valid.
4201 Invalid address for the return parameter was specified.

17.2.2.4.4 IsSymbolHREF

Method: IsSymbolHREF () as Boolean

Description
Indicates if the symbol is of kind URL.

Errors
4200 The object is no longer valid.
4201 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

17.2.2.4.5 Parent

Property: Parent as Application 952 (read-only)

Description
The parent object according to the object model.

Errors
4200 The object is no longer valid.
4201 Invalid address for the return parameter was specified.

17.2.2.5 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.

17.2.2.5.1 CountItems

Method: CountItems `()` nItems as long

Return Value
Returns the number of menu items.

Errors
2501 Invalid object.

17.2.2.5.2 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

17.2.2.5.3 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

17.2.2.5.4 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

17.2.2.5.5 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

17.2.2.6 AuthenticEventContext

The `EventContext` interface gives access to many properties of the context in which a macro is executed.

 © 2016-2022 Altova GmbH

17.2.2.6.1 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.

17.2.2.6.2 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.

17.2.2.6.3 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.

17.2.2.6.4 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

2201 Invalid object.
2202 No context.
2204 No such variable in scope
2205 Variable cannot be evaluated
2206 Variable returns sequence
2209 Invalid parameter

17.2.2.6.5 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.

© 2016-2022 Altova GmbH

17.2.2.6.6 IsAvailable

Method: IsAvailable`()` as Boolean

Return Value
Returns true if `EventContext` is set, false otherwise.

Errors
2201 Invalid object.

17.2.2.6.7 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

17.2.2.7 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 972	Clone 973	MoveBegin 990

FirstTextPosition 976	CollapsToBegin 973	MoveEnd 991
FirstXMLData 977	CollapsToEnd 974	NextCursorPosition 982
FirstXMLDataOffset 978	ExpandTo 976	PreviousCursorPosition 983
LastTextPosition 988	Goto 981	Select 993
LastXMLData 989	GotoNext 982	SelectNext 994
LastXMLDataOffset 989	GotoPrevious 983	SelectPrevious 994
Parent 992	IsEmpty 986	SetFromRange 996
	IsEqual 986	

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 997	Copy 974	AppendRow 971
	Cut 974	DeleteRow 975
	Delete 974	DuplicateRow 975
	IsCopyEnabled 985	InsertRow 985
	IsCutEnabled 985	IsFirstRow 986
	IsDeleteEnabled 986	IsInDynamicTable 987
	IsPasteEnabled 987	IsLastRow 987
	Paste 992	MoveRowDown 991
		MoveRowUp 991

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 973	GetElementAttributeValue 980	GetEntityNames 980
CanPerformAction 972	GetElementAttributeNames 979	InsertEntity 984
PerformAction 992	GetElementHierarchy 980	
	HasElementAttribute 984	
	IsTextStateApplied 988	

© 2016-2022 Altova GmbH

	SetElementAttributeValue 995	

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 1010, 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 1005, AuthenticView.DocumentEnd 1006, and AuthenticView.WholeDocument 1012. The most flexible method is AuthenticView.Goto 1008, 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 976 and LastTextPosition 988. 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 977, FirstXMLDataOffset 978, LastXMLData 989, and LastXMLDataOffset 989. 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 997 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 974, Copy 974, Paste 992 and Delete 974.
- Table operations for tables that can grow dynamically.
- Methods that map the functionality of the Authentic entry helper windows.
- Access to the XMLData 1033 objects of the underlying document to modify them directly.

17.2.2.7.1 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
```

17.2.2.7.2 Application

Property: Application as Application 952 (read-only)

Description

Accesses the StyleVision application 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.

17.2.2.7.3 CanPerformAction

Method: CanPerformAction (*eAction* as SPYAuthenticActions 1047, *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 973.

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

 © 2016-2022 Altova GmbH

See PerformAction 992.

17.2.2.7.4 CanPerformActionWith

Method: CanPerformActionWith (*eAction* as SPYAuthenticActions 1047, *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 973.

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 992.

17.2.2.7.5 Clone

Method: Clone() as AuthenticRange 969

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.

17.2.2.7.6 CollapsToBegin

Method: CollapsToBegin() as AuthenticRange 969

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.

17.2.2.7.7 CollapsToEnd

Method: CollapsToEnd() as AuthenticRange 969

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.

17.2.2.7.8 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.

17.2.2.7.9 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.

17.2.2.7.10 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.

 © 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.

17.2.2.7.11 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
```

17.2.2.7.12 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
```

17.2.2.7.13 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
2008	Internal error
2202	Missing context node
2211	XPath error

17.2.2.7.14 ExpandTo

Method: ExpandTo (*eKind* as SPYAuthenticElementKind[1047]), as AuthenticRange[969]

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.

17.2.2.7.15 FirstTextPosition

Property: FirstTextPosition as Long

Description

 © 2016-2022 Altova GmbH

Set or get the left-most text position index of the range object. This index is always less or equal to LastTextPosition[988]. 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[988], LastTextPosition[988] 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
```

17.2.2.7.16 FirstXMLData

Property: FirstXMLData as XMLData[1033]

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[978] attribute.

Whenever you set FirstXMLData to a new data object, FirstXMLDataOffset[978] 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[978] selects a position greater then the current LastXMLData[989] / LastXMLDataOffset[989], the latter gets moved to the new start position.

HINT: You can use the FirstXMLData[977] and LastXMLData[989] properties to directly access and manipulate the underlying XML document in those cases where the methods available with the AuthenticRange[969] 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
' element up.
Set objXMLData = objXMLData.Parent
MsgBox "Current selection selects element " & objXMLData.Name
```

17.2.2.7.17 FirstXMLDataOffset

Property: FirstXMLDataOffset as Long

Description

Set or get the cursor position offset inside FirstXMLData[977] element for the beginning of the range. Offset positions are based on the characters returned by the Text[997] 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.

 © 2016-2022 Altova GmbH

If FirstXMLData / FirstXMLDataOffset 978 selects a position after the current LastXMLData 989 / LastXMLDataOffset 989, 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
```

17.2.2.7.18 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 980 and SetElementAttributeValue 995.

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 995.

17.2.2.7.19 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 979, or HasElementAttribute 984.

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 995.

17.2.2.7.20 GetElementHierarchy

Method: GetElementHierarchy (*out_arrElementNames* as Variant)

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 979, 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.

17.2.2.7.21 GetEntityNames

Method: GetEntityNames (*out_arrEntityNames* as Variant)

Description

 © 2016-2022 Altova GmbH

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 984 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 980 and InsertEntity 984.

17.2.2.7.22 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

17.2.2.7.23 Goto

Method: Goto (*eKind* as SPYAuthenticElementKind 1047, *nCount* as Long, *eFrom* as SPYAuthenticDocumentPosition 1047) as AuthenticRange 969

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.
2004 Target lies before begin of document.
2005 Invalid element kind specified.
Invalid start position specified.
Invalid address for the return parameter was specified.

17.2.2.7.24 GotoNext

Method: GotoNext (*eKind* as SPYAuthenticElementKind[1047]) as AuthenticRange[969]

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
```

17.2.2.7.25 GotoNextCursorPosition

Method: GotoNextCursorPosition() as AuthenticRange[969]

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.

© 2016-2022 Altova GmbH

17.2.2.7.26 GotoPrevious

Method: GotoPrevious (*eKind* as SPYAuthenticElementKind[1047]) as AuthenticRange[969]

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
```

17.2.2.7.27 GotoPreviousCursorPosition

Method: GotoPreviousCursorPosition() as AuthenticRange[969]

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.

17.2.2.7.28 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.

17.2.2.7.29 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 980.

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
```

 © 2016-2022 Altova GmbH

17.2.2.7.30 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
```

17.2.2.7.31 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.

17.2.2.7.32 IsCutEnabled

Property: IsCutEnabled as Boolean (read-only)

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.

17.2.2.7.33 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.

17.2.2.7.34 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.

17.2.2.7.35 IsEqual

Method: IsEqual (*objCmpRange* as AuthenticRange 969) 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.

17.2.2.7.36 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.

 © 2016-2022 Altova GmbH

2005 Invalid address for the return parameter was specified.

17.2.2.7.37 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.

17.2.2.7.38 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.

17.2.2.7.39 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.

17.2.2.7.40 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.

17.2.2.7.41 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.

17.2.2.7.42 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 976. 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 976, FirstTextPosition 976 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
```

 © 2016-2022 Altova GmbH

```
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
```

17.2.2.7.43 LastXMLData

Property: LastXMLData as XMLData 1033

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 989 attribute.

Whenever you set LastXMLData to a new data object, LastXMLDataOffset 989 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 989, select a position less then the current FirstXMLData 977 / FirstXMLDataOffset 978, the latter gets moved to the new end position.

HINT: You can use the FirstXMLData 977 and LastXMLData 989 properties to directly access and manipulate the underlying XML document in those cases, where the methods available with the AuthenticRange 969 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.

17.2.2.7.44 LastXMLDataOffset

Property: LastXMLDataOffset as Long

Description
Set or get the cursor position inside LastXMLData 989 element for the end of the range.

Offset positions are based on the characters returned by the Text 997 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 989 / LastXMLDataOffset 989 selects a position before FirstXMLData 977 / FirstXMLDataOffset 978 , 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
```

17.2.2.7.45 MoveBegin

Method: MoveBegin (*eKind* as SPYAuthenticElementKind 1047, *nCount* as Long) as AuthenticRange 969

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.

 © 2016-2022 Altova GmbH

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.

17.2.2.7.46 MoveEnd

Method: MoveEnd (*eKind* as SPYAuthenticElementKind[1047], *nCount* as Long) as AuthenticRange[969]

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.

17.2.2.7.47 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

2001 The authentic range object or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

17.2.2.7.48 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.

17.2.2.7.49 Parent

Property: Parent as AuthenticView 997 (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.

17.2.2.7.50 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.

17.2.2.7.51 PerformAction

Method: PerformAction (*eAction* as SPYAuthenticActions 1047, *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 973.

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

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
```

17.2.2.7.52 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
' 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()
```

17.2.2.7.53 SelectNext

Method: SelectNext (*eKind* as SPYAuthenticElementKind[1047]) as AuthenticRange[969]

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
```

17.2.2.7.54 SelectPrevious

Method: GotoPrevious (*eKind* as SPYAuthenticElementKind[1047]) as AuthenticRange[969]

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.

© 2016-2022 Altova GmbH

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
```

17.2.2.7.55 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 979, or HasElementAttribute 984.

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 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
```

17.2.2.7.56 SetFromRange

Method: SetFromRange (*objSrcRange* as AuthenticRange[969])

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.

17.2.2.7.57 SetVariableValue

Method: SetVariableValue`(strName as string, strValue as string)`

Return Value
Sets the value (second parameter) of the variable named in the first parameter.

© 2016-2022 Altova GmbH

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

17.2.2.7.58 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 974, Delete 974 or PerformAction 992 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.

17.2.2.8 AuthenticView

Properties

Application 1004
AsXMLString 1004
DocumentBegin 1005
DocumentEnd 1006
Event 1007
MarkupVisibility 1009
Parent 1009
Selection 1010
XMLDataRoot 1012
WholeDocument 1012

Methods

Goto 1008
IsRedoEnabled 1008
IsUndoEnabled 1009
Print 1009
Redo 1010
Undo 1011
UpdateXMLInstanceEntities 1011

Events

OnBeforeCopy 998
OnBeforeCut 998
OnBeforeDelete 999
OnBeforeDrop 999
OnBeforePaste 1000

OnMouseEvent 1001
OnSelectionChanged 1001

Description

AuthenticView and its child objects AuthenticRange[969] 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[969])
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[969] to make navigation inside the document straight-forward, and to allow for the flexible selection of logical text elements. Use the properties DocumentBegin[1005], DocumentEnd[1006], or WholeDocument[1012] for simple selections, while using the Goto[1008] method for more complex selections. To navigate relative to a given document range, see the methods and properties of the AuthenticRange[969] object.

17.2.2.8.1 Events

17.2.2.8.1.1 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 */
}
```

17.2.2.8.1.2 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()
```


© 2016-2022 Altova GmbH

```
{
        // return false;  /* to disable operation */
}
```

17.2.2.8.1.3 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 */
}
```

17.2.2.8.1.4 OnBeforeDrop

Event: OnBeforeDrop (*i_nXPos* as Long, *i_nYPos* as Long, *i_ipRange* as AuthenticRange 969, *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 */
}
```

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.

17.2.2.8.1.5 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 */
}
```

Description

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.

17.2.2.8.1.6 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. (Please note, that the event fires when saving the XML document, and not when saving the SPS design in StyleVision.)

17.2.2.8.1.7 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 );
```

 © 2016-2022 Altova GmbH

```
    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[1002].

17.2.2.8.1.8 OnMouseEvent

Event: OnMouseEvent (*nXPos* as Long, *nYPos* as Long, *eMouseEvent* as SPYMouseEvent[1048], *objRange* as AuthenticRange[969]) 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 */
}
```

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[1048] 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*.

17.2.2.8.1.9 OnSelectionChanged

Event: OnSelectionChanged (*objNewSelection* as AuthenticRange[969])

Scripting environment - VBScript:

```
Function On_AuthenticSelectionChanged (objNewSelection)
End Function
```

Scripting environment - JScript:

```
function On_AuthenticSelectionChanged (objNewSelection)
{
}
```

Description

This event gets triggered whenever the selection in the user interface changes.

17.2.2.8.1.10 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

© 2016-2022 Altova GmbH

- 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.

17.2.2.8.1.11 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 1002.

`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.

17.2.2.8.1.12 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.

17.2.2.8.2 Application

Property: Application as Application 952 (read-only)

Description
Accesses the StyleVision application object.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

17.2.2.8.3 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.

17.2.2.8.4 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.

17.2.2.8.5 CreateXMLNode

Method: CreateXMLNode (*nKind* as SPYXMLDataKind 1050) as XMLData 1033

Return Value
The method returns the new XMLData 1033 object.

Description
To create a new XMLData object use the CreateXMLNode() method.

 © 2016-2022 Altova GmbH

Errors

2000 Invalid object.
2012 Cannot create XML node.

17.2.2.8.6 DisableAttributeEntryHelper

Method: DisableAttributeEntryHelper()

Description

DisableAttributeEntryHelper() disables the attribute entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

Errors

2000 Invalid object.

17.2.2.8.7 DisableElementEntryHelper

Method: DisableElementEntryHelper()

Description

DisableElementEntryHelper() disables the element entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

Errors

2000 Invalid object.

17.2.2.8.8 DisableEntityEntryHelper

Method: DisableEntityEntryHelper()

Description

DisableEntityEntryHelper() disables the entity entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

Errors

2000 Invalid object.

17.2.2.8.9 DocumentBegin

Property: DocumentBegin as AuthenticRange 969 (read-only)

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.

17.2.2.8.10 DocumentEnd

Property: DocumentEnd as AuthenticRange 969 (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.

17.2.2.8.11 DoNotPerformStandardAction

Method: DoNotPerformStandardAction `()`

Description
`DoNotPerformStandardAction()` serves as cancel bubble for macros, and stops further execution after macro has finished.

Errors
2000 Invalid object.

17.2.2.8.12 EvaluateXPath

Method: EvaluateXPath (XMLData as XMLData 1033, 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.

© 2016-2022 Altova GmbH

17.2.2.8.13 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.

17.2.2.8.14 EventContext

Property: EventContext() as EventContext

Description
`EventContext` property gives access to the running macros context. See the `EventContext` 966 interface description for more details.

Errors
2000 Invalid object.

17.2.2.8.15 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 1002). 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 1048.

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.

2005 Invalid parameter.
2008 Internal error.
2014 Invalid button identifier.

17.2.2.8.16 Goto

Method: Goto (*eKind* as SPYAuthenticElementKind[1047], *nCount* as Long, *eFrom* as *SPYAuthenticDocumentPosition*[1047]) as *AuthenticRange*[969]

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
```

17.2.2.8.17 IsRedoEnabled

Property: IsRedoEnabled as Boolean (read-only)

Description
True if redo steps are available and Redo[1010] is possible.

Errors

 © 2016-2022 Altova GmbH

2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

17.2.2.8.18 IsUndoEnabled

Property: IsUndoEnabled as Boolean (read-only)

Description
True if undo steps are available and Undo[1011] is possible.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

17.2.2.8.19 MarkupVisibility

Property: MarkupVisibility as SPYAuthenticMarkupVisibility[1048]

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.

17.2.2.8.20 Parent

Property: Parent as Document[1012] (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.

17.2.2.8.21 Print

Method: Print (*bWithPreview* as Boolean, *bPromptUser* as Boolean)

Description

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.

17.2.2.8.22 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.

17.2.2.8.23 Selection

Property: Selection as AuthenticRange 969

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
```

 © 2016-2022 Altova GmbH

17.2.2.8.24 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 1002). 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 1048.

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.

17.2.2.8.25 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.

17.2.2.8.26 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
// ----------

```
// 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();
	}
}
```

17.2.2.8.27 WholeDocument

Property: WholeDocument as AuthenticRange 969 (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.

17.2.2.8.28 XMLDataRoot

Property: XMLDataRoot as XMLData 1033 (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.

17.2.2.9 Document

The `Document` interface has the following methods and properties:

Methods

 © 2016-2022 Altova GmbH

- Activate 1014
- Application 1014
- AssignWorkingXMLFile 1014
- Close 1015
- Save 1017
- SaveAs 1017
- Saved 1017
- SaveGeneratedFOFile 1018
- SaveGeneratedFOFileEx 1018
- SaveGeneratedHTMLFile 1018
- SaveGeneratedHTMLFileEx 1018
- SaveGeneratedPDFFile 1019
- SaveGeneratedPDFFileEx 1019
- SaveGeneratedRTFFile 1019
- SaveGeneratedRTFFileEx 1019
- SaveGeneratedWord2007File 1020
- SaveGeneratedWord2007FileEx 1020
- SaveGeneratedTextFile 1020
- SaveGeneratedTextFileEx 1020
- SaveGeneratedXSLTFOFile 1021
- SaveGeneratedXSLTFOFileEx 1021
- SaveGeneratedXSLTHTMLFile 1021
- SaveGeneratedXSLTHTMLFileEx 1021
- SaveGeneratedXSLTRTFFile 1022
- SaveGeneratedXSLTRTFFileEx 1022
- SaveGeneratedXSLTWord2007File 1022
- SaveGeneratedXSLTWord2007FileEx 1022
- SaveGeneratedXSLTTextFile 1023
- SaveGeneratedXSLTTextFileEx 1023

Properties

- Application 1014
- FullName 1015
- Name 1016
- Parameters 1016
- Parent 1016
- Path 1017
- SchemaSources 1023

17.2.2.9.1 Events

17.2.2.9.1.1 OnDocumentClosed

Event: OnDocumentClosed(*Document* as Document 1012)

Description
This event gets fired as a result of closing a document.

17.2.2.9.1.2 OnModifiedFlagChanged

Event: OnModifiedFlagChanged(as Boolean)

Description
Returns true if the Modified flag has been changed.

17.2.2.9.2 Activate

Method: Activate ()

Description
Activate document frame.

Errors
1200 Document object is invalid.
1201 Invalid input parameter.

17.2.2.9.3 Application

Property: Application as Application 952 (read-only)

Description
Accesses the StyleVision application object.

Errors
140011 The object is no longer valid.
11
140711 Invalid address for the return parameter was specified.
00

17.2.2.9.4 AssignWorkingXMLFile

Method: AssignWorkingXMLFile (*strWorkingXMLFileName* as String, *strSchemaSourceName* as String)

Description
Assigns the Working XML File by supplying its URI as a string and the Schema Source Name as a string. The Schema Source Name is the name assigned to the schema in the SPS; (the default schema name in an SPS created new in StyleVision is: `XML`).

Errors
1200 The document object is invalid.
1201 Invalid input parameter.

© 2016-2022 Altova GmbH

1203 Error assigning Working XML File
1404 Missing XML Schema or DTD

17.2.2.9.5 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.

17.2.2.9.6 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[1015] and SetPathName[1023] obsolete.

Errors
1400 The document object is no longer valid.
1402 Empty string has been specified as full file name.

17.2.2.9.7 GetPathName (obsolete)

Superseded by Document.FullName[1015]

```
// ----- 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[1023] (obsolete).

17.2.2.9.8 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[1015].

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

17.2.2.9.9 Parameters

Property: Parameters as Parameters (read-only)

Description
Reference to the current `Parameters` object.

Errors
1200 Document object is invalid.
1201 Invalid input parameter.

17.2.2.9.10 Parent

Property: Parent as Documents[1024] (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[952] (read-only)

Description
Access the StyleVision application object.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

17.2.2.9.11 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[1015].

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

17.2.2.9.12 Save

Method: Save()

Description
The method writes any modifications of the document to the associated file.

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.

17.2.2.9.13 SaveAs

Method: SaveAs (*strFileName* as String)

Description
Save the document to the file specified.

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.

17.2.2.9.14 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.

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

17.2.2.9.15 SaveGeneratedFOFile

Method: SaveGeneratedFOFile (*strFileName* as String)

Description
Saves the generated file to the location specified.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.16 SaveGeneratedFOFileEx

Method: SaveGeneratedFOFileEx (*strFileName* as String, *pbError* as Variant) as AppOutputLines

Description
Saves the generated file to the location specified and gives error description.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.17 SaveGeneratedHTMLFile

Method: SaveGeneratedHTMLFile (*strFileName* as String)

Description
Saves the generated file to the location specified.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.18 SaveGeneratedHTMLFileEx

Method: SaveGeneratedHTMLFileEx (*strFileName* as String, *pbError* as Variant) as AppOutputLines

Description
Saves the generated file to the location specified and gives error description.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

© 2016-2022 Altova GmbH

17.2.2.9.19 SaveGeneratedPDFFile

Method: SaveGeneratedPDFFile (*strFileName* as String)

Description
Saves the generated file to the location specified.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.20 SaveGeneratedPDFFileEx

Method: SaveGeneratedPDFFileEx (*strFileName* as String, *pbError* as Variant) as AppOutputLines

Description
Saves the generated file to the location specified and gives error description.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.21 SaveGeneratedRTFFile

Method: SaveGeneratedRTFFile (*strFileName* as String)

Description
Saves the generated file to the location specified.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.22 SaveGeneratedRTFFileEx

Method: SaveGeneratedRTFFileEx (*strFileName* as String, *pbError* as Variant) as AppOutputLines

Description
Saves the generated file to the location specified and gives error description.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.23 SaveGeneratedWord2007File

Method: SaveGeneratedWord2007File (*strFileName* as String)

Description
Saves the generated file to the location specified.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.24 SaveGeneratedWord2007FileEx

Method: SaveGeneratedWord2007FileEx (*strFileName* as String, *pbError* as Variant) as AppOutputLines

Description
Saves the generated file to the location specified and gives error description.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.25 SaveGeneratedTextFile

Method: SaveGeneratedTextFile (*strFileName* as String)

Description
Saves the generated file to the location specified.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.26 SaveGeneratedTextFileEx

Method: SaveGeneratedTextFileEx (*strFileName* as String, *pbError* as Variant) as AppOutputLines

Description
Saves the generated file to the location specified and gives error description.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

© 2016-2022 Altova GmbH

17.2.2.9.27 SaveGeneratedXSLTFOFile

Method: SaveGeneratedXSLTFOFile (*strFileName* as String)

Description
Saves the generated file to the location specified.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.28 SaveGeneratedXSLTFOFileEx

Method: SaveGeneratedXSLTFOFileEx (*strFileName* as String, *pbError* as Variant) as AppOutputLines

Description
Saves the generated file to the location specified and gives error description.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.29 SaveGeneratedXSLTHTMLFile

Method: SaveGeneratedXSLTHTMLFile (*strFileName* as String)

Description
Saves the generated file to the location specified.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.30 SaveGeneratedXSLTHTMLFileEx

Method: SaveGeneratedXSLTHTMLFileEx (*strFileName* as String, *pbError* as Variant) as AppOutputLines

Description
Saves the generated file to the location specified and gives error description.

Errors

1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.31 SaveGeneratedXSLTRTFFile

Method: SaveGeneratedXSLTRTFFile (*strFileName* as String)

Description
Saves the generated file to the location specified.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.32 SaveGeneratedXSLTRTFFileEx

Method: SaveGeneratedXSLTRTFFileEx (*strFileName* as String, *pbError* as Variant) as AppOutputLines

Description
Saves the generated file to the location specified and gives error description.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.33 SaveGeneratedXSLTWord2007File

Method: SaveGeneratedXSLTWord2007File (*strFileName* as String)

Description
Saves the generated file to the location specified.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.34 SaveGeneratedXSLTWord2007FileEx

Method: SaveGeneratedXSLTWord2007FileEx (*strFileName* as String, *pbError* as Variant) as AppOutputLines

Description
Saves the generated file to the location specified and gives error description.

© 2016-2022 Altova GmbH

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.35 SaveGeneratedXSLTTextFile

Method: SaveGeneratedXSLTTextFile (*strFileName* as String)

Description
Saves the generated file to the location specified.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.36 SaveGeneratedXSLTTextFileEx

Method: SaveGeneratedXSLTTextFileEx (*strFileName* as String, *pbError* as Variant) as AppOutputLines

Description
Saves the generated file to the location specified and gives error description.

Errors
1201 Invalid document object.
1204 Cannot generate output file.

17.2.2.9.37 SchemaSources

Property: SchemaSources as SchemaSources (read-only)

Description
Reference to the current `SchemaSources` object.

Errors
1200 Document object is invalid.
1201 Invalid input parameter.

17.2.2.9.38 SetPathName (obsolete)

Superseded by Document.FullName 1015

```
// ----- 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.

17.2.2.10 Documents

The `Documents` interface has the following methods and properties:

Methods

- ActiveDocument 1024
- Item 1025
- NewDocument 1025
- OpenDocument 1026

Properties

- Application 1025
- Count 1025
- Parent 1026

17.2.2.10.1 ActiveDocument

Property: ActiveDocument as Document 1012

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.
1600	Invalid `Documents` object
1601	Invalid input parameter

© 2016-2022 Altova GmbH

17.2.2.10.2 Application

Property: Application as Application 952 (read-only)

Description
Access the StyleVision application object.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.10.3 Count

Property: Count as long

Description
Count of open documents.

Errors
1600 Invalid `Documents` object
1601 Invalid input parameter

17.2.2.10.4 Item

Method: Item (*n* as long) as Document 1012

Description
Gets the document with the index n in this collection. Index is 1-based.

Errors
1600 Invalid `Documents` object
1601 Invalid input parameter

17.2.2.10.5 NewDocument

Method: NewDocument() as Document 1012

Return Value
None

Description

Creates a new empty document based on the previous template.

Errors

1000 The application object is invalid.
1005 Error when creating a new document
1006 Cannot create document
1600 Invalid `Documents` object
1601 Invalid input parameter

17.2.2.10.6 OpenDocument

Method: OpenDocument(strFileName as *String*) as Document 1012

Return Value
None

Description
Opens an existing SPS file.

Errors

1000 The application object is invalid.
1002 Invalid file extension.
1003 Error when opening document.
1004 Cannot open document.
1600 Invalid `Documents` object
1601 Invalid input parameter

17.2.2.10.7 Parent

Property: Parent as Application 952 (read-only)

Description
Access the StyleVision application object.

Errors

1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.11 Parameter

The `Parameter` interface has the following properties:

Properties

© 2016-2022 Altova GmbH

- Application 1027
- Name 1027
- Parent 1027
- Value 1027

17.2.2.11.1 Application

Property: Application as Application 952 (read-only)

Description
Access the StyleVision application object.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.11.2 Name

Property: Name as String (read-only)

Errors
1600 Invalid `Documents` object
1601 Invalid input parameter

17.2.2.11.3 Parent

Property: Parent as Application 952 (read-only)

Description
Access the StyleVision application object.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.11.4 Value

Property: Value as String (read-only)

Errors
1600 Invalid `Documents` object
1601 Invalid input parameter

17.2.2.12 Parameters

The `Parameters` interface has the following properties:

Properties

- Application 1028
- Count 1028
- Item 1028
- Parent 1029

17.2.2.12.1 Application

Property: Application as Application 952 (read-only)

Description
Access the StyleVision application object.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.12.2 Count

Property: Count as long

Description
Count of parameters.

Errors
1600 Invalid `Parameters` object
1601 Invalid input parameter

17.2.2.12.3 Item

Method: Item (*n* as long) as Document 1012

Description
Gets the document with the index n in this collection. Index is 1-based.

© 2016-2022 Altova GmbH

Errors

1600 Invalid `Documents` object
1601 Invalid input parameter

17.2.2.12.4 Parent

Property: Parent as Application 952 (read-only)

Description
Access the StyleVision application object.

Errors

1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.13 SchemaSource

The `SchemaSource` interface has the following properties:

Properties

- Application 1029
- IsMainSchemaSource 1030
- Name 1030
- Parent 1030
- SchemaFileName 1030
- TemplateFileName 1031
- Type 1031
- TypeName 1031
- WorkingXMLFileName 1032

17.2.2.13.1 Application

Property: Application as Application 952 (read-only)

Description
Access the StyleVision application object.

Errors

1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.13.2 IsMainSchemaSource

Property: IsMainSchemaSource as Boolean (read-only)

Description
Returns true if schema source is the main schema source.

Errors
1400 Invalid schema source object.
1401 Invalid parameter.

17.2.2.13.3 Name

Property: Name as String (read-only)

Description
Use this property to retrieve the name of the schema source.

Errors
1400 The schema source object is not valid.
1401 Invalid parameter.

17.2.2.13.4 Parent

Property: Parent as `SchemaSource` (read-only)

Description
The parent object according to the object model.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.13.5 SchemaFileName

Property: SchemaFileName as String

Description
Use this property to retrieve the name of the Schema File.

Errors
1400 Invalid schema source object.
1401 Invalid parameter.

© 2016-2022 Altova GmbH

1403 Missing XMLSchema or DTD
1406 Error assigning schema File

17.2.2.13.6 TemplateFileName

Property: TemplateFileName as String

Description
Use this property to retrieve the name of the Working XML File.

Errors
1400 Invalid schema source object.
1401 Invalid parameter.
1403 Missing XMLSchema or DTD
1407 Error assigning Template XML File

17.2.2.13.7 Type

Property: Type as ENUMSchemaSourceType (read-only)

Description
Use this property to retrieve the type of the schema source.

Errors
1400 Invalid schema source object.
1401 Invalid parameter.

17.2.2.13.8 TypeName

Property: TypeName as String (read-only)

Description
Use this property to retrieve the type of the schema source.

Errors
1400 Invalid schema source object.
1401 Invalid parameter.

17.2.2.13.9 WorkingXMLFileName

Property: WorkingXMLFileName as String

Description
Use this property to retrieve the name of the Working XML File.

Errors
1400 Invalid schema source object.
1401 Invalid parameter.
1403 Missing XMLSchema or DTD
1203 Error assigning Working XML File

17.2.2.14 SchemaSources

The `SchemaSources` interface has the following properties:

Properties

- Application 1032
- MainSchemaSource 1033
- Parent 1033

17.2.2.14.1 Application

Property: Application as Application 952 (read-only)

Description
Access the StyleVision application object.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.14.2 Count

Property: Count as long (read-only)

Description
Count of schema sources.

Errors
1300 Invalid `SchemaSources` object

© 2016-2022 Altova GmbH

1301 Invalid input parameter

17.2.2.14.3 Item

Method: Item (*n* as long) as Document[1012]

Description
Gets the document with the index n in this collection. Index is 1-based.

Errors
1300 Invalid `SchemaSources` object
1301 Invalid input parameter

17.2.2.14.4 MainSchemaSource

Property: MainSchemaSource as SchemaSource (read-only)

Errors
1300 Invalid `SchemaSources` object
1301 Invalid input parameter

17.2.2.14.5 Parent

Property: Parent as SchemaSources (read-only)

Description
The parent object according to the object model.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

17.2.2.15 XMLData

Properties
Kind[1043]
Name[1044]
TextValue[1044]

HasChildren[1041]

MayHaveChildren 1043
Parent 1044

Methods
GetFirstChild 1039
GetNextChild 1040
GetCurrentChild 1039

InsertChild 1042
InsertChildAfter 1042
InsertChildBefore 1042
AppendChild 1035

EraseAllChildren 1036
EraseChild 1036
EraseCurrentChild 1036

IsSameNode 1043

CountChildren 1035
CountChildrenKind 1035

GetChild 1037
GetChildAttribute 1038
GetChildElement 1038
GetChildKind 1038
GetNamespacePrefixForURI 1039

HasChildrenKind 1041
SetTextValueXMLEncoded 1044

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 1044 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.

 © 2016-2022 Altova GmbH

17.2.2.15.1 AppendChild

Declaration: AppendChild (*pNewData* as XMLData[1033])

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
```

17.2.2.15.2 CountChildren

Declaration: CountChildren as long

Description
CountChildren gets the number of children.

Available with TypeLibrary version 1.5

Errors
1500 The XMLData object is no longer valid.

17.2.2.15.3 CountChildrenKind

Declaration: CountChildrenKind (*nKind* as SPYXMLDataKind[1050]) 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.

17.2.2.15.4 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
```

17.2.2.15.5 EraseChild

Method: EraseChild (Child as XMLData[1033])

Description
Deletes the given child node.

Errors
1500 Invalid object.
1506 Invalid input xml
1510 Invalid parameter.

17.2.2.15.6 EraseCurrentChild

Declaration: EraseCurrentChild

Description
EraseCurrentChild deletes the current XMLData child object. Before you call EraseCurrentChild you must initialize an internal iterator with XMLData.GetFirstChild[1039]. After deleting the current child, EraseCurrentChild increments the internal iterator of the XMLData element. No error is returned when the last child gets erased

 © 2016-2022 Altova GmbH

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); }
}
```

17.2.2.15.7 GetChild

Declaration: GetChild (*position* as long) as XMLData 1033

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.

17.2.2.15.8 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.

17.2.2.15.9 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.

17.2.2.15.10 GetChildKind

Declaration: GetChildKind (*position* as long, *nKind* as SPYXMLDataKind[1050]) as XMLData[1033]

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.

© 2016-2022 Altova GmbH

17.2.2.15.11 GetCurrentChild

Declaration: GetCurrentChild as XMLData[1033]

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[1039].

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.

17.2.2.15.12 GetFirstChild

Declaration: GetFirstChild (*nKind* as SPYXMLDataKind[1050]) as XMLData[1033]

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[1039], GetNextChild[1040] or EraseCurrentChild[1036].

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[1040].

17.2.2.15.13 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.

17.2.2.15.14 GetNextChild

Declaration: GetNextChild as XMLData 1033

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 1039.

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);
```


© 2016-2022 Altova GmbH

```
        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);
}
```

17.2.2.15.15 GetTextValueXMLDecoded

Method: GetTextValueXMLDecoded ()as string

Description
Gets the decoded text value of the XML.

Errors
1500 Invalid object.
1510 Invalid parameter.

17.2.2.15.16 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.

17.2.2.15.17 HasChildrenKind

Declaration: HasChildrenKind (*nKind* as SPYXMLDataKind[1050]) 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.

17.2.2.15.18 InsertChild

Declaration: InsertChild (*pNewData* as XMLData[1033])

Description

InsertChild inserts the new child before the current child (see also XMLData.GetFirstChild[1039], XMLData.GetNextChild[1040] 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

17.2.2.15.19 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

17.2.2.15.20 InsertChildBefore

Method: InsertChildBefore (Node as XMLData, NewData as XMLData)

© 2016-2022 Altova GmbH

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

17.2.2.15.21 IsSameNode

Declaration: IsSameNode (*pNodeToCompare* as XMLData[1033]) 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.

17.2.2.15.22 Kind

Declaration: Kind as SPYXMLDataKind[1050]

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.

17.2.2.15.23 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.

17.2.2.15.24 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.

17.2.2.15.25 Parent

Declaration: Parent as XMLData 1033

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.

17.2.2.15.26 SetTextValueXMLEncoded

Method: SetTextValueXMLEncoded (*strVal* as String 1050)

Description
Sets the encoded text value of the XML.

Errors
1500 Invalid object.
1513 Modification not allowed.

17.2.2.15.27 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.

© 2016-2022 Altova GmbH

1510 Invalid address for the return parameter was specified.

17.2.3 Enumerations

This is a list of all enumerations used by the StyleVision API. If your scripting environment does not support enumerations use the number-values instead.

17.2.3.1 ENUMApplicationStatus

Description
Enumeration to specify the current Application status.

Possible values:

eApplicationRunning	= 0
eApplicationAfterLicenseCheck	= 1
eApplicationBeforeLicenseCheck	= 2
eApplicationConcurrentLicenseCheckFailed	= 3
eApplicationProcessingCommandLine	= 4

17.2.3.2 ENUMAppOutputLine_Severity

Description
Enumeration values to identify the severity of an AppOutputLine.

Possible values:

eSeverity_Undefined	= -1
eSeverity_Info	= 0
eSeverity_Warning	= 1
eSeverity_Error	= 2
eSeverity_CriticalError	= 3
eSeverity_Success	= 4
eSeverity_Summary	= 5
eSeverity_Progress	= 6
eSeverity_DataEdit	= 7
eSeverity_ParserInfo	= 8
eSeverity_PossibleInconsistencyWarning	= 9
eSeverity_Message	= 10
eSeverity_Document	= 11
eSeverity_Rest	= 12
eSeverity_NoSelect	= 13
eSeverity_Select	= 14
eSeverity_Autoinsertion	= 15
eSeverity_GlobalResources_DefaultWarning	= 16

17.2.3.3 ENUMAppOutputLine_TextDecoration

Description
Enumeration values for the different kinds of text decoration of an AppOutputLine.

Possible values:

eTextDecorationDefault	= 0
eTextDecorationBold	= 1
eTextDecorationDebugValues	= 2
eTextDecorationDB_ObjectName	= 3
eTextDecorationDB_ObjectLink	= 4
eTextDecorationDB_ObjectKind	= 5
eTextDecorationDB_TimeoutValue	= 6
eTextDecorationFind_MatchingString	= 7
eTextDecorationValidation_Speclink	= 8
eTextDecorationValidation_ErrorPosition	= 9
eTextDecorationValidation_UnkownParam	= 10

17.2.3.4 ENUMSchemaSourceType

Description
Enumeration to specify the source schema type: XML Schema, DTD, DB, DB cell, User-Defined, or XBRL.

Possible values:

eSchemaSourceType_XSDorDTD	= 0
eSchemaSourceType_DB	= 1
eSchemaSourceType_DBCell	= 2
eSchemaSourceType_User	= 3
eSchemaSourceType_XBRL	= 4

17.2.3.5 ENUMSchemaType

Description
Enumeration to specify the schema type: W3C XML Schema or DTD.

Possible values:

eSchemaTypeW3CSchema	= 0
eSchemaTypeDTD	= 1

© 2016-2022 Altova GmbH

17.2.3.6 SPYAuthenticActions

Description
Actions that can be performed on AuthenticRange 969 objects.

Possible values:

spyAuthenticInsertAt	= 0
spyAuthenticApply	= 1
spyAuthenticClearSurr	= 2
spyAuthenticAppend	= 3
spyAuthenticInsertBefore	= 4
spyAuthenticRemove	= 5

17.2.3.7 SPYAuthenticDocumentPosition

Description
Relative and absolute positions used for navigating with AuthenticRange 969 objects.

Possible values:

spyAuthenticDocumentBegin	= 0
spyAuthenticDocumentEnd	= 1
spyAuthenticRangeBegin	= 2
spyAuthenticRangeEnd	= 3

17.2.3.8 SPYAuthenticElementKind

Description
Enumeration of the different kinds of elements used for navigation and selection within the AuthenticRange 969 and AuthenticView 997 objects.

Possible values:

spyAuthenticChar	= 0
spyAuthenticWord	= 1
spyAuthenticLine	= 3
spyAuthenticParagraph	= 4
spyAuthenticTag	= 6
spyAuthenticDocument	= 8
spyAuthenticTable	= 9
spyAuthenticTableRow	= 10
spyAuthenticTableColumn	= 11

17.2.3.9 SPYAuthenticMarkupVisibility

Description
Enumeration values to customize the visibility of markup with MarkupVisibility[1009].

Possible values:

spyAuthenticMarkupHidden	= 0
spyAuthenticMarkupSmall	= 1
spyAuthenticMarkupLarge	= 2
spyAuthenticMarkupMixed	= 3

17.2.3.10 SPYAuthenticToolbarButtonState

Description
Authentic toolbar button states are given by the following enumeration:

Possible values:

authenticToolbarButtonDefault	= 0
authenticToolbarButtonEnabled	= 1
authenticToolbarButtonDisabled	= 2

17.2.3.11 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

© 2016-2022 Altova GmbH

```
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
```

17.2.3.12 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
```

17.2.3.13 SPYValidateErrorFormat

Description

Enumeration values that select the format of the error message.

Possible values

```
spyValidateErrorFormat_Text           = 0
spyValidateErrorFormat_ShortXML       = 1
spyValidateErrorFormat_LongXML        = 2
```

17.2.3.14 SPYXMLDataKind

Description
The different types of XMLData elements available for XML documents.

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

© 2016-2022 Altova GmbH

17.3 ActiveX Integration

The StyleVision 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 StyleVision Integration Package must be installed (see https://www.altova.com/components/download). Ensure that you install StyleVision first, and then the StyleVision Integration Package. Other prerequisites apply, depending on language and platform (see Prerequisites 1051).

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 StyleVision (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 StyleVision 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 StyleVision graphical user interface according to your custom requirements.

Integration at document level means embedding StyleVision into your own application piece-by-piece. This includes implementing not only the main StyleVision 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 1054 and Integration at Document Level 1056 describe the key steps at these respective levels. The ActiveX Integration Examples 1059 section provides examples in C#, HTML, and Java. Looking through these examples will help you to make the right decisions quickly. The Object Reference 1086 section describes all COM objects that can be used for integration, together with their properties and methods.

For information about using StyleVision as a Visual Studio plug-in, see StyleVision in Visual Studio 708.

17.3.1 Prerequisites

To integrate the StyleVision ActiveX control into a custom application, the following must be installed on your computer:

- StyleVision
- The StyleVision 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 StyleVision and StyleVision Integration Package. For applications developed under Microsoft .NET platform with Visual Studio, both the 32-bit and 64-bit versions of StyleVision and StyleVision Integration Package must be installed, as explained below.

Microsoft .NET (C#, VB.NET) with Visual Studio

To integrate the StyleVision 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
- StyleVision 32-bit and StyleVision Integration Package 32-bit
- The ActiveX controls must be added to the Visual Studio toolbox (see Adding the ActiveX Controls to the Toolbox 1052).

If you want to integrate the 64-bit ActiveX control, the following prerequisites apply in addition to the ones above:

- StyleVision 32-bit and StyleVision 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)
- StyleVision 64-bit and StyleVision 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 1059.

Java

To integrate the StyleVision 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
- StyleVision and StyleVision Integration Package

Note: To run the 64-bit version of the StyleVision ActiveX control, use a 64-bit version of Eclipse, as well as the 64-bit version of StyleVision and the StyleVision Integration Package.

StyleVision 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):

- StyleVision
- The StyleVision Integration Package
- The custom integration code or application.

17.3.2 Adding the ActiveX Controls to the Toolbox

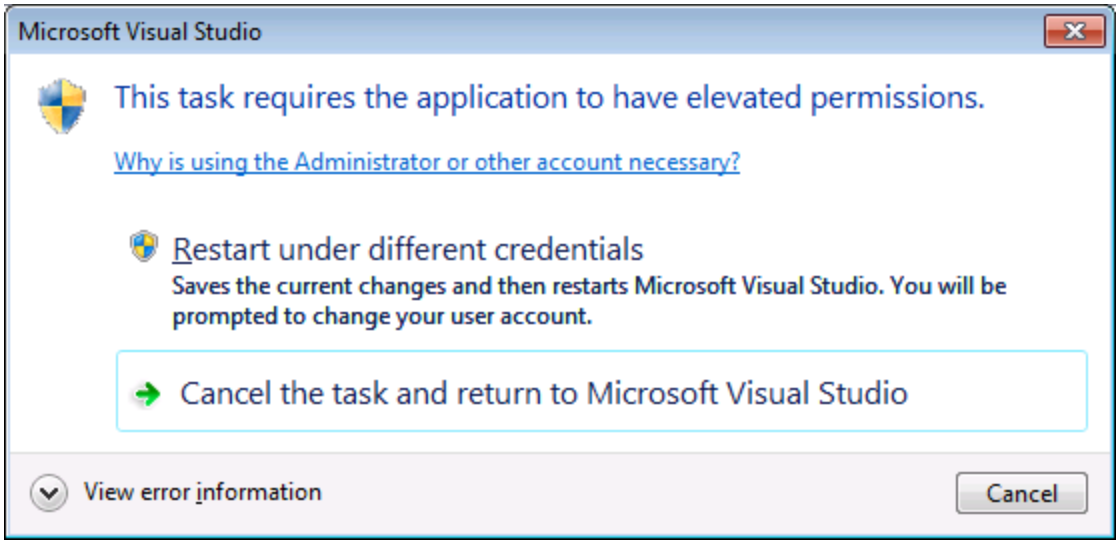
To use the StyleVision ActiveX controls in an application developed with Visual Studio, the controls must first be added to the Visual Studio Toolbox, as follows:

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 StyleVisionControl, StyleVisionControl Document, and StyleVisionControl Placeholder.

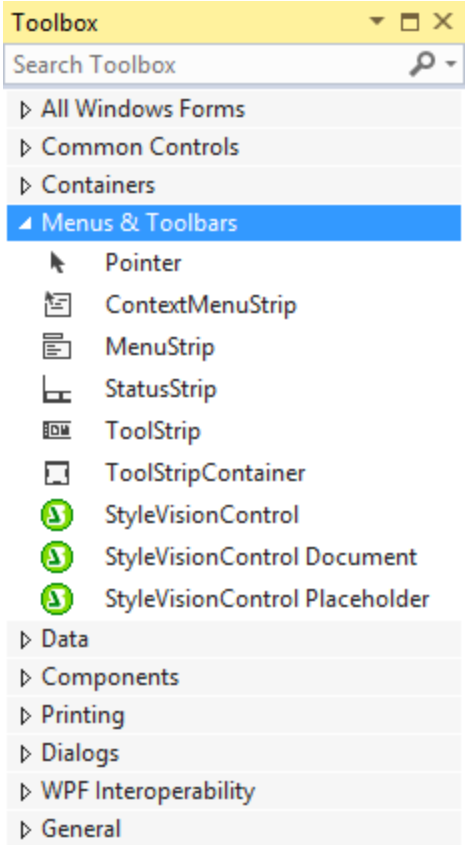
 © 2016-2022 Altova GmbH

In case the controls above are not available, follow the steps below:

1. On the **COM Components** tab, click **Browse**, and select the **StyleVisionControl.ocx** file from the StyleVision installation folder. Remember that the StyleVision Integration Package must be installed; otherwise, this file is not available, see Prerequisites 1051.
2. If prompted to restart Visual Studio with elevated permissions, click **Restart under different credentials**.



If the steps above were successful, the StyleVision ActiveX controls become available in the Visual Studio Toolbox.



Note: For an application-level integration, only the **StyleVisionControl** ActiveX control is used (see Integration at Application Level[1054]). The **StyleVisionControl Document** and **StyleVisionControl Placeholder** controls are used for document-level integration (see Integration at Document Level[1056]).

17.3.3 Integration at Application Level

Integration at application level allows you to embed the complete interface of StyleVision into a window of your application. With this type of integration, you get the whole user interface of StyleVision, including all menus, toolbars, the status bar, document windows, and helper windows. Customization of the application's user interface is restricted to what StyleVision provides. This includes rearrangement and resizing of helper windows and customization of menus and toolbars.

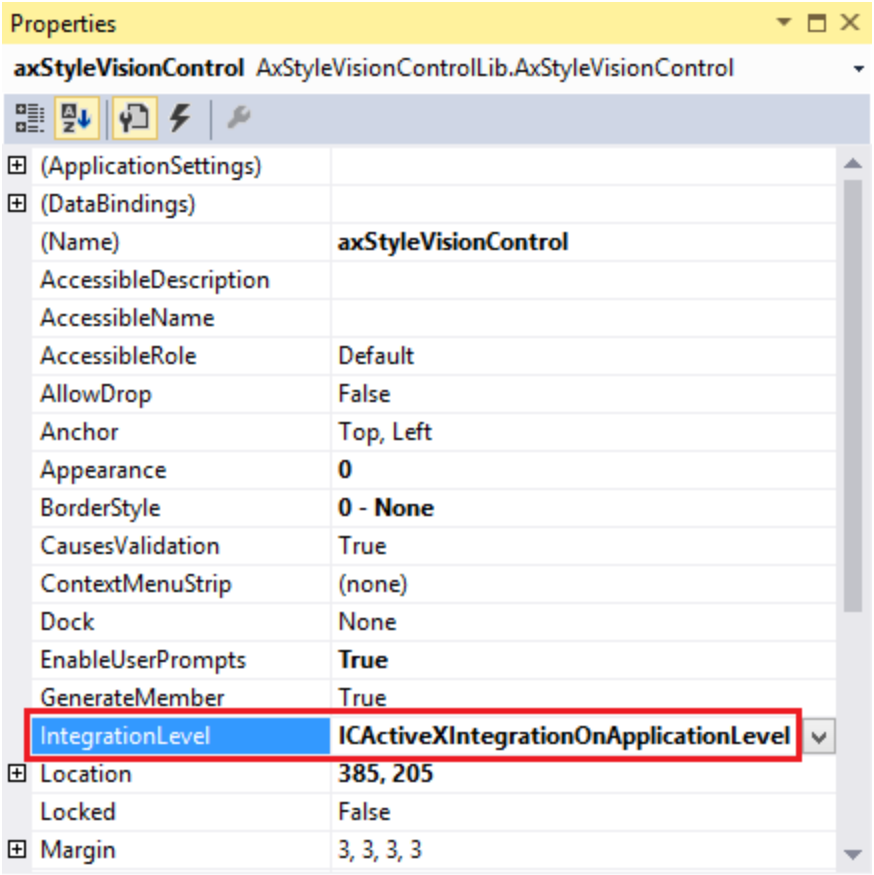
The only ActiveX control you need to integrate is `StyleVisionControl`[1089]. Do not instantiate or access `StyleVisionControlDocument`[1097] or `StyleVisionControlPlaceHolder`[1103] ActiveX controls when integrating at application-level.

If you have any initialization to do or if you want to automate some behaviour of StyleVision, use the properties, methods, and events described for `StyleVisionControl`[1089]. Consider using `StyleVisionControl.Application`[1090] for more complex access to StyleVision functionality.

 © 2016-2022 Altova GmbH

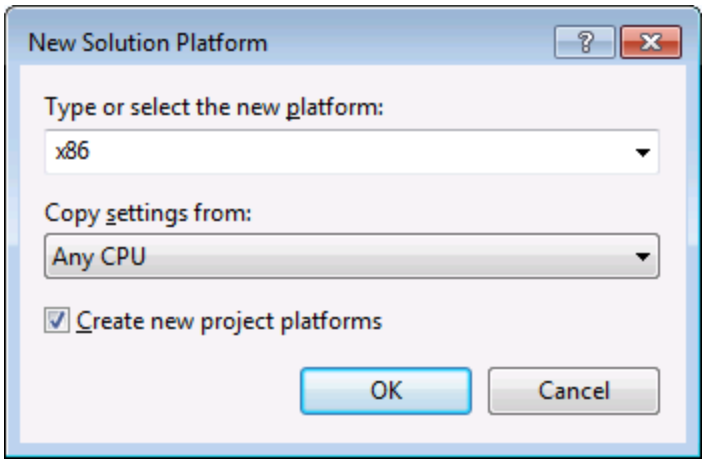
In C# or VB.NET with Visual Studio, the steps to create a basic, one-form application which integrates the StyleVision ActiveX controls at application level are as follows:

1. Check that all prerequisites are met (see Prerequisites[1051]).
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[1052]).
4. Drag the **StyleVisionControl** from the toolbox onto your new form.
5. Select the **StyleVisionControl** 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**).



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).

17.3.4 Integration at Document Level

Compared to integration at application level, integration at document level is a more complex, yet more flexible way to embed StyleVision functionality into your application by means of ActiveX controls. With this approach, your code can access selectively the following parts of the StyleVision user interface:

- Document editing window
- Project window
- Design Overview window
- Schema Tree window
- Design Tree window
- Styles window
- Style Repository window
- Properties window
- Messages window

As mentioned in Integration at Application Level [1054], for an ActiveX integration at application level, only one control is required, namely the **StyleVisionControl**. However, for an ActiveX integration at document level, StyleVision functionality is provided by the following ActiveX controls:

- StyleVisionControl [1089]
- StyleVisionControl Document [1097]
- StyleVisionControl Placeholder [1103]

These controls are supplied by the **StyleVisionControl.ocx** file available in the application installation folder of StyleVision. 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 [1052]).

The basic steps to integrate the ActiveX controls at document level into your application are as follows:

1. First, instantiate **StyleVisionControl** in your application. Instantiating this control is mandatory; it enables support for the **StyleVisionControl Document** and **StyleVisionControl Placeholder** controls mentioned above. It is important to set the `IntegrationLevel` [1091] property to

 © 2016-2022 Altova GmbH

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 **StyleVisionControl**; this might lead to unexpected results. Use the corresponding open methods of **StyleVisionControl Document** and **StyleVisionControl PlaceHolder** instead.

2. Create at least one instance of **StyleVisionControl Document** in your application. This control supplies the document editing window of StyleVision 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 **StyleVisionControl Placeholder** control for each additional window (other than the document window) that must be available to your application.

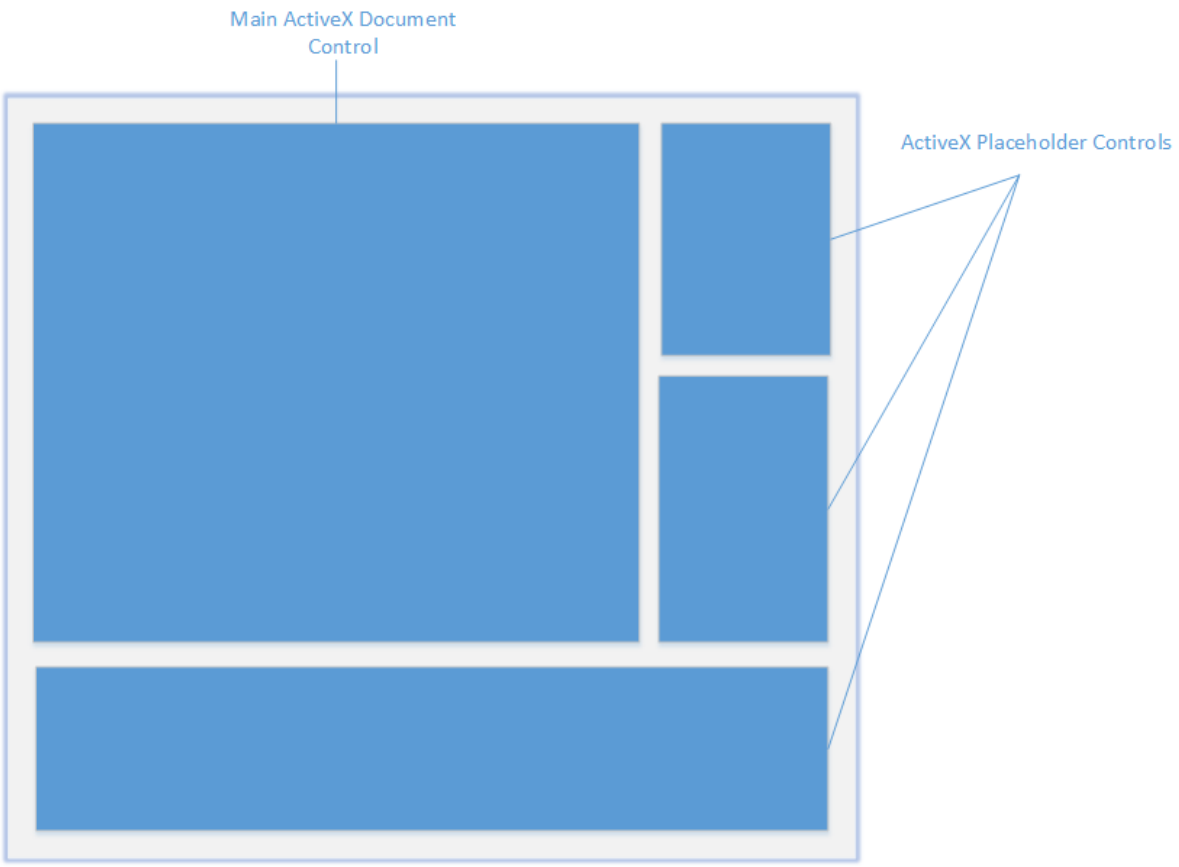
   Instances of **StyleVisionControl PlaceHolder** allow you to selectively embed additional windows of StyleVision into your application. The window kind (for example, Project window) is defined by the property **PlaceholderWindowID**. Therefore, to set the window kind, set the property **PlaceholderWindowID**. For valid window identifiers, see StyleVisionControlPlaceholderWindow 1106.

Note: Use only one **StyleVisionControl PlaceHolder** for each window identifier.

For placeholder controls that select the StyleVision project window, additional methods are available. Use **OpenProject** to load a StyleVision project. Use the property Project and the methods and properties from the StyleVision 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 StyleVision 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 1052).
3. Drag the **StyleVisionControl** 1089 from the toolbox onto your new form.
4. Set the **IntegrationLevel** property of the **StyleVisionControl** to **ICActiveXIntegrationOnDocumentLevel**, and the **Visible** property to **False**. You can do this either from code or from the **Properties** window.
5. Drag the **StyleVisionControl Document** 1097 from the toolbox onto the form. This control provides the main document window of StyleVision to your application, so you may need to resize it to a reasonable size for a document.
6. Optionally, add one or more **StyleVisionControl Placeholder** 1103 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:



7. Set the **PlaceholderWindowID** property of each **StyleVisionControl Placeholder** control to a valid window identifier. For the list of valid values, see StyleVisionControlPlaceholderWindow[1106].
8. Add commands to your application (at minimum, you will need to open, save and close documents), as shown below.

Querying StyleVision Commands

When you integrate at document level, no StyleVision 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`[1090] property of the **StyleVisionControl**.
- To retrieve commands organized according to their menu structure, use the `MainMenu`[1091] property.
- To retrieve commands organized by the toolbar in which they appear, use the `Toolbars`[1092] property.
- To send commands to StyleVision, use the `Exec`[1093] method.
- To query if a command is currently enabled or disabled, use the `QueryStatus`[1094] method.

This enables you to flexibly integrate StyleVision commands into your application's menus and toolbars.

Your installation of StyleVision also provides you with command label images used within StyleVision. See the folder **<ApplicationFolder>\Examples\ActiveX\Images** of your StyleVision installation for icons in GIF format. The file names correspond to the command names as they are listed in the Command Reference[1072] section.

 © 2016-2022 Altova GmbH

General considerations

To automate the behaviour of StyleVision, use the properties, methods, and events described for the **StyleVisionControl**[1089], **StyleVisionControl Document**[1097], and **StyleVisionControl Placeholder**[1103].

For more complex access to StyleVision functionality, consider using the following properties:

- `StyleVisionControl.Application`[1090]
- `StyleVisionControlDocument.Document`[1098]
- `StyleVisionControlPlaceHolder.Project`[1104]

These properties give you access to the StyleVision automation interface (StyleVisionAPI)

Note: To open a document, always use `StyleVisionControlDocument.Open`[1100] or `StyleVisionControlDocument.New`[1099] on the appropriate document control. To open a project, always use `StyleVisionControlPlaceHolder.OpenProject`[1104] on a placeholder control embedding a StyleVision project window.

For examples that show how to instantiate and access the necessary controls in different programming environments, see ActiveX Integration Examples[1059].

17.3.5 ActiveX Integration Examples

This section contains examples of StyleVision 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 StyleVision installation.

17.3.5.1 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[1059]).

17.3.5.1.1 Running the Sample C# Solution

The sample Visual Studio solution available in the folder **<ApplicationFolder>\Examples\ActiveX\C#** illustrates how to consume the StyleVision 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[1051].

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.

Step 3: Check and set all required control properties

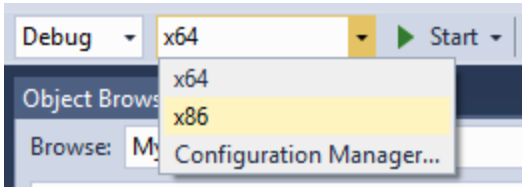
The sample application contains one instance of StyleVisionControlDocument[1097] and several instances of StyleVisionControlPlaceHolder[1103] controls. Double-check that the following properties of these controls are set as shown in the table below:

Control name	Property	Property value
axStyleVisionControl	IntegrationLevel	ICActiveXIntegrationOnDocumentLevel
axStyleVisionControlHelperWndSchemaSource	PlaceholderWindowID	2
axStyleVisionControlHelperWndStyleRepository	PlaceholderWindowID	4
axStyleVisionControlHelperWndContextStyle	PlaceholderWindowID	6
axStyleVisionControlHelperWndContextProperties	PlaceholderWindowID	5
axStyleVisionControlHelperWndProject	PlaceholderWindowID	0
axStyleVisionControlHelperWndDesignOverview	PlaceholderWindowID	1
axStyleVisionControlHelperWndDesignTree	PlaceholderWindowID	3

Here is how you can view or set the properties of an ActiveX control:

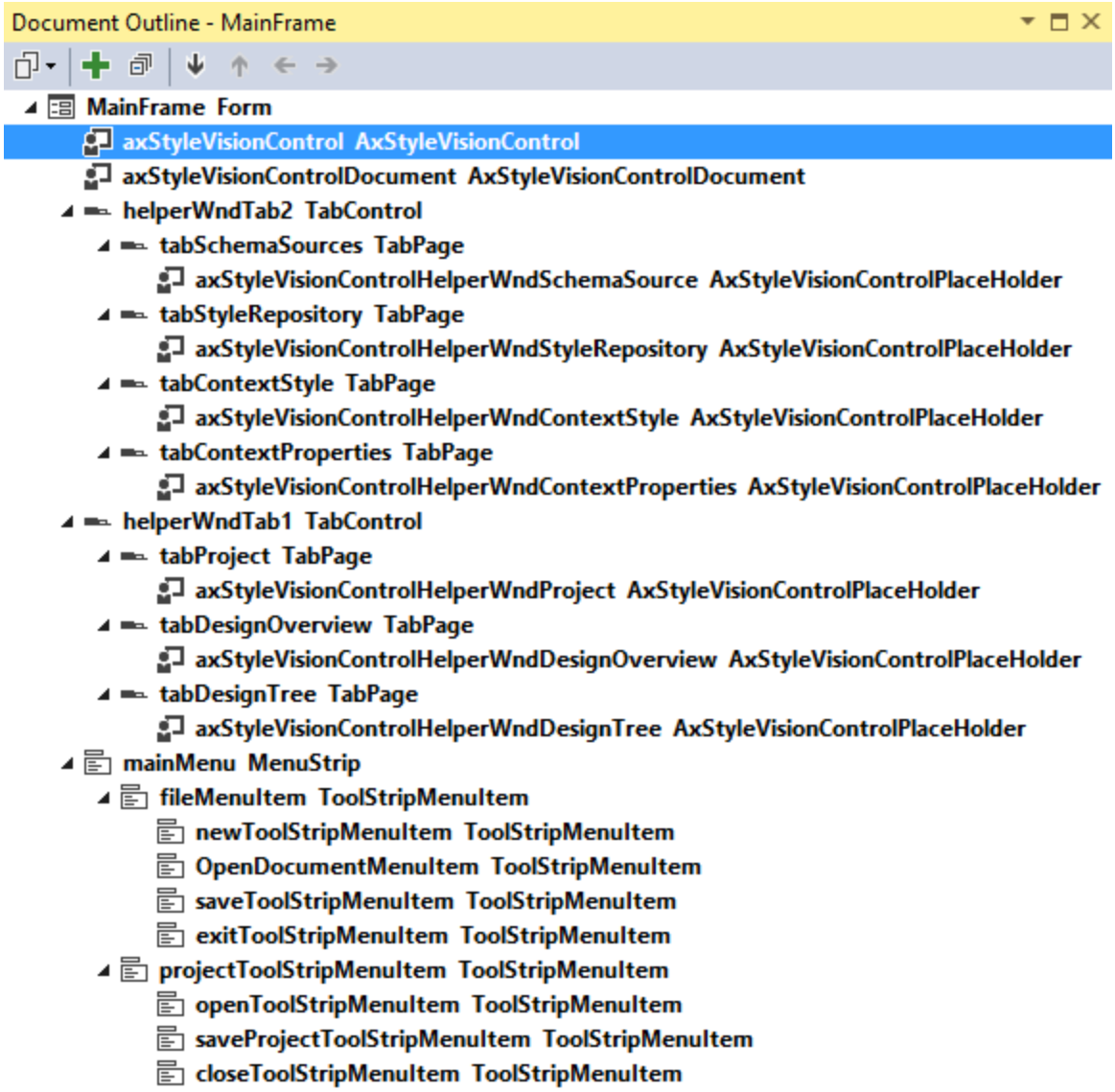
1. Open the **MDIMain.cs** form in the designer window.

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[1051].

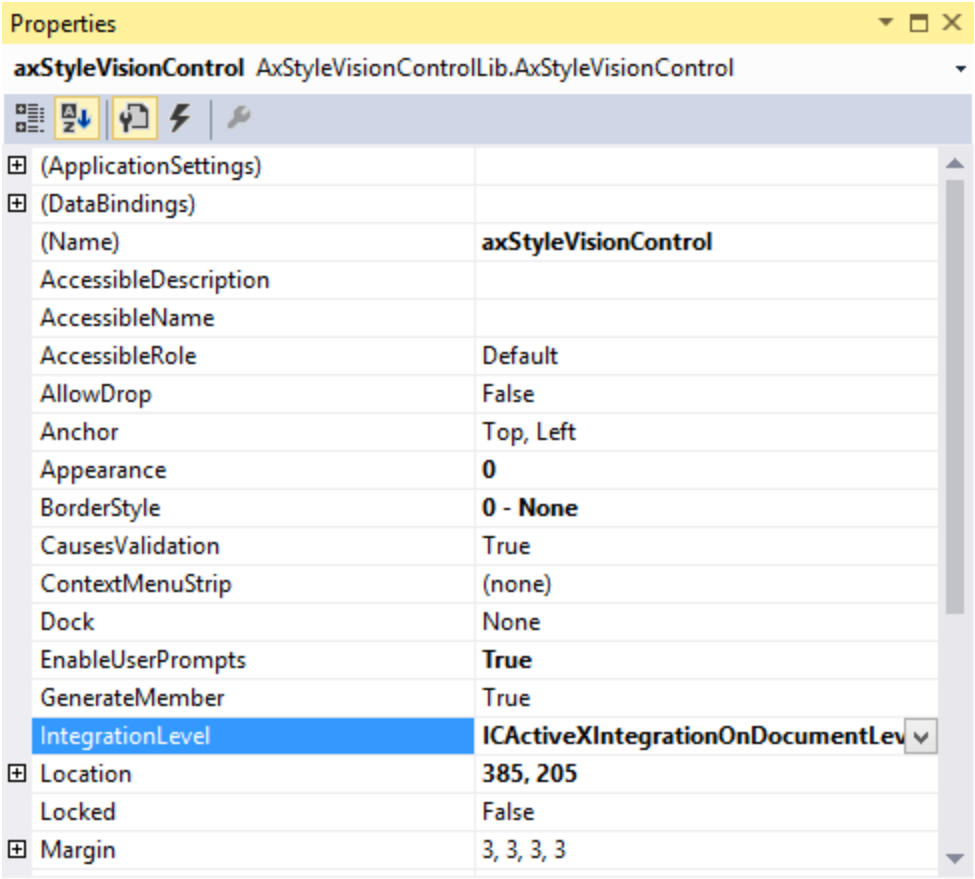


2. Open the **Document Outline** window of Visual Studio (On the **View** menu, click **Other Windows | Document Outline**).

© 2016-2022 Altova GmbH



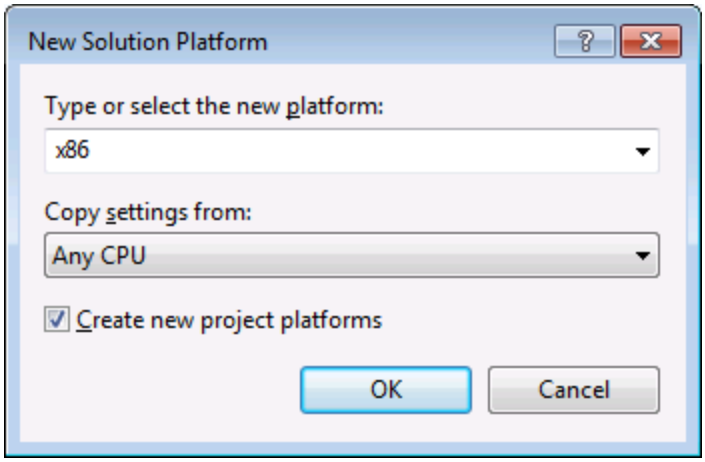
3. Click an ActiveX control in the **Document Outline** window, and edit its required property in the **Properties** window, for example:



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.

17.3.5.2 Java

StyleVision 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 StyleVision installation, after you have installed both StyleVision and the StyleVision Integration Package (see also Prerequisites[1051]).

- `AltovaAutomation.dll`: a JNI wrapper for Altova automation servers (in case of the 32-bit installation of StyleVision)
- `AltovaAutomation_x64.dll`: a JNI wrapper for Altova automation servers (in case of the 64-bit installation of StyleVision)
- `AltovaAutomation.jar`: Java classes to access Altova automation servers
- `StyleVisionActiveX.jar`: Java classes that wrap the StyleVision ActiveX interface
- `StyleVisionActiveX_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[1064].

Rules for mapping the ActiveX Control names to Java

For the documentation of ActiveX controls, see Object Reference[1086]. 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 StyleVision 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 StyleVisionControl, 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:
  `StyleVisionControl`: Java class to access the application
  `StyleVisionControlEvents`: Events interface for the StyleVisionControl
  `StyleVisionControlEventsDefaultHandler`: Default handler for `StyleVisionControlEvents`

Exceptions to mapping rules

There are some exceptions to the rules listed above. These are listed below:

Interface	Java name
`StyleVisionControlDocument`, method `New`	`newDocument`
`AuthenticView`, method `Goto`	`gotoElement`
`AuthenticRange`, method `Goto`	`gotoElement`
`AuthenticRange`, method `Clone`	`cloneRange`

This section

This section shows how some basic StyleVision ActiveX functionality can be accessed from Java code. It is organized into the following sub-sections:

- Example Java Project 1064
- Creating the ActiveX Controls 1066
- Loading Data in the Controls 1067
- Basic Event Handling 1067
- Menus 1068
- UI Update Event Handling 1070
- Creating a StyleVision Node Tree 1070

17.3.5.2.1 Example Java Project

The StyleVision 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 StyleVisionControl 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.

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

© 2016-2022 Altova GmbH

`.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.
`StyleVisionActiveX.jar`	Java classes of the StyleVision ActiveX control
`StyleVisionActiveX_JavaDoc.zip`	Javadoc file containing help documentation for the Java API
`StyleVisionContainer.java`	Java example source code
`StyleVisionContainerEventHandler.java`	Java example source code
`StyleVisionDialog.java`	Java example source code

What the example does

The example places one StyleVision document editor window, the StyleVision project window, the StyleVision schema tree window and the StyleVision design overview window in an AWT frame window. It reads out the main menu defined for StyleVision 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 1066: Starts StyleVision, which is registered as an automation server, or activates StyleVision if it is already running.
- Loading Data in the Controls 1067: Locates one of the example documents installed with StyleVision and opens it.
- Basic Event Handling 1067: Changes the view of all open documents to Text View. The code also shows how to iterate through open documents.
- Menus 1068: Validates the active document and shows the result in a message box. The code shows how to use output parameters.
- UI Update Event Handling 1070: Shows how to handle StyleVision events.
- Creating a StyleVision Node Tree 1070: Shows how to create a StyleVision node 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 **StyleVisionContainer.java** file. Namely, check that the following path refers to the actual folder where the StyleVision example files are stored on your operating system:

```
// Locate samples installed with the product.
```

```
final String strExamplesFolder = System.getenv( "USERPROFILE" ) + "\\Documents\\Altova\
\StyleVision2023\\StyleVisionExamples\\";
```

Running the example from the command line

To run the example from the command line:

1. Check that all prerequisites are met (see Prerequisites 1051).
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 `StyleVisionContainer.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 1051).
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**.

17.3.5.2.2 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.

```
01   /**
02    * StyleVision manager control - always needed
03    */
04   public static StyleVisionControl         styleVisionControl = null;
05
06   /**
07    * StyleVisionDocument editing control
08    */
09   public static StyleVisionControlDocument     styleVisionDocument = null;
```

```
10
11    /**
12     * Tool windows - StyleVision place-holder controls
13     */
14    private static StyleVisionControlPlaceHolder   styleVisionProjectToolWindow = null;
15    private static StyleVisionControlPlaceHolder   styleVisionDesignToolWindow = null;
16    private static StyleVisionControlPlaceHolder   styleVisionSchemaTreeToolWindow = null;
17
18    // Create the StyleVision ActiveX controls. First should be StyleVisionControl
      // determining that we want to place document controls and place-holder
19    // controls individually. It gives us full control over the menu, as well.
20       styleVisionControl = new StyleVisionControl(
         ICActiveXIntegrationLevel.ICActiveXIntegrationOnDocumentLevel.getValue() );
21
22       styleVisionDocument = new StyleVisionControlDocument();
23       frame.add( styleVisionDocument, BorderLayout.CENTER );
24
25
26    // Create a project window and open the sample project in it
27       styleVisionProjectToolWindow = new StyleVisionPlaceHolder(
         StyleVisionControlPlaceholderWindow.StyleVisionControlProjectWindowToolWnd.getValue
() );
28       styleVisionProjectToolWindow.setPreferredSize( new Dimension( 200, 200 ) );
```

17.3.5.2.3 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\\StyleVision2022\\StyleVisionExamples\\";
3        styleVisionProjectToolWindow.openProject( strExamplesFolder + "Examples.svp" );
```

17.3.5.2.4 Basic Event Handling

The code listing below shows how basic events can be handled. When calling the StyleVisionControl'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 StyleVisionDocumentControl'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              styleVisionControl.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          StyleVisionContainer.initStyleVisionDocument();
19          StyleVisionContainer.styleVisionDocument.open( i_strFileName );
20        } catch (Exception e) {
21          e.printStackTrace();
22        }
23      }
24    }
```

17.3.5.2.5 Menus

The code listing below shows how menu items can be created. Each `StyleVisionCommand` 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 `StyleVisionControl` object by calling its `exec` method. The `menuMap` object that is filled during menu creation is used later (see section UI Update Event Handling 1070).

```
01
02        // Load the file menu when the button is pressed
03        btnMenu.addActionListener( new ActionListener() {
04          public void actionPerformed(ActionEvent e) {
05            try {
06              // Create the menubar that will be attached to the frame
07              MenuBar mb = new MenuBar();
08              // Load the main menu's first item - the File menu
09              StyleVisionCommand xmlSpyMenu =
styleVisionControl.getMainMenu().getSubCommands().getItem( 0 );
10              // Create Java menu items from the Commands objects
11              Menu fileMenu = new Menu();
12              handlerObject.fillMenu( fileMenu, xmlSpyMenu.getSubCommands() );
13              fileMenu.setLabel( xmlSpyMenu.getLabel().replace( "&", "" ) );
14              mb.add( fileMenu );
15              frame.setMenuBar( mb );
16              frame.validate();
17            } catch (AutomationException e1) {
18              e1.printStackTrace();
19            }
20            // Disable the button when the action has been performed
21            ((AbstractButton) e.getSource()).setEnabled( false );
22          }
23        } ) ;
24    /**
25     * Populates a menu with the commands and submenus contained in a StyleVisionCommands
object
26     */
27    public void fillMenu(Menu newMenu, StyleVisionCommands styleVisionMenu) throws
AutomationException
28    {
```

```
29      // For each command/submenu in the xmlSpyMenu
30      for ( int i = 0 ; i < styleVisionMenu.getCount() ; ++i  )
31      {
32        StyleVisionCommand styleVisionCommand = styleVisionMenu.getItem( i );
33        if ( styleVisionCommand.getIsSeparator() )
34          newMenu.addSeparator();
35        else
36        {
37          StyleVisionCommands subCommands = styleVisionCommand.getSubCommands();
38          // Is it a command (leaf), or a submenu?
39          if ( subCommands.isNull() || subCommands.getCount() == 0 )
40          {
41            // Command -> add it to the menu, set its ActionCommand to its ID and store it
in the menuMap
42            MenuItem mi = new MenuItem( styleVisionCommand.getLabel().replace( "&",
"" ) );
43            mi.setActionCommand( "" + styleVisionCommand.getID() );
44            mi.addActionListener( this );
45            newMenu.add( mi );
46            menuMap.put( styleVisionCommand.getID(), mi );
47          }
48          else
49          {
50            // Submenu -> create submenu and repeat recursively
51            Menu newSubMenu = new Menu();
52            fillMenu( newSubMenu, subCommands );
53            newSubMenu.setLabel( styleVisionCommand.getLabel().replace( "&", "" ) );
54            newMenu.add( newSubMenu );
55          }
56        }
57      }
58    }
59      /**
60     * Action handler for the menu items
61     * Called when the user selects a menu item; the item's action command corresponds to
the command table for XMLSpy
62     */
63    public void actionPerformed( ActionEvent e )
64    {
65      try
66      {
67        int iCmd = Integer.parseInt( e.getActionCommand() );
68        // Handle explicitly the Close commands
69        switch ( iCmd )
70        {
71          case 57602:       // Close
72          case 34050:       // Close All
73            StyleVisionContainer.initStyleVisionDocument();
74            break;
75          default:
76            StyleVisionContainer.styleVisionControl.exec( iCmd );
77            break;
78        }
79      }
80      catch ( Exception ex )
81      {
82        ex.printStackTrace();
83      }
84
```

```
85    }
```

17.3.5.2.6 UI Update Event Handling

The code listing below shows how a UI-Update event handler can be created.

```
01    /**
02     * Call-back from the StyleVisionControl.
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( StyleVisionContainer.styleVisionControl.queryStatus( pair
.getKey() ) > 2 );
11    }
12
13    /**
14     * Call-back from the StyleVisionControl.
15     * Usually called while enabling/disabling commands due to UI updates
16     */
17    @Override
18    public boolean onIsActiveEditor( String i_strFilePath ) throws AutomationException
19    {
20      try {
21        return
StyleVisionContainer.styleVisionDocument.getDocument().getFullName().equalsIgnoreCase( i_st
rFilePath );
22      } catch ( Exception e ) {
23        return false;
24      }
25    }
```

17.3.5.2.7 Listing the Properties of a StyleVision Document

The listing below shows the properties of a StyleVision document (schemas, parameters, etc) can be loaded as nodes in a tree.

```
001 //access StyleVision Java-COM bridge
002 import com.altova.automation.StyleVision.*;
003
004 // access AWT/Swing components
005 import java.awt.*;
006 import javax.swing.*;
007 import javax.swing.tree.*;
008
009 /**
```


© 2016-2022 Altova GmbH

```
010  * A simple example of a tree control loading the structure from a StyleVision Document
object.
011  * The class receives a Document object, loads its nodes in a JTree, and prepares
012  * for modal activation.
013  *
014  * Feel free to modify and extend this sample.
015  *
016  * @author Altova GmbH
017  */
018 class StyleVisionDialog extends JDialog
019 {
020   /**
021    * The tree control
022    */
023   private JTree myTree;
024
025   /**
026    * Root node of the tree control
027    */
028   private DefaultMutableTreeNode top ;
029
030   /**
031    * Constructor that prepares the modal dialog containing the filled tree control
032    * @param xml   The data to be displayed in the tree
033    * @param parent  Parent frame
034    */
035   public StyleVisionDialog( Document doc, Frame parent )
036   {
037     // Construct the modal dialog
038     super( parent, "Data tree", true );
039     // Arrange controls in the dialog
040     top = new DefaultMutableTreeNode("root");
041     myTree = new JTree(top);
042     setContentPane( new JScrollPane( myTree ) );
043     // Build up the tree; expand all nodes, hide root node
044     fillTree( top, doc );
045     for(int i = 0 ; i < myTree.getRowCount() ; ++i )
046       myTree.expandRow( i );
047     myTree.setRootVisible( false );
048   }
049
050   /**
051    * Loads the nodes of an XML element under a given tree node
052    * @param node  Target tree node
053    * @param doc Source XML element
054    */
055   private void fillTree( DefaultMutableTreeNode node, Document doc)
056   {
057     try
058     {
059       DefaultMutableTreeNode titleNode = new
DefaultMutableTreeNode( "SchemaSources" ) ;
060       for (SchemaSource schema : doc.getSchemaSources())
061       {
062         String nodeText = "$" + schema.getName() ;
063         if ( schema.getIsMainSchemaSource() )
064           nodeText += " (main)";
065         DefaultMutableTreeNode newNode = new DefaultMutableTreeNode(nodeText ) ;
```

```
066         node.add( titleNode ) ;
067         titleNode.add( newNode );
068         String attribute = schema.getSchemaFileName() ;
069         if ( attribute.length() > 0 )
070         {
071           DefaultMutableTreeNode subNode = new DefaultMutableTreeNode( "SchemaFile" ) ;
072           subNode.add( new DefaultMutableTreeNode( attribute ) );
073           newNode.add( subNode );
074         }
075         attribute = schema.getWorkingXMLFileName() ;
076         if ( attribute.length() > 0 )
077         {
078           DefaultMutableTreeNode subNode = new DefaultMutableTreeNode( "Working
XML" ) ;
079           subNode.add( new DefaultMutableTreeNode( attribute ) );
080           newNode.add( subNode );
081         }
082         attribute = schema.getTemplateFileName() ;
083         if ( attribute.length() > 0 )
084         {
085           DefaultMutableTreeNode subNode = new DefaultMutableTreeNode( "Template" ) ;
086           subNode.add( new DefaultMutableTreeNode( attribute ) );
087           newNode.add( subNode );
088         }
089       }
090
091       titleNode = new DefaultMutableTreeNode( "Parameters" ) ;
092       for (Parameter param : doc.getParameters())
093       {
094         DefaultMutableTreeNode newNode = new DefaultMutableTreeNode( "$" +
param.getName() );
095         node.add( titleNode ) ;
096         titleNode.add( newNode );
097         DefaultMutableTreeNode subNode = new DefaultMutableTreeNode( "Value" ) ;
098         subNode.add( new DefaultMutableTreeNode( param.getValue() ) );
099         newNode.add( subNode );
100       }
101     }
102     catch (Exception e)
103     {
104       e.printStackTrace();
105     }
106   }
107
108 }
```

17.3.6 Command Reference

This section lists the names and identifiers of all menu commands that are available within StyleVision. Every sub-section lists the commands from the corresponding top-level menu of StyleVision. The command tables are organized as follows:

- The "Menu Item" column shows the command's menu text as it appears in StyleVision, 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 StyleVision installation directory.

© 2016-2022 Altova GmbH

- 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 StyleVisionControl.Exec[1093] or the StyleVisionControlDocument.Exec[1099] methods. To query the status of a command, use the StyleVisionControl.QueryStatus[1094] or StyleVisionControlDocument.QueryStatus[1100] methods.

Depending on the edition of StyleVision you have installed, some of these commands might not be supported.

17.3.6.1 "File" Menu

The "File" menu has the following commands:

Menu item	Command name	ID
New from XML Schema/DTD/XML...	IDC_FILE_NEW_FROM_SCHEMA	37579
New from DB...	IDC_FILE_NEW_FROM_DB	37577
New from XML column in DB table...	IDC_FILE_NEW_FROM_DBCELL	37861
New from XBRL Taxonomy...	IDC_FILE_NEW_FROM_XBRL_TAXONOMY	37912
New from HTML file...	IDC_FILE_NEW_FROM_HTML	37578
New from XSLT file...	IDC_FILE_NEW_FROM_XSLT	39737
New from Word 2007+ file...	IDC_FILE_NEW_FROM_DOCX	39814
New from Excel 2007+ file...	IDC_FILE_NEW_FROM_EXCEL	36054
New (empty)	IDC_FILE_NEW_EMPTY	37576
Open...	ID_FILE_OPEN	57601
Reload	IDC_FILE_RELOAD	36002
Close	ID_FILE_CLOSE	57602
Close All	ID_FILE_CLOSE_ALL	37777
Save Design	ID_FILE_SAVE	57603
Save As...	ID_FILE_SAVE_AS	57604
Save All	ID_FILE_SAVE_ALL	37778
Export as MobileTogether Design file...	IDC_FILE_EXPORT_AS_MTD	36055
Save Authentic XML Data	ID_SAVE_AUTHENTIC_XML	32860
Save Authentic XML Data as ...	ID_SAVE_AUTHENTIC_XML_AS	32812
Save Generated XSLT-HTML File...	IDC_SAVE_GEN_XSLT_HTML	37642

Menu item	Command name	ID
Save Generated HTML File(s)...	IDC_SAVE_GEN_HTML	37636
Save Generated XSLT-RTF File...	IDC_SAVE_GEN_XSLT_RTF	37643
Save Generated RTF File(s)...	IDC_SAVE_GEN_RTF	37638
Save Generated XSLT-FO File...	IDC_SAVE_GEN_XSLT_FO	37641
Save Generated FO File(s)...	IDC_SAVE_GEN_FO	37635
Save Generated PDF File(s)...	IDC_SAVE_GEN_PDF	37637
Save Generated XSLT-Word 2007+ File...	IDC_SAVE_GEN_XSLT_WORDML	37523
Save Generated Word 2007+ File(s)...	IDC_SAVE_GEN_WORDML	37529
Save Generated DB Schema...	IDC_SAVE_GEN_DB_SCHEMA	37633
Save Generated DB XML Data...	IDC_SAVE_GEN_DB_XML	37634
Save Generated User-Defined Schema...	IDC_SAVE_GEN_USERDEF_SCHEMA	37639
Save Generated User-Defined XML Data...	IDC_SAVE_GEN_USERDEF_XML	37640
Deploy to FlowForce...	IDC_STYLEVISIONGUI_DEPLOY_TO_FLOWFORCE	39787
Generate ASPX Web Application...	IDC_SAVE_SCRIPT_FOR_DBEXTRACT_AND_XSLTGEN_ASPX	39755
Properties...	IDC_FILE_PROPERTIES	36027
Print Preview...	ID_FILE_PRINT_PREVIEW	57609
Print...	ID_FILE_PRINT	57607
Recent File	ID_FILE_MRU_FILE1	57616
Exit	ID_APP_EXIT	57665

17.3.6.2 "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

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Copy	ID_EDIT_COPY	57634
Paste	ID_EDIT_PASTE	57637
Delete	ID_EDIT_CLEAR	57632
Find	ID_EDIT_FIND	57636
Find Next	ID_EDIT_FINDNEXT	36802
Replace...	ID_EDIT_REPLACE	57641
Stylesheet Parameters...	IDC_EDIT_PARAMETERS	37573
Collapse/Expand Markup	IDC_SPSGUI_COLLAPSE_EXPAND_MARKUP	36804
Select All	ID_EDIT_SELECT_ALL	57642

17.3.6.3 "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
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

Menu item	Command name	ID
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
Recent Project	IDC_ICPROJECTGUI_RECENT	37224

17.3.6.4 "View" Menu

The "View" menu has the following commands:

Menu item	Command name	ID
Dummy entry	ID_VIEW_TOOLBARS	37807
Project	ID_VIEW_PROJECT	57682
Design Overview	ID_VIEW_DESIGNOVERVIEW	37883
Schema Tree	ID_VIEW_SCHEMASOURCES	37805
Design Tree	ID_VIEW_DESIGNTREE	37804
Style Repository	ID_VIEW_STYLEREPOSITORY	37806
Context Properties	ID_VIEW_CONTEXTPROPERTY_COMMON	37802
Context Styles	ID_VIEW_CONTEXTPROPERTY_STYLES	37803
Show only one template at once	IDC_DESIGN_FILTER_ONE	37704
Show all template types	IDC_DESIGN_FILTER_ALL	37700
Show imported templates	IDC_DESIGN_FILTER_IMPORTED	37870
Show/Hide main template	IDC_DESIGN_FILTER_MAIN	37701
Show/Hide global templates	IDC_DESIGN_FILTER_MATCH	37702
Show/Hide design fragments	IDC_DESIGN_FILTER_NAMED	37703
Zoom In	IDC_SPSGUI_ZOOM_IN	36805
Zoom Out	IDC_SPSGUI_ZOOM_OUT	36806
Zoom 500%	IDC_SPSGUI_ZOOM_500	36807
Zoom 400%	IDC_SPSGUI_ZOOM_400	36808
Zoom 200%	IDC_SPSGUI_ZOOM_200	36809

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Zoom 150%	IDC_SPSGUI_ZOOM_150	36810
Zoom 100%	IDC_SPSGUI_ZOOM_100	36811
Zoom 75%	IDC_SPSGUI_ZOOM_75	36812
Zoom 50%	IDC_SPSGUI_ZOOM_50	36813
Status Bar	ID_VIEW_STATUS_BAR	59393

17.3.6.5 "Insert" Menu

The "Insert" menu has the following commands:

Menu item	Command name	ID
Insert Contents	IDC_INSERT_CONTENTS	37846
Insert Rest of Contents	IDC_INSERT_REST_OF_CONTENTS	37617
Insert RichEdit	IDC_INSERT_RICHEDIT	39773
Input Field	IDC_INSERT_EDITFIELD	37854
Multiline Input Field	IDC_INSERT_MULTILINEEDITFIELD	37855
Check Box...	IDC_INSERT_CHECKBOX	37857
Combo Box...	IDC_INSERT_COMBOBOX	37856
Radio Button...	IDC_INSERT_RADIOBUTTON	37858
Button	IDC_INSERT_BUTTON	37859
Navigation	IDC_INSERT_DBNAVIGATION	37603
Navigation + Goto	IDC_INSERT_DBNAVIGATIONGOTO	37604
Query Button	IDC_INSERT_DBQUERY	37605
Value...	IDC_INSERT_AUTOCALC_VALUE	37596
Input Field...	IDC_INSERT_AUTOCALC_FIELD	37594
Multiline Input Field...	IDC_INSERT_AUTOCALC_MULTILINE_FIELD	37595
Insert Date Picker	IDC_INSERT_DATEPICKER	37602
Insert Paragraph	IDC_INSERT_PARAGRAPH	37847
Address	IDC_INSERT_ADDRESS	37843

Menu item	Command name	ID
Block (div)	IDC_INSERT_BLOCK	37528
Blockquote	IDC_INSERT_BLOCKQUOTE	37527
Center	IDC_INSERT_CENTER	37534
Fieldset	IDC_INSERT_FIELDSET	37574
Preformatted	IDC_INSERT_FORMATTED	37580
Preformatted, wrapping	IDC_INSERT_FORMATTED_WRAP	37876
Heading 1 (h1)	IDC_INSERT_HEADING1	37585
Heading 2 (h2)	IDC_INSERT_HEADING2	37586
Heading 3 (h3)	IDC_INSERT_HEADING3	37587
Heading 4 (h4)	IDC_INSERT_HEADING4	37588
Heading 5 (h5)	IDC_INSERT_HEADING5	37589
Heading 6 (h6)	IDC_INSERT_HEADING6	37590
Insert Barcode...	IDC_INSERT_BARCODE	40221
Insert Chart...	IDC_INSERT_CHART	40223
Insert Image...	IDC_INSERT_IMAGE	37593
Insert Horizontal Line	IDC_INSERT_HORIZONTAL_LINE	37606
Insert Table...	IDC_INSERT_TABLE	40212
Insert Bullets and Numbering...	IDC_INSERT_FORMAT_BULLETS	37848
Insert Bookmark...	IDC_INSERT_BOOKMARK	37530
Insert Hyperlink...	IDC_INSERT_HYPERLINK	37849
Insert Footnote	IDC_INSERT_FOOTNOTE	39797
Insert Condition...	IDC_INSERT_CONDITION	37850
Insert Output-based Condition	IDC_INSERT_CONDITION_PER_OUTPUT	37851
Insert Template...	IDC_INSERT_TEMPLATE	40216
Insert User-Defined Template...	IDC_INSERT_USER_DEFINED_TEMPLATE	40184
Insert Variable Template...	IDC_INSERT_VARIABLE_TEMPLATE	37531
Dummy entry	IDC_INSERT_NAMED_TEMPLATE	39300
Insert Layout Container	IDC_INSERT_LAYOUT_CONTAINER	40213

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Insert Layout Box	IDC_INSERT_LAYOUT_BOX	40214
Insert Line	IDC_INSERT_SHAPE_LINE	40215
Table of Contents…	IDC_INSERT_TOC_WIZARD	37618
TOC Bookmark	IDC_INSERT_MARKER	37608
TOC Bookmark (Wizard)...	IDC_INSERT_MARKER_WIZARD	37609
TOC Reference	IDC_INSERT_REF	37613
Entry Text	IDC_INSERT_REF_ENTRY_TEXT	37614
Leader	IDC_INSERT_REF_LEADER	37615
Page Reference	IDC_INSERT_REF_PAGE_REFERENCE	37616
Hierarchical Numbering	IDC_INSERT_AUTO_NUMBER_LEVEL	37598
Sequential Numbering	IDC_INSERT_AUTO_NUMBER_FLAT	37597
TOC Level	IDC_INSERT_LEVEL	37607
TOC Level Reference	IDC_INSERT_REFLEVEL	37860
Template Serves as Level	IDC_ASSIGN_LEVEL	37524
Insert New Document	IDC_INSERT_RESULT_DOCUMENT	39756
New Page	IDC_INSERT_PAGE_BREAK	37610
Page Number	IDC_INSERT_PAGE_NUMBER	37611
Page Total	IDC_INSERT_PAGE_TOTAL	37612
New Column	IDC_INSERT_COLUMN_BREAK	37864
New Document Section	IDC_INSERT_DOCUMENT_SECTION	37951
Insert XBRL Table from Taxonomy…	IDC_INSERT_XBRL_TABLE_FROM_TAXONOMY	39813
Context	IDC_INSERT_XBRL_CONTEXT	37942
Item	IDC_INSERT_XBRL_ITEM	39793
Period	IDC_INSERT_XBRL_PERIOD	37909
Period Start	IDC_INSERT_XBRL_PERIOD_START	37947
Period End	IDC_INSERT_XBRL_PERIOD_END	37948
Identifier	IDC_INSERT_XBRL_IDENTIFIER	37910

Menu item	Command name	ID
Unit	IDC_INSERT_XBRL_UNITASPECT	39789
UnitRef	IDC_INSERT_XBRL_UNITREF	37946
Footnote	IDC_INSERT_XBRL_FOOTNOTE	37911
Non-XDT Segment	IDC_INSERT_XBRL_NONXDTSEGMENT_ASPECT	39790
Non-XDT Scenario	IDC_INSERT_XBRL_NONXDTSCENARIO_ASPECT	39791
XML Node	IDC_INSERT_XBRL_XMLNODE	39792
iXBRL Hidden Data	IDC_INSERT_IXBRL_HIDE	39721
iXBRL Fraction	IDC_INSERT_IXBRL_FRACTION	39723
User-Defined Element	IDC_STYLEVISIONGUI_INSERT_USERXMLELEM	37908
User-Defined Block	IDC_STYLEVISIONGUI_INSERT_USERXMLTEXT	37920

17.3.6.6 "Enclose with" Menu

The "Enclose with" menu has the following commands:

Menu item	Command name	ID
Template...	IDC_ENCLOSE_WITH_TEMPLATES	40180
User-Defined Template...	IDC_ENCLOSE_WITH_USER_DEFINED_TEMPLATES	40217
Variable Template...	IDC_ENCLOSE_WITH_VARIABLE_TEMPLATES	37520
Paragraph	IDC_ENCLOSE_PARAGRAPH	37627
Address	IDC_ENCLOSE_ADDRESS	40194
Block (div)	IDC_ENCLOSE_BLOCK	40195
Blockquote	IDC_ENCLOSE_BLOCKQUOTE	40196
Center	IDC_ENCLOSE_CENTER	40197
Fieldset	IDC_ENCLOSE_FIELDSET	40198
Preformatted	IDC_ENCLOSE_FORMATTED	40199

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Preformatted, wrapping	IDC_ENCLOSE_FORMATTED_WRAP	40200
Heading 1 (h1)	IDC_ENCLOSE_HEADING1	40201
Heading 2 (h2)	IDC_ENCLOSE_HEADING2	40202
Heading 3 (h3)	IDC_ENCLOSE_HEADING3	40203
Heading 4 (h4)	IDC_ENCLOSE_HEADING4	40204
Heading 5 (h5)	IDC_ENCLOSE_HEADING5	40205
Heading 6 (h6)	IDC_ENCLOSE_HEADING6	40206
Bullets and Numbering...	IDC_ENCLOSE_FORMAT_BULLETS	37581
Bookmark...	IDC_ENCLOSE_BOOKMARK	40207
Hyperlink...	IDC_ENCLOSE_HYPERLINK	37620
Condition...	IDC_ENCLOSE_CONDITION	37599
Output-based Condition	IDC_ENCLOSE_CONDITION_PER_OUTPUT	37600
TOC Bookmark	IDC_ENCLOSE_MARKER	40210
TOC Bookmark (Wizard)...	IDC_ENCLOSE_MARKER_WIZARD	40211
TOC Level	IDC_ENCLOSE_LEVEL	40208
TOC Level Reference	IDC_ENCLOSE_REFLEVEL	40209
New Document	IDC_ENCLOSE_RESULT_DOCUMENT	39758
User-Defined Element...	IDC_ENCLOSE_WITH_USER_DEFINED_XML_ELEMENT	40222

17.3.6.7 "Table" Menu

The "Table" menu has the following commands:

Menu item	Command name	ID
Insert Table...	IDC_INSERT_TABLE	40212
Delete Table	IDC_TABLE_DELETE	37658
Add Table Header Column	IDC_TABLE_ADD_HEADERCOL	37888
Add Table Footer Column	IDC_TABLE_ADD_FOOTERCOL	37889
Add Table Header Row	IDC_TABLE_ADD_HEADERROW	37900

Menu item	Command name	ID
Add Table Footer Row	IDC_TABLE_ADD_FOOTERROW	37901
Append Row	IDC_TABLE_APPEND_ROW	37657
Append Column	IDC_TABLE_APPEND_COL	37656
Insert Row	IDC_TABLE_INSERT_ROW	37664
Insert Column	IDC_TABLE_INSERT_COL	37662
Delete Row	IDC_TABLE_DELETE_ROW	37660
Delete Column	IDC_TABLE_DELETE_COL	37659
Join Cell Left	IDC_TABLE_JOIN_LEFT	37666
Join Cell Right	IDC_TABLE_JOIN_RIGHT	37667
Join Cell Below	IDC_TABLE_JOIN_DOWN	37665
Join Cell Above	IDC_TABLE_JOIN_UP	37668
Split Cell Horizontally	IDC_TABLE_SPLIT_HORZ	37670
Split Cell Vertically	IDC_TABLE_SPLIT_VERT	37671
View Cell Bounds	IDC_TABLE_SHOW_ZERO_BORDER	37669
View Table Markup	IDC_TABLE_SHOW_UICELLS	37887
Table Properties...	IDC_TABLE_EDIT_PROPERTIES	37661
Edit CALS/HTML Tables...	IDC_EDIT_XMLTABLES	37533

17.3.6.8 "Authentic" Menu

The "Authentic" menu has the following commands:

Menu item	Command name	ID
Edit Authentic Scripts...	IDC_EDIT_AUTHENTIC_SCRIPTPROJECT	37601
Custom Toolbar Buttons...	IDC_TEXT_STATE_ICONS	37672
Check Macro References...	IDC_AUTHENTIC_CHECK_MACRO_REFERENCES	39751
Auto-add Date Picker	IDC_TOGGLE_DATEPICKER_AUTOINSERT	37674
Auto-add DB Controls	IDC_TOGGLE_DBCONTROL_AUTOINSERT	37675
Reload Authentic View	IDC_AUTHENTICGUI_RELOAD	32800

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Validate XML	IDC_VALIDATE	32954
Select New Row with XML Data for Editing...	IDC_CHANGE_WORKING_DB_XML_CELL	32861
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
Justify	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_JUSTIFY	32821
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

Menu item	Command name	ID
Delete Row	IDC_ROW_DELETE	32807

17.3.6.9 "Database" Menu

The "Database" menu has the following commands:

Menu item	Command name	ID
Query Database	ID_VIEW_DBQUERY	37818
Edit DB Filter...	IDC_EDIT_DBFILTERS	37571
Clear DB Filter	IDC_CLEAR_DBFILTERS	37547

17.3.6.10 "Properties" Menu

The "Properties" menu has the following commands:

Menu item	Command name	ID
Edit Bullets and Numbering...	IDC_EDIT_BULLETS_AND_NUMBERING	37839
Predefined Value Formatting Strings...	IDC_FORMAT_PREDEFINES	37582

17.3.6.11 "Tools" Menu

The "Tools" menu has the following commands:

Menu item	Command name	ID
Spelling...	IDC_SPSGUI_SPELL_CHECK	36800
Spelling Options...	IDC_SPSGUI_SPELL_OPTIONS	36801
Global Resources	IDC_GLOBALRESOURCES	37401
<plugin not loaded>	IDC_GLOBALRESOURCES_SUBMENUENTRY1	37408
Customize...	IDC_APP_TOOLS_CUSTOMIZE	32959
Options...	IDC_TOOLS_OPTIONS	37676

© 2016-2022 Altova GmbH

17.3.6.12 "Window" Menu

The "Window" menu has the following commands:

Menu item	Command name	ID
Cascade	ID_WINDOW_CASCADE	57650
Tile Horizontal	ID_WINDOW_TILE_HORZ	57651
Tile Vertical	ID_WINDOW_TILE_VERT	57652
Arrange Icons	ID_WINDOW_ARRANGE	57649

17.3.6.13 "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
Software Activation...	IDC_ACTIVATION	32970
Order form...	IDC_OPEN_ORDER_PAGE	32971
Registration...	IDC_REGISTRATION	32972
Check for Updates...	IDC_CHECK_FOR_UPDATES	32973
StyleVision 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
StyleVision on the Internet...	IDC_OPEN_HOME_PAGE	32964
StyleVision Training...	IDC_OPEN_TRAINING_PAGE	32965
About StyleVision...	ID_APP_ABOUT	57664

17.3.7 Object Reference

Objects:
StyleVisionCommand 1086
StyleVisionCommands 1088
StyleVisionControl 1089
StyleVisionControlDocument 1097
StyleVisionControlPlaceHolder 1103

To give access to standard StyleVision functionality, objects of the **StyleVision automation interface** can be accessed as well. See StyleVisionControl.Application 1090, StyleVisionControlDocument.Document 1098 and StyleVisionControlPlaceHolder.Project 1104 for more information.

17.3.7.1 StyleVisionCommand

Properties:
`ID` 1087
`Label` 1087
`Name` 1087
`IsSeparator` 1087
`ToolTip` 1088
`StatusText` 1088
`Accelerator` 1086
`SubCommands` 1088

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

17.3.7.1.1 Accelerator

Property: `Accelerator` as string

© 2016-2022 Altova GmbH

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`.

17.3.7.1.2 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`[1093]) or query its status (using `QueryStatus`[1094]). If the command is a container for other commands (for example, a top-level menu), or a separator, the ID is 0.

17.3.7.1.3 IsSeparator

Property: `IsSeparator` as boolean

Description:
The property returns `true` if the command object is a menu separator; `false` otherwise. See also Command[1086].

17.3.7.1.4 Label

Property: `Label` as string

Description:
This property gets the text of the command as it is displayed in the graphical user interface of StyleVision. 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.

17.3.7.1.5 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 StyleVision installation.

17.3.7.1.6 StatusText

Property: `Label` as string

Description:
The status text is the text shown in the status bar of StyleVision 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.

17.3.7.1.7 SubCommands

Property: `SubCommands` as `Commands` 1088

Description:
The `SubCommands` property gets the collection of `Command` 1086 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`.

17.3.7.1.8 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.

17.3.7.2 StyleVisionCommands

Properties:
`Count` 1088
`Item` 1089

Description:
Collection of `Command` 1086 objects to get access to command labels and IDs of the StyleVisionControl. Those commands can be executed with the `Exec` 1093 method and their status can be queried with `QueryStatus` 1094.

17.3.7.2.1 Count

Property: `Count` as long

Description:
Number of `Command` 1086 objects on this level of the collection.

© 2016-2022 Altova GmbH

17.3.7.2.2 Item

Property: `Item` (`n` as long) as `Command` 1086

Description:
Gets the command with the index `n` in this collection. Index is 1-based.

17.3.7.3 StyleVisionControl

Properties:
`IntegrationLevel` 1091
`Appearance` 1090
`Application` 1090
`BorderStyle` 1090
`CommandsList` 1090
`EnableUserPrompts` 1091
`MainMenu` 1091
`Toolbars` 1092

Methods:
`Open` 1093
`Exec` 1093
`QueryStatus` 1094

Events:
`OnUpdateCmdUI` 1096
`OnOpenedOrFocused` 1096
`OnCloseEditingWindow` 1094
`OnFileChangedAlert` 1095
`OnContextChanged` 1095
`OnDocumentOpened` 1095
`OnValidationWindowUpdated` 1096

This object is a complete ActiveX control and should only be visible if the StyleVision library is used in the Application Level mode.

17.3.7.3.1 Properties

The following properties are defined:

`IntegrationLevel` 1091
`EnableUserPrompts` 1091
`Appearance` 1090
`BorderStyle` 1090

Command related properties:
`CommandsList` 1090
`MainMenu` 1091
`Toolbars` 1092

Access to StyleVisionAPI:
Application[1090]

17.3.7.3.1.1 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`.

17.3.7.3.1.2 Application

Property: `Application` as Application

Dispatch Id: *1*

Description:
The `Application` property gives access to the `Application` object of the complete StyleVision automation server API. The property is read-only.

17.3.7.3.1.3 BorderStyle

Property: `BorderStyle` as short

Dispatch Id: *-504*

Description:
A value of `1` displays the control with a thin border. Default value is `0`.

17.3.7.3.1.4 CommandsList

Property: `CommandList` as Commands[1088]=(read-only)

Dispatch Id: *1004*

Description:
This property returns a flat list of all commands defined available with StyleVisionControl. To get commands organized according to their menu structure, use MainMenu[1091]. To get toolbar commands, use Toolbars[1092].

```
public void GetAllStyleVisionCommands()
{
    // Get all commands from the StyleVision ActiveX control assigned to the current form
```


© 2016-2022 Altova GmbH

```
    StyleVisionControlLib.StyleVisionCommands commands =
this.axStyleVisionControl1.CommandList;
    // Iterate through all commands
    for (int i = 0; i < commands.Count; i++)
    {
      // Get each command by index and output it to the console
      StyleVisionControlLib.StyleVisionCommand cmd =
axStyleVisionControl1.CommandList[i];
      Console.WriteLine("{0} {1} {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&", ""));
    }
}
```

C# example

17.3.7.3.1.5 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*.

17.3.7.3.1.6 IntegrationLevel

Property: `IntegrationLevel` as `ICActiveXIntegrationLevel` 1106

Dispatch Id: *1000*

Description:
The `IntegrationLevel` property determines the operation mode of the control. See also Integration at Application Level 1054 and Integration at Document Level 1056 for more information.

Note: It is important to set this property immediately after the creation of the `StyleVisionControl` object.

17.3.7.3.1.7 MainMenu

Property: `MainMenu`=~ë=`` `çãã~åÇ `` 1086=(read-only)

Dispatch Id: *1003*

Description:
This property provides information about the structure and commands available in the StyleVisionControl main menu, as a `Command` object. The `Command` object contains all available submenus of StyleVision (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 GetStyleVisionMenus()
{
    // Get the main menu from the StyleVision ActiveX control assigned to the current
form
    StyleVisionControlLib.StyleVisionCommand mainMenu =
this.axStyleVisionControl1.MainMenu;

    // Loop through entries of the main menu (e.g. File, Edit, etc.)
    for (int i = 0; i < mainMenu.SubCommands.Count; i++)
    {
      StyleVisionControlLib.StyleVisionCommand menu = mainMenu.SubCommands[i];
      Console.WriteLine("{0} menu has {1} children items (including separators)",
menu.Label.Replace("&", ""), menu.SubCommands.Count);
    }
}
```

C# example

17.3.7.3.1.8 Toolbars

Property: `Toolbars` as `çãã~åÇë[1088]=(read-only)

Dispatch Id: *1005*

Description:
This property provides information about the structure of StyleVisionControl toolbars, as a `Command` object. The `Command` object contains all available toolbars of StyleVision. 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 GetStyleVisionToolbars()
{
    // Get the application toolbars from the StyleVision ActiveX control assigned to the
current form
    StyleVisionControlLib.StyleVisionCommands toolbars =
this.axStyleVisionControl1.Toolbars;

    // Iterate through all toolbars
    for (int i = 0; i < toolbars.Count; i++)
    {
      StyleVisionControlLib.StyleVisionCommand toolbar = toolbars[i];
      Console.WriteLine();
      Console.WriteLine("The toolbar \"{0}\" has the following commands:",
toolbar.Label);
```

 © 2016-2022 Altova GmbH

```
      // Iterate through all commands of this toolbar
      for (int j = 0; j < toolbar.SubCommands.Count; j++)
      {
         StyleVisionControlLib.StyleVisionCommand 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("&",
""));
         }
      }
   }
}
```

C# example

17.3.7.3.2 Methods

The following methods are defined:

Open 1093
Exec 1093
QueryStatus 1094

17.3.7.3.2.1 Exec

Method: `Exec` (`nCmdID` as long) as boolean

Dispatch Id: *6*

Description:
This method calls the StyleVision 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 1090. To retrieve the status of any command, use QueryStatus 1094.

17.3.7.3.2.2 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 StyleVisionControlDocument.Open[1100] and StyleVisionControlPlaceHolder.OpenProject[1104].

17.3.7.3.2.3 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 StyleVision command. If `QueryStatus` returns a value of `1` or `5`, the command is disabled.

17.3.7.3.3 Events

The StyleVisionControl ActiveX control provides the following connection point events:

OnUpdateCmdUI[1096]
OnOpenedOrFocused[1096]
OnCloseEditingWindow[1094]
OnFileChangedAlert[1095]
OnContextChanged[1095]

OnDocumentOpened[1095]
OnValidationWindowUpdated[1096]

17.3.7.3.3.1 OnCloseEditingWindow

Event: `OnCloseEditingWindow` (i_strFilePath as String) as boolean

Dispatch Id: *1002*

Description:
This event is triggered when StyleVision 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 StyleVisionControl should try to close the editor and destroy the associated document control.

 © 2016-2022 Altova GmbH

17.3.7.3.3.2 OnContextChanged

Event: `OnContextChanged` (`i_strContextName` as String, `i_bActive` as bool) as bool

Dispatch Id: *1004*

Description:
This event is not used in StyleVision

17.3.7.3.3.3 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 StyleVision 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 `StyleVisionControlDocument.OnDocumentOpened` 1102 instead.

17.3.7.3.3.4 OnFileChangedAlert

Event: `OnFileChangedAlert` (`i_strFilePath` as String) as bool

Dispatch Id: *1001*

Description:
This event is triggered when a file loaded with StyleVisionControl is changed on the hard disk by another application. Clients should return true, if they handled the event, or false, if StyleVision should handle it in its customary way, i.e. prompting the user for reload.

17.3.7.3.3.5 OnLicenseProblem

Event: `OnLicenseProblem` (`i_strLicenseProblemText` as String)

Dispatch Id: *1005*

Description:
This event is triggered when StyleVisionControl 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).

17.3.7.3.3.6 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 StyleVision.

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 StyleVisionControl, since internal access is required. Otherwise, the file can be opened with different editors.

17.3.7.3.3.7 OnToolWindowUpdated

Event: `OnToolWindowUpdated(pToolWnd` as long `)`

Dispatch Id: *1006*

Description:
This event is triggered when the tool window is updated.

17.3.7.3.3.8 OnUpdateCmdUI

Event: `OnUpdateCmdUI ()`

Dispatch Id: *1003*

Description:
Called frequently to give integrators a good opportunity to check status of StyleVision commands using `StyleVisionControl.QueryStatus`[1094]. Do not perform long operations in this callback.

17.3.7.3.3.9 OnValidationWindowUpdated

Event: `OnValidationWindowUpdated` ()

Dispatch Id: *3*

Description:
This event is triggered whenever the validation output window is updated with new information.

© 2016-2022 Altova GmbH

17.3.7.4 StyleVisionControlDocument

Properties:
Appearance 1098
BorderStyle 1098
Document 1098
IsModified 1098
Path 1098
ReadOnly 1099

Methods:
Exec 1099
New 1099
Open 1100
QueryStatus 1100
Reload 1100
Save 1101
SaveAs 1101

Events:
OnDocumentOpened 1102
OnDocumentClosed 1102
OnModifiedFlagChanged 1102
OnContextChanged 1101
OnFileChangedAlert 1102
OnActivate 1101

If the StyleVisionControl is integrated in the Document Level mode each document is displayed in an own object of type `StyleVisionControlDocument`. The `StyleVisionControlDocument` 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.

17.3.7.4.1 Properties

The following properties are defined:

ReadOnly 1099
IsModified 1098
Path 1098
Appearance 1098
BorderStyle 1098

Access to StyleVisionAPI:
Document 1098

17.3.7.4.1.1 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`.

17.3.7.4.1.2 BorderStyle

Property: `BorderStyle` as short

Dispatch Id: *-504*

Description:
A value of `1` displays the control with a thin border. Default value is `0`.

17.3.7.4.1.3 Document

Property: `Document` as Document

Dispatch Id: *1*

Description:
The `Document` property gives access to the `Document` object of the StyleVision automation server API. This interface provides additional functionality which can be used with the document loaded in the control. The property is read-only.

17.3.7.4.1.4 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.

17.3.7.4.1.5 Path

Property: `Path` as string

Dispatch Id: *1005*

© 2016-2022 Altova GmbH

Description:
Sets or gets the full path name of the document loaded into the control.

17.3.7.4.1.6 ReadOnly

Property: `ReadOnly` as boolean

Dispatch Id: *1007*

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.

17.3.7.4.2 Methods

The following methods are defined:

Document handling:
`New` 1099
`Open` 1100
`Reload` 1100
`Save` 1101
`SaveAs` 1101

Command Handling:
`Exec` 1099
`QueryStatus` 1100

17.3.7.4.2.1 Exec

Method: `Exec` (`nCmdID` as long) as boolean

Dispatch Id: *8*

Description:
`Exec` calls the StyleVision 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` 1091 property of StyleVisionControl. To get toolbar commands, use the `Toolbars` 1092 property of the StyleVisionControl.

17.3.7.4.2.2 New

Method: `New` () as boolean

Dispatch Id: *1000*

Description:
This method initializes a new document inside the control.

17.3.7.4.2.3 Open

Method: `Open` (`strFileName` as string) as boolean

Dispatch Id: *1001*

Description:
`Open` loads the file `strFileName` as the new document into the control.

17.3.7.4.2.4 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 StyleVision 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.

17.3.7.4.2.5 Reload

Method: `Reload` () as boolean

Dispatch Id: *1002*

Description:
`Reload` updates the document content from the file system.

© 2016-2022 Altova GmbH

17.3.7.4.2.6 Save

Method: `Save` () as boolean

Dispatch Id: *1003*

Description:
`Save` saves the current document at the location `Path`[1098].

17.3.7.4.2.7 SaveAs

Method: `SaveAs` (`strFileName` as string) as boolean

Dispatch Id: *1004*

Description:
`SaveAs` sets `Path`[1098] to *strFileName* and then saves the document to this location.

17.3.7.4.3 Events

The StyleVisionControlDocument ActiveX control provides following connection point events:

`OnDocumentOpened`[1102]
`OnDocumentClosed`[1102]
`OnModifiedFlagChanged`[1102]
`OnContextChanged`[1101]
`OnFileChangedAlert`[1102]
`OnActivate`[1101]
`OnSetEditorTitle`[1103]

17.3.7.4.3.1 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.

17.3.7.4.3.2 OnContextChanged

Event: `OnContextChanged` (`i_strContextName` as String, `i_bActive` as bool) as bool

Dispatch Id: *1004*

Description: None

17.3.7.4.3.3 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 StyleVision automation interface and should be used with care.

17.3.7.4.3.4 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 StyleVision automation interface, and can be used to query for more details about the document, or perform additional operations.

17.3.7.4.3.5 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.

17.3.7.4.3.6 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 StyleVision should handle it in its customary way, i.e. prompting the user for reload.

17.3.7.4.3.7 OnModifiedFlagChanged

Event: `OnModifiedFlagChanged` (`i_bIsModified` as boolean)

© 2016-2022 Altova GmbH

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.

17.3.7.4.3.8 OnSetEditorTitle

Event: OnSetEditorTitle ()

Dispatch Id: *1006*

Description:
This event is being raised when the contained document is being internally renamed.

17.3.7.5 StyleVisionControlPlaceHolder

Properties available for all kinds of placeholder windows:
`PlaceholderWindowID` 1104

Properties for project placeholder window:
`Project` 1104

Methods for project placeholder window:
`OpenProject` 1104
`CloseProject` 1105

The `StyleVisionControlPlaceHolder` control is used to show the additional StyleVision windows like Overview, Library or Project window. It is used like any other ActiveX control and can be placed anywhere in the client application.

17.3.7.5.1 Properties

The following properties are defined:

`PlaceholderWindowID` 1104

Access to StyleVisionAPI:
`Project` 1104

17.3.7.5.1.1 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.

17.3.7.5.1.2 PlaceholderWindowID

Property: `PlaceholderWindowID as StyleVisionControlPlaceholderWindow` 1106

Dispatch Id: *1*

Description:
This property specifies which StyleVision 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 `StyleVisionControlPlaceholderWindow` 1106 enumeration. The control changes its state immediately and shows the new StyleVision window.

17.3.7.5.1.3 Project

Property: `Project` as Project (read-only)

Dispatch Id: *2*

Description:
The `Project` property gives access to the `Project` object of the StyleVision 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` 1104 with a value of `StyleVisionXProjectWindow (=3)`. The property is read-only.

17.3.7.5.2 Methods

The following method is defined:

`OpenProject` 1104
`CloseProject` 1105

17.3.7.5.2.1 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`** 1104 different to `StyleVisionXProjectWindow (=3)`.

© 2016-2022 Altova GmbH

17.3.7.5.2.2 CloseProject

Method: `CloseProject ()`

Dispatch Id: *4*

Description:
`CloseProject` closes the project loaded by the control. The method will fail if the placeholder window has a `PlaceholderWindowID` 1104 different to `StyleVisionXProjectWindow (=3)`.

17.3.7.5.3 Events

The StyleVisionControlPlaceholder ActiveX control provides following connection point events:

`OnModifiedFlagChanged` 1105

17.3.7.5.3.1 OnModifiedFlagChanged

Event: `OnModifiedFlagChanged` (`i_bIsModified` as boolean)

Dispatch Id: *1*

Description:
This event gets triggered only for placeholder controls with a `PlaceholderWindowID` 1104 of `StyleVisionXProjectWindow (=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.

17.3.7.5.3.2 OnSetLabel

Event: `OnSetLabel(i_strNewLabel` as string)

Dispatch Id: *1000*

Description:
Raised when the title of the placeholder window is changed.

17.3.7.6 Enumerations

The following enumerations are defined:

`ICActiveXIntegrationLevel` 1106
`StyleVisionControlPlaceholderWindow` 1106

17.3.7.6.1 ICActiveXIntegrationLevel

Possible values for the IntegrationLevel[1091] property of the StyleVisionControl.

```
ICActiveXIntegrationOnApplicationLevel  = 0
ICActiveXIntegrationOnDocumentLevel     = 1
```

17.3.7.6.2 StyleVisionControlPlaceholderWindow

This enumeration contains the list of the supported additional StyleVision windows.

```
StyleVisionControlNoToolWnd       = -1
StyleVisionControlProjectWnd            = 0
StyleVisionControlDesignOverviewWnd     = 1
StyleVisionControlSchemaSourcesWnd= 2
StyleVisionControlDesignTreeWnd          = 3
StyleVisionControlStyleRepositoryWnd     = 4
StyleVisionControlContextPropertiesWnd   = 5
StyleVisionControlContextStylesWnd= 6
StyleVisionControlMessageWnd            = 7
StyleVisionControlDBQueryWnd             = 8
```


© 2016-2022 Altova GmbH

18 Anexos

Estos anexos contienen (i) información sobre los motores XSLT utilizados en StyleVision, (ii) información sobre la conversión de tipos de datos de BD en tipos de datos XML Schema, (iii) información técnica sobre la aplicación e (iv) información importante sobre las licencias de StyleVision. Cada anexo contiene varios apartados.

Información sobre motores XSLT 1108
Ofrece información sobre la implementación de los motores XSLT de Altova que StyleVision utiliza para generar documentos de salida.

- Motores XSLT 1.0, 2.0 y 3.0 de Altova
- Funciones XSLT y XPath/XQuery

Conversión de tipos de datos de BD y XML Schema 1217
Cuando se convierte un campo de BD en un nodo XML, los tipos de datos de BD se convierten en tipos de datos XML Schema. Este anexo incluye tablas de correspondencia para la conversión de tipos de datos de estas bases de datos:

- MS Access
- MS SQL Server
- MySQL
- Oracle
- ODBC
- ADO
- Sybase

Datos técnicos 1224
Información técnica sobre StyleVision

- Requisitos de sistema operativo y memoria
- Analizador XML de Altova
- Motores XSLT y XQuery de Altova
- Compatibilidad con Unicode
- Uso de Internet

Información de la licencia 1227
Información sobre cómo se distribuye StyleVision y sus licencias.

- Distribución electrónica de software
- Activación del software y medición de licencias
- Derechos de autor
- Contrato de licencia para el usuario final

18.1 Información sobre motores XSLT y XQuery

Los motores XSLT y XQuery de StyleVision 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, StyleVision 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 1108
- XSLT 2.0 1108
- XSLT 3.0 1110
- XQuery 1.0 1111
- XQuery 3.1 1113

18.1.1 XSLT 1.0

El motor XSLT 1.0 de StyleVision 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 (` `).

18.1.2 XSLT 2.0

Temas de este apartado:

- Especificaciones con las que cumple el motor 1109
- Compatibilidad con versiones antiguas 1109
- Espacios de nombres 1109

© 2016-2022 Altova GmbH

- Compatibilidad con esquemas 1110
- Comportamiento propio de esta implementación 1110

Especificaciones

El motor XSLT 2.0 de StyleVision 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`.

18.1.3 XSLT 3.0

El motor XSLT 3.0 de StyleVision 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 1108. 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

18.1.4 XQuery 1.0

Temas de este apartado:

- Especificaciones con las que cumple el motor 1111
- Compatibilidad con esquemas 1111
- Codificación 1111
- Espacios de nombres 1111
- Fuentes XML y validación 1112
- Comprobación de tipos estática y dinámica 1112
- Módulos biblioteca 1112
- Funciones externas 1113
- Intercalaciones 1113
- Precisión de datos numéricos 1113
- Compatibilidad con instrucciones XQuery 1113

Especificaciones compatibles

El motor XQuery 1.0 de StyleVision 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 StyleVision.

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[1114]. Para usar una intercalación concreta, indique su URI tal y como aparece en la lista de intercalaciones compatibles[1114]. 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.

18.1.5 XQuery 3.1

El motor XQuery 3.1 de StyleVision 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[1111].

18.2 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()`.

18.2.1 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 1116

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 1119
- Funciones de geoubicación 1137
- Funciones relacionadas con imágenes 1149
- Funciones numéricas 1153
- Funciones de secuencia 1176
- Funciones de cadena 1184
- Funciones varias 1192

Funciones para códigos de barras

Las funciones de extensión para códigos de barras de Altova 1194 permiten generar códigos de barras y colocarlos en los resultados generados con hojas de estilos XSLT.

18.2.1.1 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 conseguirá funciones adicionales no disponibles en la biblioteca de funciones estándar de XPath, XQuery y

© 2016-2022 Altova GmbH

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

- Variables estáticas: `<xsl:value-of select="$i3, $i2, $i1" />`

© 2016-2022 Altova GmbH

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 1116]

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 1116]

18.2.1.2 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 1121
- Agregar una duración a xs:date y devolver xs:date 1123
- Agregar una duración a xs:time y devolver xs:time 1125
- Recuperar duraciones y aplicarles formato 1124
- Quitar la zona horaria de las funciones que generan la fecha/hora actual 1126
- Devolver el número de días, horas, minutos y segundos de duraciones 1127
- Devolver el día de la semana de una fecha como número entero 1129
- Devolver el número de semana de una fecha como número entero 1130
- Generar la fecha, la hora y el tipo de duración a partir de los componentes léxicos de cada tipo 1131
- Construir un tipo date, dateTime o a partir de la cadena de entrada 1133
- Funciones para calcular la edad 1134
- Funciones para calcular el tiempo Unix 1136

Funciones por orden alfabético

`altova:add-days-to-date` 1123
`altova:add-days-to-dateTime` 1121
`altova:add-hours-to-dateTime` 1121
`altova:add-hours-to-time` 1125
`altova:add-minutes-to-dateTime` 1121
`altova:add-minutes-to-time` 1125
`altova:add-months-to-date` 1123
`altova:add-months-to-dateTime` 1121
`altova:add-seconds-to-dateTime` 1121
`altova:add-seconds-to-time` 1125
`altova:add-years-to-date` 1123
`altova:add-years-to-dateTime` 1121
`altova:age` 1134
`altova:age-details` 1134

 © 2016-2022 Altova GmbH

altova:build-date 1131
altova:build-duration 1131
altova:build-time 1131
altova:current-dateTime-no-TZ 1126
altova:current-date-no-TZ 1126
altova:current-time-no-TZ 1126
altova:date-no-TZ 1126
altova:dateTime-from-epoch 1136
altova:dateTime-from-epoch-no-TZ 1136
altova:dateTime-no-TZ 1126
altova:days-in-month 1127
altova:epoch-from-dateTime 1136
altova:hours-from-dateTimeDuration-accumulated 1127
altova:minutes-from-dateTimeDuration-accumulated 1127
altova:seconds-from-dateTimeDuration-accumulated 1127
altova:format-duration 1124
altova:parse-date 1133
altova:parse-dateTime 1133
altova:parse-duration 1124
altova:parse-time 1133
altova:time-no-TZ 1126
altova:weekday-from-date 1129
altova:weekday-from-dateTime 1129
altova:weeknumber-from-date 1130
altova:weeknumber-from-dateTime 1130

[Subir 1119]

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 1119]

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 1119]

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 1119]

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 1119]

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() asxs: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 1119]

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 1119]

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*)

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 1119]

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 1119]

© 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 1119]

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 1119]

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 1119]

18.2.1.3 Funciones XPath/XQuery: de geoubicación

Las funciones de extensión XPath/XQuery de geoubicación son compatibles con la versión actual de StyleVision 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` 1149 y el atributo `@Geolocation` 1149 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` `(48.858222222222, 24.2945)`
- **`altova:parse-geolocation`**`('48°51''29.6"N  24°17''40.2"')` devuelve la secuencia de dos `xs:decimals` `(48.858222222222, 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` 1149 y el atributo de metadatos Exif `@Geolocation` 1149 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 1149 y el atributo de metadatos Exif @Geolocation 1149 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` [1149] 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 1149 y el atributo de metadatos Exif @Geolocation 1149 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 1149 y el atributo de metadatos Exif @Geolocation 1149 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 1137]

18.2.1.4 Funciones XPath/XQuery: relacionadas con imágenes

Las funciones de extensión XPath/XQuery para trabajar con imágenes son compatibles con la versión actual de StyleVision 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

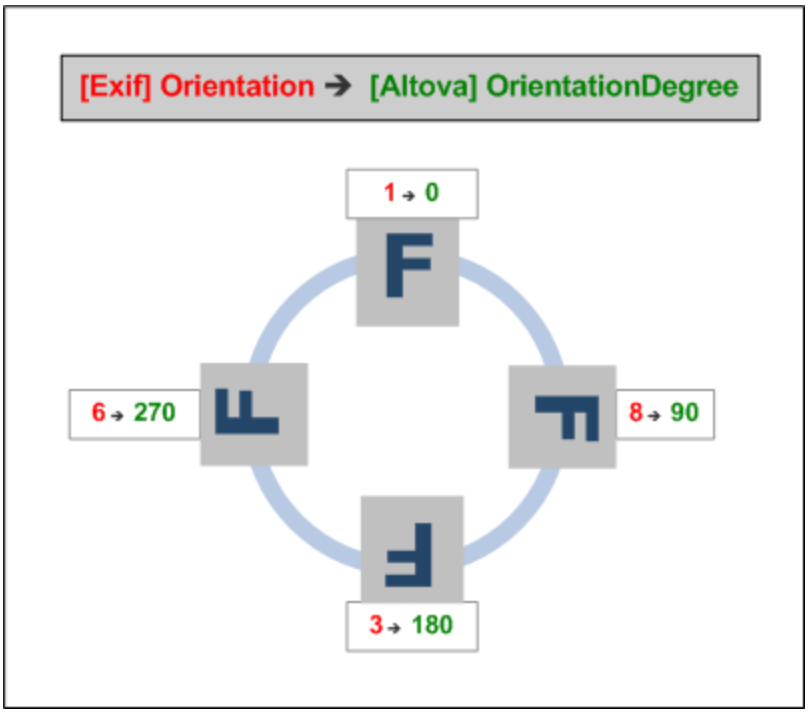
Atributo Exif de Altova: OrientationDegree

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** 1149]

18.2.1.5 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 1153]

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 1153]

Funciones de formato numérico

[Subir 1153]

18.2.1.6 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

18.2.1.7 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:

```
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...

```
(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)

18.2.1.8 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"

18.2.1.9 Funciones XPath/XQuery varias

Estas funciones de extensión XPath/XQuery generales son compatibles con la versión actual de StyleVision 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

 © 2016-2022 Altova GmbH

- **altova:decode-string**($XML1/MailData/Meta/b64B) devuelve la entrada en base64Binary como cadena de texto cifrada en UTF-8
- **altova:decode-string**($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 '1.1676614615456 6E6'

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 1192]

18.2.1.10 Funciones para códigos de barras

Los motores XSLT de Altova usan bibliotecas Java de terceros para crear códigos de barras. A continuación enumeramos las clases y los métodos públicos utilizados. Las clases se empaquetan en AltovaBarcodeExtension.jar, que está en la carpeta *<CarpetaArchivosPrograma>*\Altova\Common2023\jar.

Las bibliotecas Java utilizadas están en las subcarpetas de la carpeta *<CarpetaArchivosPrograma>*\Altova\Common2023\jar:

- barcode4j\barcode4j.jar (sitio web: http://barcode4j.sourceforge.net/)
- zxing\core.jar (sitio web: http://code.google.com/p/zxing/)

Los archivos de licencia están también en estas carpetas.

© 2016-2022 Altova GmbH

Equipo virtual Java

Para poder usar las funciones de códigos de barras debe existir un equipo virtual en su equipo. A continuación se describe cómo encontrar la ruta de acceso a ese equipo.

- 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**.

Ejemplo XSLT

A continuación puede ver un ejemplo de XSLT que ilustra el uso de funciones para códigos de barras en una hoja de estilos XSLT.

```
<?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:altova="http://www.altova.com"
   xmlns:altovaext="”http://www.altova.com/xslt-extensions”
   xmlns:altovaext-barcode="java:com.altova.extensions.barcode.BarcodeWrapper"
   xmlns:altovaext-barcode-
property="java:com.altova.extensions.barcode.BarcodePropertyWrapper">
   <xsl:output method="html" encoding="UTF-8" indent="yes"/>
   <xsl:template match="/">
      <html>
         <head><title/></head>
         <body>
            <img alt="barcode" src="{altovaext:get-temp-folder()}barcode.png"/>
         </body>
      </html>
      <xsl:result-document
         href="{altovaext:get-temp-folder()}barcode.png"
         method="text" encoding="base64tobinary" >
         <xsl:variable name="barcodeObject"
            select="altovaext-barcode:newInstance(&apos;Code39&apos;,string(&apos;some
value&apos;),
            96,0, (altovaext-barcode-property:new( &apos;setModuleWidth&apos;, 25.4 div 96
* 2 ) ) )"/>
         <xsl:value-of select="xs:base64Binary(xs:hexBinary(string(altovaext-
barcode:generateBarcodePngAsHexString($barcodeObject)) ))"/>
      </xsl:result-document>
   </xsl:template>
</xsl:stylesheet>
```

Paquete com.altova.extensions.barcode

El paquete com.altova.extensions.barcode se utiliza para generar la mayoría de códigos de barras.

Se utilizan estas clases:

```
public class BarcodeWrapper
       static BarcodeWrapper newInstance( String name, String msg, int dpi, int
orientation, BarcodePropertyWrapper[] arrProperties )
       double getHeightPlusQuiet()
       double getWidthPlusQuiet()
       org.w3c.dom.Document generateBarcodeSVG()
       byte[] generateBarcodePNG()
       String generateBarcodePngAsHexString()
```

public class **BarcodePropertyWrapper** *Utilizada para almacenar las propiedades del código de barras que se establecerán más tarde de forma dinámica*

```
    BarcodePropertyWrapper( String methodName, String propertyValue )
       BarcodePropertyWrapper( String methodName, Integer propertyValue )
       BarcodePropertyWrapper( String methodName, Double propertyValue )
       BarcodePropertyWrapper( String methodName, Boolean propertyValue )
       BarcodePropertyWrapper( String methodName, Character propertyValue )
       String getMethodName()
       Object getPropertyValue()
```

public class **AltovaBarcodeClassResolver** *Registra la clase* com.altova.extensions.barcode.proxy.zxing.QRCodeBean *para el bean* qrcode*, además de las clases registradas por* org.krysalis.barcode4j.DefaultBarcodeClassResolver.

Ejemplo XQuery para generar códigos QR

A continuación puede ver un ejemplo de XQuery en el que se ven las funciones de código de barras que puede usar para generar una imagen de código QR.

```
declare variable $lines := unparsed-text-
lines('https://info.healthministry.gv.at/data/timeline-cases-provinces.csv', 'utf-8');
declare variable $main := map:merge(tokenize(head($lines), ';')!map{.:position()});
declare variable $data := map:merge(tail($lines)!array{tokenize(., ';')}!map{?($main?Name):
[?($main?Date), xs:integer(?($main?ConfirmedCasesProvinces)) - xs:integer(?($main?
Recovered))]}, map{'duplicates':'combine'});
declare variable $chart_img := altovaext:chart(map{'width': 1900, 'height': 600}, map:for-
each($data, function($k, $v){[$k, $v?1!substring-before(., 'T'), $v?2][$k != 'Austria']}));
<img src="{$chart_img}"/>
(:$main,  $data,:)
```

Paquete com.altova.extensions.barcode.proxy.zxing

El paquete com.altova.extensions.barcode.proxy.zxing se utiliza para generar códigos de barras de tipo QRCode.

Se utilizan estas clases:

© 2016-2022 Altova GmbH

class **QRCodeBean**

- *Extiende* `org.krysalis.barcode4j.impl.AbstractBarcodeBean`
- *Crea una interfaz* `AbstractBarcodeBean` *para* `com.google.zxing.qrcode.encoder`

```
void generateBarcode(CanvasProvider canvasImp, String msg)
void setQRErrorCorrectionLevel(QRCodeErrorCorrectionLevel level)
BarcodeDimension calcDimensions(String msg)
double getVerticalQuietZone()
double getBarWidth()
```

class **QRCodeErrorCorrectionLevel** *Nivel de corrección de errores para* `QRCode`

```
static QRCodeErrorCorrectionLevel byName(String name)
"L" = ~7% correction
"M" = ~15% correction
"H" = ~25% correction
"Q" = ~30% correction
```

18.2.2 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 1198 y .NET 1207, así como scripts MSXSL para XSLT 1213.

Esta sección describe cómo usar funciones de extensión y scripts MSXSL en hojas de estilos XSLT. Las funciones de extensión pueden organizarse en varios grupos:

- Funciones de extensión Java 1198
- Funciones de extensión .NET 1207
- Scripts MSXSL para XSLT 1213

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.

18.2.2.1 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 1203
- Métodos estáticos y campos estáticos 1204
- Métodos de instancia y campos de instancia 1205
- Tipos de datos: conversión de XPath/XQuery en Java 1206
- Tipos de datos: conversión de Java en XPath/XQuery 1207

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 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.

 © 2016-2022 Altova GmbH

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 1199 y Archivos Jar definidos por el usuario 1202. 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.

Nota: si desea añadir un espacio de nombres a una hoja de estilos generada desde un archivo SPS creado con StyleVision, el espacio de nombres se debe añadir al elemento de nivel superior `schema` del esquema XML en el que se basa el archivo SPS. Recuerde que la declaración de espacio de nombres `xmlns:java="java"` se crea automáticamente por defecto en todos los SPS creados con StyleVision.

18.2.2.1.1 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* 1200.)
- 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* 1200.)

- El archivo de clases está en un paquete. El archivo XSLT/XQuery está en una carpeta cualquiera. (*ver ejemplo*[1201].)
- El archivo de clases no está en un paquete. El archivo XSLT/XQuery está una carpeta cualquiera. (*ver ejemplo*[1202].)

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>
```

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"
```


© 2016-2022 Altova GmbH

```
        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.

18.2.2.1.2 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:

© 2016-2022 Altova GmbH

`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.

18.2.2.1.3 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 [1207], 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* 1205):
```
<xsl:value-of select="date:toString(date:new())" xmlns:date="java:java.util.Date" />
```

18.2.2.1.4 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 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)" />
```

 © 2016-2022 Altova GmbH

Ejemplo de XQuery

Un ejemplo de XQuery similar sería:

```
<cosine xmlns:jMath="java:java.lang.Math">
   {jMath:cos(3.14)}
</cosine>
```

18.2.2.1.5 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`.

18.2.2.1.6 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)`.
- 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.

 © 2016-2022 Altova GmbH

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.

18.2.2.1.7 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).

18.2.2.2 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 1209
- Métodos estáticos y campos estáticos 1210
- Métodos de instancia y campos de instancia 1211
- Tipos de datos: conversión de XPath/XQuery en .NET 1212
- Tipos de datos: conversión de .NET en XPath/XQuery 1213

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"
```

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.

18.2.2.2.1 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 [1207], 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* [1205]):
```
<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" />
```

18.2.2.2.2 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()`.

Ejemplos

Este ejemplo de código XSLT muestra una llamada a un método con un argumento (`System.Math.Sin(arg)`):

© 2016-2022 Altova GmbH

```
<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>
```

18.2.2.2.3 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.

18.2.2.2.4 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
xs:float	FloatValue, float, double
xs:double	DoubleValue, double

© 2016-2022 Altova GmbH

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.

18.2.2.2.5 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).

18.2.2.3 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

18.3 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 1217
- MS Access 1218
- MS SQL Server 1219
- MySQL 1220
- ODBC 1220
- Oracle 1221
- Sybase 1222

18.3.1 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

18.3.2 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

18.3.3 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

18.3.4 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

18.3.5 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

18.3.6 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.

18.3.7 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

18.4 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 1224
- Motores de Altova 1224
- Compatibilidad con Unicode 1225
- Uso de Internet 1225

18.4.1 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.

18.4.2 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.

18.4.3 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.

18.4.4 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 StyleVision, 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

18.5 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.

18.5.1 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.

18.5.2 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.

18.5.3 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 (solo en la edición Enterprise Edition), 21, 39
.NET,
diferencias del complemento para VS, 710
integración de StyleVision con, 708

A

Abreviaturas,
utilizadas en el manual del usuario, 33
Abrir,
archivos usados recientemente, 792
Abrir (SPS), 768
Abrir proyecto, 803
Acoplar,
las ventanas de los ayudantes de entrada, 41
Activar el software, 901
ActiveX,
integration at application level, 1054
integration at document level, 1056
integration prerequisites, 1051
ActiveX controls,
adding to the Visual Studio Toolbox, 1052
Actualizar nodos,
con el valor de cálculos automáticos, 298
Actualizar nodos (ediciones Enterprise y Professional Edition),
con el resultado de un cálculo automático, 293
con el valor de cálculos automáticos, 296
ADO,
como interfaz de conexión de datos, 523
configurar una conexión, 529
conversión de tipos de datos en esquemas XML generados a partir de BD, 1217
ADO.NET,
configurar una conexión, 534
Agregar archivo activo al proyecto, 806
Agregar archivo activo y relacionados al proyecto, 806
Agregar archivos al proyecto, 804
Agregar carpeta de proyecto al proyecto, 806
Agregar carpeta externa / carpeta web al proyecto, 806
Agregar esquema, 761
Agregar nombre, 462
Agregar recurso de Altova al proyecto, 804
Agregar URL al proyecto, 804
Agrupación, 312
ejemplo de agrupar-por (Persons.sps), 315
ejemplo de agrupar-por (Scores.sps), 316
Alertas de modificación de archivos,
en archivos SPS modulares, 255
Alias,
ver Recursos globales, 444
Alineación vertical del contenido de las celdas de la tabla,
en el diseño SPS, 848
Alinear el contenido de las celdas de la tabla,
en el diseño SPS, 848
AltovaRowStatus,
en diseños SPS basados en bases de datos, 628
Ambigüedades,
del modelo de contenido, 138
Añadir selector de fecha automáticamente, 855
Anexar,
columna en una tabla en el diseño SPS, 845
fila en una tabla en el diseño SPS, 845
Anexos, 1107
Año base,
en formato de entrada, 380
API,
documentation, 931
overview, 932
API de XMLSpy,
documentation, 931
overview, 932
Aplicación ASP.NET, 509
Aplicación web ASPX, 509
Aplicar formato,
a nodos al insertarlos, 137
resumen de procedimientos, 374
ver Presentación, 374
Application,
ActiveDocument, 953
Application, 953
Documents, 953
Parent, 955
Quit, 956
Archivo (menú), 761
Archivo | Abrir, 768

Archivo (menú), 761
Archivo | Cerrar, 768
Archivo | Codificación, 787
Archivo | Guardar archivos generados, 782
Archivo | Guardar como, 778
Archivo | Guardar diseño, 773
Archivo | Guardar los datos XML de Authentic, 782
Archivo | Imprimir, 791
Archivo | Nuevo, 761
Archivo | Vista previa de impresión, 791
comando Salir, 792
Archivo de plantilla XML (ediciones Enterprise y Professional Edition), 113
definición, 29
Archivo SPS modular,
activar y desactivar, 255
agregar el módulo SPS, 255
archivos de plantilla XML, 252
archivos XML de trabajo, 252
crear, 255
declaraciones de espacio de nombres, 252
efecto en el orden de prioridad, 255
ejemplo de proyecto, 259
esquemas fuente, 252
estilos CSS, 252
módulo SPS que se debe agregar, 255
objetos modulares, 252
scripts, 252
trabajar con, 255
y alertas de modificación de archivos, 255
y esquemas fuente, 255
y estilos CSS, 255
Archivo XML,
con datos de BD, 632
Archivo XML de recursos globales, 444
Archivo XML de trabajo, 48, 113
definición, 29
en la vista Authentic, 38
en las vistas de resultados, 39
imprimir, 791
validar en la vista Authentic, 856
vista previa de impresión, 791
Archivo XML para diseños SPS basados en bases de datos, 617
Archivos,
abrir archivos usados recientemente, 792
Archivos CSS,
gestión en la ventana Vista general del diseño, 45
Archivos de origen para el SPS, 113
Archivos de salida,
de hojas de estilos SPS basadas en BD, 632
generar, 121
usar la línea de comandos para generar, 700
Archivos generados, 121
Archivos por lotes,
para generar archivos desde la línea de comandos, 700
y tareas programadas, 707
Archivos PXF, 512
crear, 512
editar, 515
guardar como, 778
implementar, 516
Archivos recientes, 792
Archivos SPS modulares,
resumen, 251
Archivos XSLT,
usar la línea de comandos para generar, 700
Asignar formatos predefinidos,
en el tutorial Quick Start, 86
Atributos (ayudante de entrada),
en la vista Authentic, 654
Atributos de clase,
en el tutorial Quick Start, 86
Authentic (menú), 850
Authentic Browser, 25
Authentic Desktop, 25
AuthenticRange,
AppendRow, 971
Application, 972
CanPerformAction, 972
CanPerformActionWith, 973
Close, 973
CollapsToBegin, 973
CollapsToEnd, 974
Copy, 974
Cut, 974
Delete, 974
DeleteRow, 975
DuplicateRow, 975
ExpandTo, 976
FirstTextPosition, 976
FirstXMLData, 977
FirstXMLDataOffset, 978
GetElementAttributeNames, 979
GetElementAttributeValue, 980
GetElementHierarchy, 980

AuthenticRange,
GetEntityNames, 980
Goto, 981
GotoNext, 982
GotoNextCursorPosition, 982
GotoPrevious, 983
GotoPreviousCursorPosition, 983
HasElementAttribute, 984
InsertEntity, 984
InsertRow, 985
IsCopyEnabled, 985
IsCutEnabled, 985
IsDeleteEnabled, 986
IsEmpty, 986
IsEqual, 986
IsFirstRow, 986
IsInDynamicTable, 987
IsLastRow, 987
IsPasteEnabled, 987
IsTextStateApplied, 988
LastTextPosition, 988
LastXMLData, 989
LastXMLDataOffset, 989
MoveBegin, 990
MoveEnd, 991
MoveRowDown, 991
MoveRowUp, 991
Parent, 992
Paste, 992
PerformAction, 992
Select, 993
SelectNext, 994
SelectPrevious, 994
SetElementAttributeValue, 995
SetFromRange, 996
Text, 997
AuthenticView, 1011
Application, 1004
AsXMLString, 1004
DocumentBegin, 1005
DocumentEnd, 1006
Event, 1007
Goto, 1008
IsRedoEnabled, 1008
IsUndoEnabled, 1009
MarkupVisibility, 1009
OnBeforeCopy, 998
OnBeforeCut, 998
OnBeforeDelete, 999
OnBeforeDrop, 999
OnBeforePaste, 1000
OnMouseEvent, 1001
OnSelectionChanged, 1001
Parent, 1009
Print, 1009
Redo, 1010
Selection, 1010
Undo, 1011
WholeDocument, 1012
XMLDataRoot, 1012
Ayuda,
ver Ayuda en pantalla, 901
Ayuda (menú), 901
Ayuda en pantalla,
búsqueda, 901
contenido, 901
índice, 901
Ayudantes de entrada,
descripción, 41
ocultar, 41
ocultar automáticamente, 41
Ayudantes de entrada de la vista Diseño,
ocultar / mostrar, 810

B

Barra de estado, 809
Barra de herramientas Authentic, 851
Barra de menú,
mover, 35
Barra lateral Mensajes, 65
Barras de herramientas, 748
agregar/quitar iconos, 748
barra de herramientas de Authentic, 753
barra de herramientas Tabla, 751
Estándar, 759
formato, 750
Insertar elementos de diseño (barra de herramientas), 755
mover, 35
ocultar / mostrar, 809
ocultar/mostrar, 748
personalizar, 809
posición en la IGU, 748
restaurar, 748

© 2016-2022 Altova GmbH

Barras de herramientas, 748
RichEdit, 755
Base de datos,
botones de barras de herramientas para editar, 759
Bases de datos,
compatibles, 518, 634
ver BD, 518
ver también BD, 674
y recursos globales, 459
Bases de datos (ediciones Enterprise y Professional Edition),
ver BD, 17
BD, 620, 674, 675
archivo de datos XML, 620
archivo de esquema para, 620
archivo de esquema XML generado, 521
archivos de datos XML generados, 521
consultas en la vista Authentic, 674
crear consultas, 675
editar en la vista Authentic, 674, 679
filtrar opciones de visualización en la vista Authentic, 675
generar archivos XML y documentos de salida HTML/PDF, 632
mecanismo de funcionamiento en StyleVision, 521
navegar por tablas en la vista Authentic, 674
parámetros en consultas de BD, 675
registros que se deben mostrar en la vista Authentic, 462
seleccionar datos XML de trabajo para diseños SPS (BD XML), 617
seleccionar datos XML de trabajo para el diseño SPS, 609
seleccionar datos XML de trabajo para el diseño SPS (BD no XML), 609
seleccionar esquema para diseños SPS (BD XML), 617
seleccionar esquema para el diseño SPS, 609
seleccionar esquema para el diseño SPS (BD no XML), 609
trabajar con bases de datos en StyleVision, 518
BD XML,
cargar fila de datos nueva en la vista Authentic, 856
cargar filas de datos XML nuevas, 674
Bloques de texto definidos por el usuario, 151, 836
Bloques de texto XML definidos por el usuario, 152
Bordes,
de las tablas en el diseño SPS, 847
Botón de consulta,
insertar en el diseño SPS, 628
Botón de consulta de BD,
insertar en el diseño SPS, 628
Botones, 197
Botones de desplazamiento,
de la ventana principal, 36
Botones de opción, 197
Botones personalizados para la barra de herramientas Authentic, 851
Buscar, 65
Buscar (comando), 794
Buscar siguiente (comando), 794

C

C#,
integration of StyleVision, 1059
Cadenas de formato,
definir para formato de entrada, 863
para formato de entrada, 863
Cálculos automáticos, 293
actualizar nodo con el valor de, 816
actualizar nodos en documento XML con el valor de, 296
actualizar nodos en el documento XML con el valor de, 298
archivos de ejemplo, 299
basados en el resultado de otros cálculos automáticos, 298
comando para insertar en diseño, 816
crear, editar, aplicar formato, 293
ejemplos, 316
en el tutorial Quick Start, 93
formato de resultados de fechas, 493
funciones Java y .NET (solo Enterprise Edition), 293
mover, 293
ocultar, 296
símbolo en la vista Diseño, 724
utilización, 293
y caracteres de escape en documentos de salida, 377
y condiciones, 308
Cambiar de vista,
a la vista Authentic, 650
Campos de entrada, 191
Campos de entrada multilínea, 191
Campos numéricos,
formato, 380
Caracteres de escape en documentos de salida, 377
Características,
de StyleVision, 21
Características del producto,
lista, 21
Casillas de verificación, 192
Catalog customization, 131

Catalog files, 128
Catalog mechanism overview, 128
Catalogs and envirnment variables, 132
Catalogs in RaptorXML, 129
Celda (de una tabla),
dividir horizontalmente, 846
dividir verticalmente, 846
Celdas,
unir en tablas del diseño SPS, 846
Centro de descargas,
del sitio web de Altova, 905
Cerrar (SPS), 768
Cerrar el proyecto, 803
Claves de ordenación, 321
Codificación,
para archivos de salida, 894
Codificación (comando), 787
Codificación de los documentos de salida, 787
Codificación de salida, 787
Código clave de evaluación,
para el software de Altova, 901
Código de barras, 818
Códigos clave,
para el software de Altova, 901
Códigos de barras, 200
Columna,
anexar en una tabla del diseño SPS, 845
eliminar en una tabla del diseño SPS, 846
insertar en una tabla del diseño SPS, 845
Columnas,
forzar saltos, 835
insertar, 835
Columnas (de tablas),
ocultar en resultado HTML, 170
Columnas para formatos de salida impresos, 425
Comando Convertir en, 215
Comando Deshabilitado, 831, 842
Comandos,
personalizar, 887
COM-API,
documentation, 931
Comentarios en las consultas de BD, 642
Command reference, 1072
Compatibilidad con Internet Explorer, 116
Compatibilidad con Unicode,
de los productos de Altova, 1225
Complemento de StyleVision para Eclipse,
instalar, 712
Complemento de StyleVision para VS .NET,
instalar, 709
Comprar software de Altova, 901
Condición,
comando para insertar en el diseño, 829
Condiciones,
basadas en el resultado, 307
editar, 307
en el tutorial Quick Start, 98
establecer, 303
incluir en, 840
para diferentes formatos de salida, 307
y cálculos automáticos, 308
Condiciones basadas en el resultado, 307
Conexión de base de datos,
configurar, 523
ejemplo de configuración, 553
iniciar el asistente, 524
reutilizar desde recursos globales, 552
Configuración activa, 871
Configuración inicial de StyleVision, 28
Configuraciones,
de un recurso global, 444, 871
Configuraciones en recursos globales, 460
Configurar un documento SPS nuevo,
en el tutorial Quick Start, 68
Consulta,
ver Ventana de consulta de BD, 633
Consulta de BD,
conectarse a la BD para consultarla, 634
crear la consulta, 644
panel de mensajes, 645
panel Explorador de la ventana de consulta de BD, 637
resultados de, 645
Consulta de la BD (ventana),
ocultar / mostrar, 860
Consultar la BD,
ver Ventana de consulta de BD, 633
Consultar la BD (comando), 633
Consultas,
para visualizar BD en la vista Authentic, 675
Contenedor de diseño, 834
Contenedores de diseño, 204
Contenedores y elementos de diseño, 755
Contenido,
comando para insertar en el diseño, 812
Contenido de documentos MS Word,
copiar y pegar en el diseño, 143

© 2016-2022 Altova GmbH

Contenido de documentos Word,
copiar y pegar en el diseño, 143
Contenido de páginas HTML,
copiar y pegar en el diseño, 143
Contenido de tablas Excel,
copiar y pegar en el diseño, 143
Contenido del documento XML,
símbolo en la vista Diseño, 724
Contenido dinámico,
en el tutorial Quick Start, 73
Contenido estático,
en el tutorial Quick Start, 80
Contrato de licencia para el usuario final, 1227, 1229
Controladores de BD,
resumen, 526
Controladores de eventos,
asignar funciones a, 499
Controladores ODBC,
comprobar disponibilidad, 541
Controles de BD, 855
comando para insertar, 816
inserción automática, 628
Controles de formulario,
comandos de menú para insertar, 815
Controles de navegación de tablas de BD, 855
conversión de tipos de datos en esquemas XML generados a partir de BD, 1220
Convertir HTML en XML, 502
Copiar (comando), 794
CoreCatalog.xml, 129
Corrector ortográfico,
de StyleVision, 866
Cortar (comando), 794
Crear un documento SPS nuevo,
en el tutorial Quick Start, 68
Criterios,
de filtro para BD, 623
Cuadro combinado,
en el tutorial Quick Start, 98
Cuadro de diálogo de expresiones XPath,
descripción, 729, 731, 744
Cuadro de diseño, 834
Cuadros combinados, 194
Cuadros de diseño, 208
CustomCatalog.xml, 129

D

Datos XML,
combinar datos de varias fuentes, 248
insertar en diseño SPS, 135
Datos XML de Authentic, 648
Datos XML para diseños SPS basados en bases de datos, 609
Decimales,
formato, 380
deshabilitar-escape-de-salida,
expresión, 377
Deshacer (comando), 794
Diccionario predeterminado del usuario,
para la revisión ortográfica en diseños SPS, 867
Diccionarios,
para la revisión ortográfica en diseños SPS, 867
Diccionarios personalizados,
para la revisión ortográfica en diseños SPS, 867
Diseño de MobileTogether,
exportar como, 782
Diseño nuevo a partir de código XSLT, 471
Diseños basados en formularios, 204, 761
Dispositivos de entrada de datos, 190
comandos de menú para insertar, 815
símbolo en la vista Diseño, 724
Distribución,
de productos de software de Altova, 1227
Dividir celda de la tabla,
horizontalmente, 846
verticalmente, 846
Document,
Application, 1014
Close, 1015
FullName, 1015
GetPathName, 1015
Name, 1016
Path, 1017
Save, 1017
SaveAs, 1017
Saved, 1017
SetPathName, 1023
Documentación,
información preliminar, 33
Document-level,

Document-level,
examples of integration of XMLSpy, 1059
Documento de salida temporal, 28
Documentos,
abrir y cerrar, 36
Documentos de salida PDF,
ver Medios impresos, 430
Documentos de salida RTF,
ver Medios impresos, 430
Documents,
Count, 1025
Item, 1025
DSN de archivo,
configurar, 541
DSN de sistema,
configurar, 541
DSN de usuario,
configurar, 541
DTD,
como fuente del SPS, 223
declarar entidades sin analizar, 469
DTDs and catalogs, 128

E

Eclipse platform,
StyleVision entry points in, 722
Edición (menú), 794
Edición de contenidos, 134
Editar comando template-match, 148
Editar expresión XPath (cuadro de diálogo),
ver Cuadro de diálogo de expresiones XPath, 729
Editar expresiones XPath (cuadro de diálogo),
ver Cuadro de diálogo de expresiones XPath, 731, 744
Editar parámetros (cuadro de diálogo), 799
Editor de script, 690
Editor de scripting,
vista general, 909
Editor JavaScript, 497, 498
en la vista Diseño, 37
Editor SQL,
crear consulta en, 644
de la ventana de consulta de BD, 642
descripción, 642
Ejemplos en varios idiomas, 371
Elemento de documento,
definición, 29
Elementos,
añadir en la vista Authentic y en el SPS, 138
definidos por el usuario, 151
Elementos (ayudante de entrada),
en la vista Authentic, 654
Elementos de diseño, 755
Elementos definidos por el usuario, 151, 836, 843
Elementos documento,
ver Elementos raíz, 221
Elementos raíz, 48
Elementos raíz (elementos document),
seleccionar para el esquema, 221
y esquema fuente, 221
Elementos TDC,
aplicar formato, 351
construir, 351
Elementos XSLT,
insertar como código en el diseño, 151
Eliminar,
columna en una tabla del diseño SPS, 846
fila en una tabla del diseño SPS, 846
tabla en el diseño SPS, 844
un filtro de BD, 623
Eliminar (comando), 794
Encabezados,
agregar a la tabla, 845
de tabla, 165
Encabezados y pies de página,
con subtotales, 432
en formatos de salida para medios impresos, 430, 432
Entero,
formato, 380
Entidades,
definir en la vista Authentic, 662, 683, 753, 857
insertar en la vista Authentic, 662
sin analizar, 469
usar como marcadores de URI, 469
Entidades (ayudante de entrada),
en la vista Authentic, 654
Entorno de scripting, 907
Enumerations,
in StyleVisionControl, 1105
SPYAuthenticActions, 1047
SPYAuthenticDocumentPosition, 1047
SPYAuthenticElementKind, 1047
SPYAuthenticMarkupVisibility, 1048
SPYMouseEvent, 1048

© 2016-2022 Altova GmbH

Enumerations,
SPYXMLDataKind, 1050
Environment variables used in catalogs, 129
Environnment variables, 132
Espacios de nombres,
agregar al archivo SPS, 48, 121, 221
agregar al SPS, 113
en el archivo SPS, 48
resumen de, 52
Esquema fuente, 113
gestión en la ventana Vista general del diseño, 45
múltiples fuentes para el SPS (solo edición Enterprise Edition), 221
resumen, 52
seleccionar fuentes para el SPS, 221
ventana de la barra lateral, 48
y elementos raíz (elementos document), 221
Esquema para diseños SPS basados en bases de datos, 609, 617
Esquema principal, 265
Esquema principal (solo en la edición Enterprise Edition), 48
Esquemas,
como fuente del SPS, 223
de BD para el archivo SPS, 228
definidos por el usuario, 229
Esquemas de BD (ediciones Enterprise y Professional Edition), 228
Esquemas definidos por el usuario, 229
Esquemas fuente, 761
cambiar de fuente, 331
varias fuentes y encontrar nodos, 331
varias fuentes y XPath, 331
Esquemas XML y DTD,
como fuente del SPS, 223
Estilos,
a partir de datos XML, 402
a través de expresiones XPath, 402
asignar hojas de estilos CSS al archivo SPS, 391
combinar varios, 405
definir, 58
definir estilos globales en el archivo SPS, 395
definir estilos locales, 398
medios para asignar hojas de estilos externas, 391
orden en cascada, 390
prioridad, 55
prioridad de los estilos, 395
reglas CSS combinadas, 405
terminología, 390
trabajar con StyleVision, 390
ventana de la barra lateral, 58
ver Ayudantes de entrada, 41
y grupos de propiedades, 58
Estilos compuestos, 405
para la vista Authentic, 408
Estilos CSS,
en archivos SPS modulares, 255
en el tutorial Quick Start, 86
ver Estilos, 58
Estilos de los documentos de salida, 291
Estilos del documento, 291
Estilos globales,
ver Estilos, 395
Estilos locales,
ver Estilos, 398
Estructura del archivo SPS, 219
Estructura del diseño, 219
ventana de la barra lateral, 52
ver Ayudantes de entrada, 41
y archivos SPS modulares, 255
Estructura del diseño SPS, 219
Estructura del esquema,
y el diseño SPS, 138
Etiquetas,
expandir y contraer, 800
Etiquetas de marcado en la vista Diseño, 37
Event, 998, 999, 1000, 1001
Expresiones,
en los criterios de filtro para BD, 623
Expresiones XPath,
evaluar expresiones en el cuadro de diálogo de expresiones XPath, 731
generar expresiones en el cuadro de diálogo Editar expresión XPath, 729, 744
y estilos, 402
Extensiones de Altova,
funciones para gráficos, 1115

F

Fecha,
formato, 380
Fechas,
cambiar manualmente, 682
cómo usar fechas en archivos SPS, 491

Fechas,
ejemplos de manipulación de fechas con XPath 2.0, 491
formato, 493
ver Selector de fecha, 818, 855
y el selector de fechas, 492
Fila,
anexar en una tabla del diseño SPS, 845
eliminar en una tabla del diseño SPS, 846
insertar en una tabla del diseño SPS, 845
Filas (de tablas),
expandir/contraer en resultado HTML, 170
Filtro XPath,
en plantillas globales, 266
Filtros,
para ver determinadas plantillas, 758
Filtros de BD,
crear y modificar, 623
filtrar datos para el archivo XML, 623
tipos de datos en, 623
Filtros de diseño,
ocultar / mostrar, 810
Filtros de la BD,
borrar, 861
editar, 861
Filtros de plantillas, 758
Filtros en plantillas nodo, 274
Filtros para BD,
borrar, 861
editar, 861
Filtros para plantillas de diseño,
ocultar /mostrar, 810
Filtros XPath en plantillas nodo, 274
Finalización automática en consultas de BD, 642
Firebird,
conectarse por JDBC, 554
conectarse por ODBC, 555
Flotantes,
ventanas de los ayudantes de entrada, 41
Formato,
alineación del texto, 750
botones de barras de herramientas para, 750
de campos numéricos, 380
formatos HTML predefinidos, 750
listas, 750
para tablas, 165
propiedades de texto, 750
Formato de entrada,
de fechas, 493
definir cadenas de formato para, 863
Formato de números,
en numeración automática, 359
Formato de valores, 380
Formatos de imagen,
en la vista Authentic, 685
Formatos gráficos,
en la vista Authentic, 685
Formatos predefinidos,
al insertar un nodo, 137
comando para insertar en el diseño, 818
símbolo en la vista Diseño, 724
Fragmento de diseño,
insertar, 834
Fragmentos de diseño, 278
Función unparsed-entity-uri de XSLT, 469
Funciones,
en XPath, definidas por el usuario, 475
Funciones de extensión .NET,
campos de instancia, 1211
campos estáticos, 1210
compatibilidad en la vista Authentic, 729, 744
constructores, 1209
conversiones de tipos de datos, 1212, 1213
en expresiones XPath, 729, 744
métodos de instancia, 1211
métodos estáticos, 1210
para XSLT y XQuery, 1207
resumen, 1207
tipos de datos .NET en XPath/XQuery, 1213
tipos de datos XPath/XQuery en .NET, 1212
Funciones de extensión .NET para XSLT y XQuery,
ver Funciones de extensión .NET, 1207
Funciones de extensión en scripts MSXSL,
msxsl:script, 1213
Funciones de extensión Java,
archivos de clases definidos por el usuario, 1199
archivos JAR definidos por el usuario, 1202
campos de instancia, 1205
campos estáticos, 1204
compatibilidad en la vista Authentic, 729, 744
constructores, 1203
conversiones de tipos de datos, 1206, 1207
en expresiones XPath, 729, 744
métodos de instancia, 1205
métodos estáticos, 1204
para XSLT y XQuery, 1198
resumen, 1198

© 2016-2022 Altova GmbH

Funciones de extensión Java,
tipos de datos Java en XPath/XQuery, 1207
tipos de datos XPath/XQuery en Java, 1206
Funciones de extensión Java para XSLT y XQuery,
ver Funciones de extensión Java, 1198
Funciones de extensión para XSLT y XQuery, 1197
Funciones Java y .NET (solo Enterprise Edition),
en cálculos automáticos, 293
Funciones XPath,
cuadro de diálogo de expresiones XPath, 729
definidas por el usuario, 475
en el cuadro de diálogo de expresiones XPath, 744
Funciones XPath definidas por el usuario, 475

G

Gestor de esquemas,
aplicar un parche a un esquema, 239, 879
desinstalar un esquema, 240, 880
estado de los esquemas en el, 237, 877
funcionamiento del, 235, 875
help (comando ILC), 241, 881
info (comando ILC), 242, 882
initialize (comando ILC), 242, 882
instalar un esquema, 239, 879
install (comando ILC), 243, 883
introducción, 232, 872
introducción a la ILC, 241, 881
list (comando ILC), 244, 884
lista de esquemas por estado, 237, 877
reset (comando ILC), 244, 884
restaurar, 240, 880
uninstall (comando ILC), 245, 885
update (comando ILC), 246, 886
upgrade (comando ILC), 247, 887
Gráficos,
uso general en archivos SPS, 183
ver Imágenes, 183
Guardar,
archivo XML de trabajo, 782
Guardar archivos generados (comando), 782
Guardar diseño (comando), 773
Guardar el proyecto, 803
Guardar los datos XML de Authentic (comando), 782

H

Herramientas (menú), 866
Hipervínculo,
comando para insertar en el diseño, 826
Hipervínculos, 199, 364
crear vínculos con marcadores, 366
crear vínculos con recursos externos, 366
crear y editar, 366
encontrar mediante hipervínculos, 469
incluir en, 840
navegar por la vista Authentic, 662
quitar y eliminar, 366
símbolo en la vista Diseño, 724
y entidades sin analizar, 366
Hojas de estilos,
ver Hojas de estilos CSS, 391
ver Hojas de estilos XSLT, 391
Hojas de estilos CSS,
hojas de estilos externas, 391
medios aplicados a, 391
prioridad de importación de hojas de estilos externas, 391
ver Estilos, 391

I

IBM DB2,
conectarse por JDBC, 558
conectarse por ODBC, 560
IBM DB2 for i,
conectarse por JDBC, 566
conectarse por ODBC, 567
IBM Informix,
conectarse por JDBC, 570
Icono del estado del texto, 851
Iconos de estado del texto, 414
IE 9,
ver Compatibilidad con Internet Explorer, 116
Ignorar contenido, 842
IGU,
descripción, 35
varios documentos, 36
ventana principal, 36

IGU,
vistas de documento, 36
Imagen,
comando para insertar en el diseño, 820
Imágenes,
archivos de ejemplo, 189
en el tutorial Quick Start, 80
encontrar imágenes mediante entidades sin analizar, 469
especificar URI para, 183
presentación de resultados, 183
símbolo en la vista Diseño, 724
tipos compatibles, 185
y URI de entidades sin analizar, 183
Imágenes modelo para crear diseños, 204
Importación XSLT, 471
Importar HTML, 502
crear un archivo SPS nuevo, 502
diseño SPS, 504
estructura del esquema, 504
generar archivos a partir del diseño SPS, 508
listas HTML, 506
tablas HTML, 506
Importar plantillas XSLT,
en archivo SPS, 282
Imprimir (comando), 791
Imprimir documento de salida,
ver Medios impresos, 419
Incluir en (menú), 837
Información del usuario, 462
Información general, 1224
Información legal, 1227
Información sobre derechos de autor, 1227
Información sobre StyleVision, 905
Información técnica, 1224
Insertar,
columna en una tabla del diseño SPS, 845
fila en una tabla del diseño SPS, 845
Insertar (menú), 812
Insertar | Cálculo automático, 816
Insertar | Condición, 829
Insertar | Contenido, 812
Insertar | Fragmento de diseño, 834
Insertar | Hipervínculo, 826
Insertar | Imagen, 820
Insertar | Línea horizontal, 822
Insertar | Marcadores, 825
Insertar | Página, 835
Insertar | Párrafo, 818
Insertar | Párrafo especial, 818
Insertar | Resto de contenidos, 813
Insertar | Selector de fecha, 818
numeración y viñetas, 823
Insertar elementos de diseño desde la barra de herramientas, 755
Integrating,
StyleVision in applications, 1051
Interfaz,
ver IGU, 35
Interfaz del usuario gráfica,
ver IGU, 35

J

Java, 1063
JavaScript,
ver Scripts, 497
JDBC,
como interfaz de conexión de datos, 523
conectarse a Teradata, 602
configurar una conexión (Windows), 544
JRE,
para el complemento de StyleVision para Eclipse, 712

L

Licencia, 1229
información sobre, 1227
Licencia del producto de software, 1229
Licencias,
para el software de Altova, 901
Línea,
en contenedores de diseño, 834
Línea de comandos, 28, 699
ejemplos de comandos, 702
sintaxis, 700
usar StyleVision desde la línea de comandos, 700
y parámetros, 327
y tareas programadas, 707
Línea horizontal,
comando para insertar en el diseño, 822
en el tutorial Quick Start, 80
Listas, 178
en el tutorial Quick Start, 98

© 2016-2022 Altova GmbH

Listas, 178
importadas de un documento HTML, 506
incluir en, 839
Listas (estáticas y dinámicas), 823
Listas dinámicas, 178, 180, 823
Listas estáticas, 178, 823, 839
Localizador de objetos,
de la ventana de consulta de BD, 637
Longitud definida en píxeles,
y medios impresos, 437

M

Macros,
desarrollar, 913
Manual del usuario,
ver Ayuda en pantalla, 901
Marcado,
en la vista Authentic, 650, 652, 858
Marcado consecutivo, 37
Marcado de tamaño mixto, 462
Marcador de posición de contenido,
en el tutorial Quick Start, 73
insertar nodo como contenido, 135
Marcadores, 199, 364
comando para insertar en el diseño, 825
crear y editar, 364
eliminar, 364
incluir en, 840
Marcadores (anclajes),
símbolo en la vista Diseño, 724
Marcadores en consultas de BD, 642
Marcadores TDC, 340
asistente para, 344
crear, 344
incluir en, 842
y niveles, 344
Marcas de agua, 439
MariaDB,
conectarse por ODBC, 572
Medición de licencias,
en los productos de Altova, 1228
Medios impresos,
definiciones de página, 425
diseñar para, 419
encabezados y pies de página, 430
márgenes, 425
paginación, 425
propiedades, 419
tamaño de página, 425
y longitud definida en píxeles, 437
Menú,
personalizar, 887
Menú Authentic,
edición de tablas dinámicas, 650
presentación del marcado, 650
Menú insertar,
Insertar | Deshabilitado, 831
Menús contextuales,
en la vista Authentic, 659
Metadatos del documento HTML de resultados, 417
Microsoft Access,
conectarse por ADO, 529, 574
Microsoft Office 2007 (solo en la edición Enterprise Edition), 21, 39
Microsoft SQL Server,
conectarse por ADO, 576
conectarse por ODBC, 579
Modelo de contenido,
consecuencias de ambigüedades en el diseño, 138
Módulos,
gestión en la ventana Vista general del diseño, 45
Módulos de diseño,
pasos para crear, 204
Mostrar marcado en tamaño grande, 650, 652
Mostrar marcado en tamaño pequeño, 650, 652
Mostrar marcado mixto, 650, 652
Mostrar propiedades, 863
Motores,
de los productos de Altova, 1224
MS Access,
conversión de tipos de datos en esquemas XML generados a partir de BD, 1218
MS SQL Server,
conversión de tipos de datos en esquemas XML generados a partir de BD, 1219
Múltiples documentos de salida, 835
crear vínculos entre, 286
ubicación de los archivos, 289
vista previa, 289
MySQL,
conectarse por ODBC, 585
conversión de tipos de datos en esquemas XML generados a partir de BD, 1220

N

Niveles TDC, 340, 341
incluir en, 842
Nodo,
convertir en otro, 215
Nodo de contexto,
cuadro de diálogo de expresiones XPath, 729
en el cuadro de diálogo de expresiones XPath, 744
Nodo de documento,
definición, 29
Nodos de documento XML,
símbolo en la vista Diseño, 724
Novedades, 18
Nuevo (comando), 761
Numeración automática, 359
Numeración y viñetas, 178, 180, 823, 863
incluir en, 839
Numerar nodos automáticamente, 359
Números,
formato, 380
Números de página (edición Enterprise Edition), 835

O

Ocultar,
los ayudantes de entrada, 41
Ocultar automáticamente,
los ayudantes de entrada, 41
Ocultar marcado, 37, 650, 652
ODBC,
como interfaz de conexión de datos, 523
conectarse a MariaDB, 572
conectarse a Teradata, 604
configurar una conexión, 541
conversión de tipos de datos en esquemas XML generados a partir de BD, 1220
Office Open XML (solo en la edición Enterprise Edition), 21, 39
OLE DB,
como interfaz de conexión de datos, 523
Omitir contenido, 831
OOXML (solo en la edición Enterprise Edition), 21, 39
Opciones de compatibilidad, 33
Opciones de la estructura del esquema, 894
Opciones del corrector ortográfico,
para el diseño SPS, 867
Opciones del proyecto, 894
Operadores XPath,
cuadro de diálogo de expresiones XPath, 729
en el cuadro de diálogo de expresiones XPath, 744
Oracle,
conectarse por JDBC, 587
conectarse por ODBC, 589
conversión de tipos de datos en esquemas XML generados a partir de BD, 1221
Ordenar, 321
archivos de ejemplo, 323
claves de ordenación, 321
grupos y elementos de grupos, 312, 315, 316
mecanismo de ordenación, 321
Organización,
de las vistas en la IGU, 41
Overview,
of API de XMLSpy, 932

P

Página,
comando para el diseño, 835
definir márgenes para formato de salida PDF, 425
definir tamaño para formato de salida PDF, 425
numeración en formato de salida PDF, 425
Panel de consulta,
de la ventana de consulta de BD, 642
Panel Explorador,
de la ventana de consulta de BD, 637
Paquete de integración de StyleVision, 709, 712
Parámetros, 327
crear y editar, 799
descripción general, 327
en el archivo SPS, 327
en filtros de BD, 623
en las consultas de BD, 675
encontrar nodos en varios documentos con, 331
gestión en la ventana Vista general del diseño, 45
para esquemas fuente, 331
para fragmentos de diseño, 329
resumen de parámetros definidos por el usuario, 52
y la línea de comandos, 327

© 2016-2022 Altova GmbH

Parámetros, 327
y la vista Authentic, 327
Parámetros de BD,
crear y editar, 799
uso, 623
Parent, 1016
Párrafo,
comando para insertar en el diseño, 818
incluir en, 838
Párrafo especial,
comando para insertar en el diseño, 818
incluir en, 838
PDF,
definir propiedades de página, 425
Pegar,
como texto, 662
como XML, 662
Pegar (comando), 794
Pegar como,
Texto, 659
XML, 659
Período de evaluación,
de los productos de software de Altova, 1227
Personalizar (cuadro de diálogo),
personalizar StyleVision, 809
Personalizar StyleVision, 887
Perspectiva StyleVision en Eclipse, 719
Pies,
de tabla, 165
Pies de página y encabezados,
con subtotales, 432
en formatos de salida para medios impresos, 430, 432
Pies de tabla,
agregar a la tabla, 845
Píxeles,
y longitud de medios impresos, 787
y resolución de la pantalla, 787
Plantilla,
cambiar el nodo por, 215
incluir en, 837
insertar, 831
Plantilla definida por el usuario,
incluir en, 838
insertar, 832
Plantilla global compleja, 266
Plantilla global sencilla, 266
Plantilla local, 265
Plantilla principal, 265
definición, 29
Plantilla TDC,
aplicar formato, 351
crear y editar, 347
estructurar, 349
niveles de referencia en, 349
referencias de nivel en, 349
Plantilla variable, 273
incluir en, 838
insertar, 833
Plantillas,
estructura, 52
incluir filas y columnas de tabla en, 164
ocultar / mostrar, 810
quitar filas y columnas de tabla de, 164
Plantillas basadas en tipos, 266
Plantillas con nombre, 265
Plantillas condicionales, 829
símbolo en la vista Diseño, 724
ver Condiciones, 303
Plantillas de documento nuevas, 284
direcciones URL de, 286
insertar, 285
y estructura del diseño, 286
Plantillas de elemento,
definidas por el usuario, 151
Plantillas de nodo,
definidas por el usuario, 148
Plantillas definidas por el usuario, 148
Plantillas globales, 265, 266
efectos en el resto de contenidos, 141
en el tutorial Quick Start, 106
Plantillas nodo,
operaciones, 274
y encadenamiento con plantillas secundarias, 274
y filtros XPath, 274
y plantillas globales, 274
Plantillas para nodos,
ver Plantillas nodos, 274
Plantillas XSLT, 52
gestión en la ventana Vista general del diseño, 45
importar en archivo SPS, 282
Plataforma Eclipse,
perspectiva StyleVision en, 719
y el paquete de integración de StyleVision, 712
y StyleVision, 711
Plataformas,
para productos de Altova, 1224

© 2016-2022 Altova GmbH

Portadas, 423
PostgreSQL,
conectarse directamente (de forma nativa), 549
conectarse por ODBC, 595
PPP, 787
y formato de salida impreso, 437
y longitud definida en píxeles, 437
Preguntas frecuentes sobre StyleVision, 905
Presencia condicional, 310
Presentación,
resumen de procedimientos, 374
ver Formatos, Aplicar formato, 374
Prioridad,
en los estilos, 55
Problemas con la vista previa, 28
Procedimientos de edición adicionales, 443
Procesador FO (edición Enterprise Edition),
configurar, 28
Procesador XQuery,
de los productos de Altova, 1224
Procesadores,
para descargar, 905
Procesadores XSLT,
de los productos de Altova, 1224
Procesamiento alternativo, 464
Procesamiento automatizado, 699
Progress OpenEdge (base de datos),
conectarse por JDBC, 596
conectarse por ODBC, 598
Propiedades,
de las tablas del diseño SPS, 751
de tablas del diseño SPS, 847
definir, 59
para nodos en la vista Authentic, 462
ventana de la barra lateral, 59
ver Ayudantes de entrada, 41
y grupos de propiedades, 59
Propiedades (ayudante de entrada),
grupo Evento, 499
Propiedades (menú), 863
numeración y viñetas, 863
Propiedades de los documentos de salida, 291
Propiedades de nodos de Authentic, 462
Propiedades de página en PDF, 425
Propiedades de un documento HTML, 417
Propiedades del documento, 291
Proyecto,
ventana de la barra lateral, 63
Proyecto (menú), 802
comando Abrir, 803
comando Agregar archivo activo al proyecto, 806
comando Agregar archivo activo y relacionados al proyecto, 806
comando Agregar archivos al proyecto, 804
comando Agregar carpeta de proyecto al proyecto, 806
comando Agregar carpeta externa / carpeta web al proyecto, 806
comando Agregar recurso global al proyecto, 804
comando Agregar URL al proyecto, 804
comando Cerrar, 803
comando Guardar, 803
comando Nuevo, 803
comando Volver a cargar, 803
Proyecto nuevo, 803
Proyectos,
arrastrar y colocar, 802
descripción detallada, 123
usar, 123

R

Rama condicional Otherwise, 303
RaptorXML, 699
y FOP, 703
Recursos globales, 444
barra de herramientas, 759
cambiar de configuración, 460
cuadro de diálogo, 870
definir, 444
definir el tipo de archivo, 447
definir el tipo de base de datos, 453
definir el tipo de carpeta, 452
seleccionar configuración desde la barra de herramientas, 759
usar, 455, 456, 459, 460
Recursos globales de Altova,
ver Recursos globales, 444
Reemplazar, 65
Reemplazar (comando), 794
Referencia del usuario, 723
Referencias cruzadas, 363
Referencias de caracteres,
y caracteres de escape en documentos de salida, 377
Referencias de entidades,
y caracteres de escape en documentos de salida, 377

© 2016-2022 Altova GmbH

Referencias de texto, 363
Referencias TDC, 350
RefTDC,
ver Referencias TDC, 350
Regiones en las consultas de BD, 642
Registrar el software de Altova, 901
Registros en la vista Authentic,
configurar, 628
Rehacer (comando), 794
Repositorio de estilos,
ventana de la barra lateral, 55
ver Ayudantes de entrada, 41
y estilos globales, 395
y hojas de estilos CSS externas, 391
Requisitos de memoria, 1224
Resolución de la pantalla,
y longitud definida en píxeles, 437
Restaurar barras de herramientas y ventanas, 894
Resto de contenido,
y plantillas globales, 266
Resto de contenidos, 141
comando para insertar en el diseño, 813
en el tutorial Quick Start, 106
Resultado HTML, 121
imágenes compatibles, 185
Resultado PDF (Edición Enterprise), 121
Resultado PDF (sólo edición Enterprise Edition),
imágenes compatibles, 185
Resultado RTF (Edición Enterprise), 121
Resultado RTF (sólo edición Enterprise Edition),
imágenes compatibles, 185
Retorno de carro,
ver Tecla Entrar, 685
RichEdit, 410
comandos en la vista Authentic, 858
insertar, 813
RichEdit (barra de herramientas), 755
RootCatalog.xml, 129

S

Salir (comando), 792
Saltos, 434
Saltos de página, 434, 835
Schemas,
looking up via catalogs, 131
Schemas and catalogs, 128
Scripting,
en la vista Authentic, 687, 690, 691
Scripting de Authentic, 687, 690
macros, 691
Scripts,
definir funciones JavaScript, 498
en la estructura del diseño, 497
funciones JavaScript como controladores de eventos, 499
resumen, 52
usar en un archivo SPS, 497
y funciones JavaScript, 497
Scripts en XSLT/XQuery,
ver Funciones de extensión, 1197
Sección inicial del documento, 423
Secciones,
del diseño SPS, 420
eliminar, 420
insertar, 835
propiedades predeterminadas para secciones nuevas, 423
sección inicial del documento, 423
y diseño de página, 420
Secciones CDATA, 135
insertar en la vista Authentic, 662
Selección de tablas (cuadro de diálogo),
para diseños SPS basados en bases de datos, 609, 617
Seleccionar todo (comando), 794
Selector de fecha,
añadir a nodos de fecha de forma predeterminada, 855
comando para insertar en el diseño, 818
usar en la vista Authentic, 681
Selector de fechas,
formato léxico de las entradas de fecha, 492
insertar en archivo SPS, 492
usar en la vista Authentic, 492
uso general, 492
y tipos de datos XSD, 492
Símbolos de la vista Diseño,
cálculos automáticos, 724
contenido del documento XML, 724
dispositivos de entrada de datos, 724
formatos predefinidos, 724
hipervínculos, 724
imágenes, 724
marcadores (anclajes), 724
nodos de documento XML, 724
plantillas condicionales, 724
Sitio web de Altova, 905

© 2016-2022 Altova GmbH

SO,
para productos de Altova, 1224
Software de acompañamiento,
para descargar, 905
Soporte técnico para StyleVision, 905
SPS,
abrir, 768
cerrar, 768
descripción general, 26
volver a cargar, 768
y BD, 518
y la vista Authentic (ediciones Enterprise y Professional Edition), 26
y las hojas de estilos XSLT, 26
y StyleVision, 26
SPS y la vista Authentic, 118
SQL Server,
conectarse por ADO, 529
conectarse por ADO.NET, 534
SQLite,
configurar una conexión (Windows), 550
StyleVision,
características del producto, 21
ejecución desde la línea de comandos, 700
integration, 1051
manual del usuario, 17
sincronizar con Authentic, 120
StyleVision Power Stylesheet,
ver SPS, 17
StyleVisionCommand,
in StyleVisionControl, 1086
StyleVisionCommands,
in StyleVisionControl, 1088
StyleVisionControl, 1089
documentation of, 1051
examples of integration at document level, 1059
integration using C#, 1059
object reference, 1086
StyleVisionControlDocument, 1097
StyleVisionControlPlaceHolder, 1103
Subtotales,
en encabezados y pies de página, 432
Sybase,
conectarse por JDBC, 601
conversión de tipos de datos en esquemas XML generados a partir de BD, 1222

T

Tabla,
agregar encabezados y pies de página, 845
alineación vertical del contenido de las celdas, 848
anexar columna en, 845
anexar fila en, 845
contenido de las celdas, 844
editar en la vista Authentic, 859
editar propiedades, 847
eliminar columna, 846
eliminar en el diseño SPS, 844
eliminar fila, 846
encabezados y pies de tabla, 844
insertar columna en, 845
insertar en una tabla estática, 844
insertar fila en, 845
mostrar/ocultar bordes en StyleVision, 847
navegar, 844
Tabla (menú), 844
Tabla de contenido,
ver TDC, 337
Tabla dinámica,
botones de la barra de herramientas para editar, 751
Tabla estática,
botones de la barra de herramientas para editar, 751
insertar, 844
insertar en el diseño SPS, 751
Tablas,
botón Cerrar para ocultar columnas, 170
crear, 822
crear tablas dinámicas, 158
crear tablas estáticas, 157
editar tablas dinámicas (SPS), 650
encabezados y pies de tabla en PDF, 165
estilos para filas alternas, 402
expandir/contraer filas, 170
formato, 165
importadas de un documento HTML, 506
ocultar columnas vacías, 170
procesamiento condicional en, 162
unir celdas, 846
vista general, 155
Tablas (SPS),
botones de la barra de herramientas para editar, 751

© 2016-2022 Altova GmbH

Tablas (SPS),
editar propiedades, 751
Tablas (SPS) dinámicas en la vista Authentic,
uso de, 667
Tablas (SPS) estáticas en la vista Authentic,
uso de, 667
Tablas CALS/HTML, 172, 848
Tablas dinámicas, 155
diferencias entre filas anexadas e insertadas, 158
edición, 650
editar en la vista Authentic, 859
encabezado y pie de tablas dinámicas, 158
tablas dinámicas anidadas, 158
ver Tablas, 165
ver Tablas SPS, 158
y plantillas globales, 158
Tablas en la vista Authentic,
iconos para editar tablas XML, 672
usar tablas SPS (estáticas y dinámicas), 667
usar tablas XML, 668
uso de, 667
Tablas en la vista Diseño,
incluir en y quitar plantillas, 164
representación, 164
Tablas estáticas, 155
ver Tablas, 165
ver Tablas SPS, 157
Tablas HTML, 172, 848
Tablas SPS,
editar tablas dinámicas, 650
ver Tablas dinámicas, 155
ver Tablas estáticas, 155
Tablas SPS en la vista Authentic,
uso de, 667
Tablas XML (ediciones Enterprise y Professional Edition), 155
Tablas XML en la vista Authentic,
iconos para editar, 672
uso de, 668
Tareas programadas, 707
TDC,
comandos de menú, 834
ejemplo de TDC jerárquica y secuencial, 356
ejemplo sencillo, 352
marcar elementos para incluir en, 340
uso general, 337
Tecla Entrar,
efectos de usar, 685
Teclas de acceso rápido,
personalizar, 887
Teradata,
conectarse por JDBC, 602
conectarse por ODBC, 604
terminate, 956
Terminología,
utilizada en StyleVision, 29
Texto,
editar en la vista Authentic, 662
formato en la vista Authentic, 662
Texto estático,
en el tutorial Quick Start, 80
y caracteres de escape en documentos de salida, 377
Texto vertical,
en celdas de tabla, 165
en cuadros de diseño, 208
Tipos como unidades de procesamiento,
en plantillas globales, 266
Tipos de datos,
en filtros de BD, 623
ver Tipos de datos XML Schema, 1218
Tipos de datos XML Schema,
al generar esquemas XML a partir de BD ADO, 1217
al generar esquemas XML a partir de BD MS Access, 1218
al generar esquemas XML a partir de BD MS SQL Server, 1219
al generar esquemas XML a partir de BD MySQL, 1220
al generar esquemas XML a partir de BD ODBC, 1220
al generar esquemas XML a partir de BD Oracle, 1221
al generar esquemas XML a partir de BD Sybase, 1222
Tipos globales,
en plantillas, 266
Toolbar buttons,
adding and removing, 749
Total de páginas (edición Enterprise Edition), 835
Totales acumulados,
en encabezados y pies de página, 432
Transformaciones FO, 703
Transformaciones XSLT, 703
Tutorial de introducción rápida,
archivos necesarios, 67
introducción, 67
Quick Start, 67
Tutorial Quick Start,
atributos de clase, 86
cálculos automáticos, 93
condiciones, 98

© 2016-2022 Altova GmbH

Tutorial Quick Start,
configurar un documento SPS nuevo, 68
contenido dinámico, 73
contenido estático, 80
crear un documento SPS nuevo, 68
cuadros combinados, 98
estilos CSS, 86
formatos predefinidos, 86
generar hojas de estilos XSLT, 111
imágenes, 80
líneas horizontales, 80
listas, 98
marcador de posición de contenido, 73
plantillas globales, 106
probar la vista Authentic (ediciones Enterprise y Professional Edition), 111
resto de contenidos, 106
texto estático, 80

U

Unir celdas,
en tablas del diseño SPS, 846
URI,
dentro de entidades sin analizar, 469
Uso, 112
Uso de Internet,
en los productos de Altova, 1225
Uso general, 112

V

Validación adicional, 467
Validar XML,
en la vista Authentic, 856
Valores predeterminados de parámetro,
en filtros de BD, 623
Valores predeterminados de parámetro de BD, 623
Variables, 327, 332
asignar valores desde Authentic, 464
editar en la vista Authentic, 334
Variables editables, 334
Varios documentos de salida, 284
Ventana de consulta de BD, 633
Ventana Fuentes,
ver Ayudantes de entrada, 41
Ventanas (menú), 900
Ventanas de los ayudantes de entrada,
acoplar, 41
flotantes, 41
Ver marcado, 37
Versión XPath en el archivo SPS, 115
Versión XSLT,
establecer versión XSLT para el diseño SPS, 759
Versión XSLT en el archivo SPS, 115
Versiones nuevas de StyleVision,
sincronizar con StyleVision, 120
Vínculos,
ver Hipervínculos, 199, 364
Vista (menú), 809
Vista Authentic,
archivo XML de trabajo, 38
ayudantes de entrada en, 654
botones de barras de herramientas para, 753
de los productos Altova, 25
descripción, 38
editar datos en una BD XML, 856
formato de texto en, 650
guardar cambios, 782
iconos de la barra de herramientas, 650
iconos para tablas XML, 672
interfaz, 649
interfaz gráfica del usuario (IGU), 649
menús contextuales, 659
pegar como XML o como texto, 659
presentación de marcado en, 652
presentación del marcado en, 650
proceso de creación de documentos, 118
sincronizar con la nueva versión de StyleVision, 120
tablas (SPS y XML), 667
Tablas SPS, 667
tablas XML, 668
uso de funciones importantes, 662
uso de tablas XML, 668
ventana principal de, 652
visualización de documentos, 652
y el archivo SPS, 118
y esquemas estándar, 118
Vista Diseño,
ayudantes de entrada (ocultar / mostrar), 810
descripción, 37
editor de JavaScript, 37
símbolos del diseño SPS, 724

© 2016-2022 Altova GmbH

Vista Diseño,
ver marcado, 37
Vista general del diseño,
ventana de la barra lateral, 45
Vista general del diseño del archivo SPS, 114
Vista previa de hojas de estilos XSLT,
en vistas de resultados, 39
Vista previa de impresión (comando), 791
Vistas,
organización en la IGU, 41
Vistas de documento,
en la IGU, 36
Vistas de resultados,
descripción, 39
Visual Studio,
adding the StyleVision ActiveX Controls to the toolbox, 1052
Visual Studio .Net,
diferencias del complemento para VS, 710
y StyleVision, 708
Volver a cargar el proyecto, 803
VS .NET,
y el paquete de integración de StyleVision, 709

W

Windows,
compatibilidad con productos de Altova, 1224
Word 2007 (solo en la edición Enterprise Edition), 21, 39
WordXML (solo en la edición Enterprise Edition), 21, 39

X

XML,
insertar en el diseño, 152
XMLData,
AppendChild, 1035
EraseAllChildren, 1036
EraseCurrentChild, 1036
GetChild, 1037
GetChildKind, 1038
GetCurrentChild, 1039
GetFirstChild, 1039
GetNextChild, 1040
HasChildren, 1041
HasChildrenKind, 1041
InsertChild, 1042
IsSameNode, 1043
Kind, 1043
MayHaveChildren, 1043
Name, 1044
Parent, 1044
TextValue, 1044
XMLSpy, 25
XMLSpyLib, 931
Application, 952
AuthenticRange, 969
AuthenticView, 997
Document, 1012
Documents, 1024
XMLData, 1033
XPath,
encontrar nodos en varios documentos, 331
XPath 1.0,
y fechas, 491
XPath 2.0,
y fechas, 491
XPath al nodo seleccionado, 649
XPath dialog,
debugging expressions in, 735
XQuery,
funciones de extensión, 1197
xs:date,
y el selector de fechas, 492
xs:dateTime,
y el selector de fechas, 492
XSLT,
diseño nuevo a partir de, 471
funciones de extensión, 1197
importar en SPS, 471
insertar fragmento de código en el diseño, 152

© 2016-2022 Altova GmbH